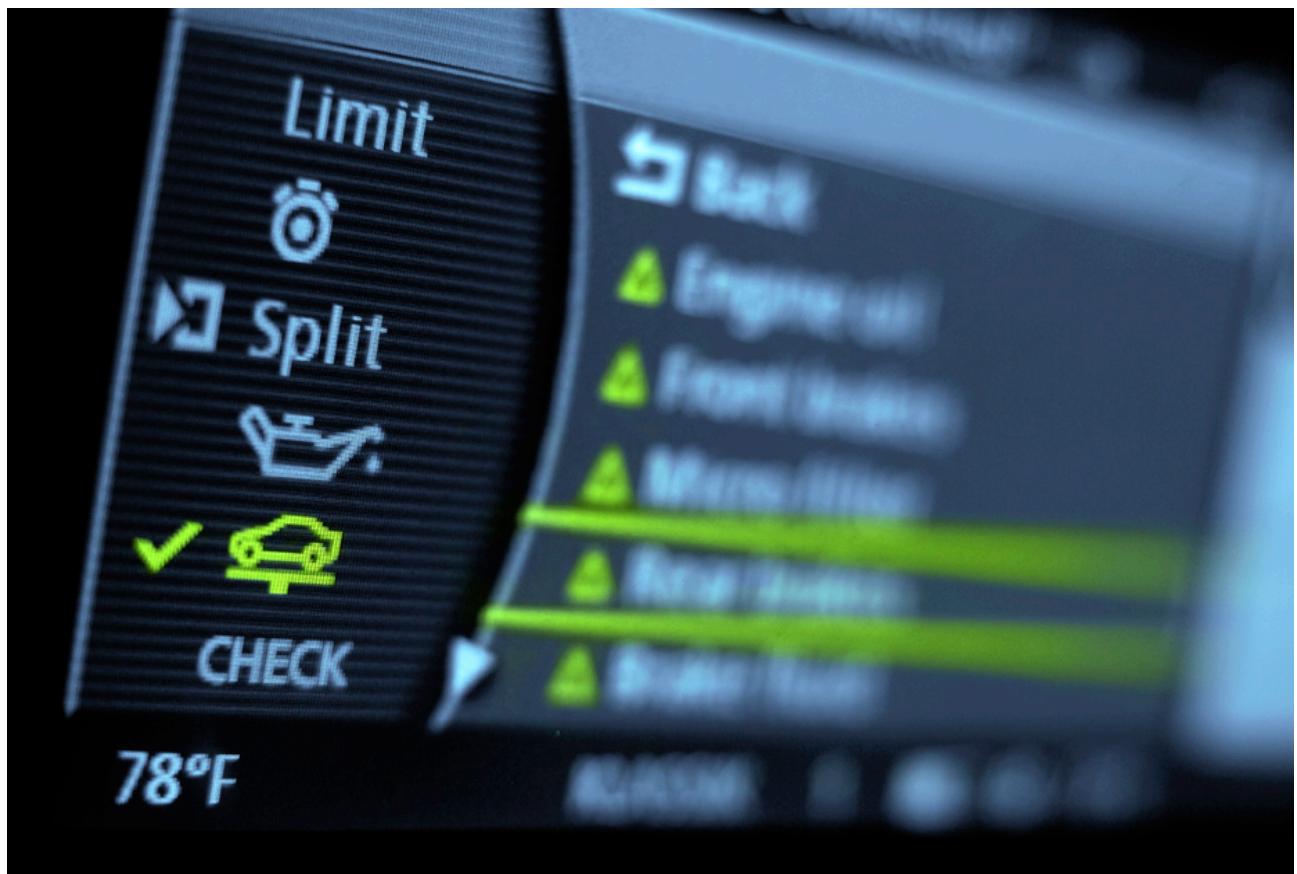




Elektrobit

# EB tresos<sup>®</sup> AutoCore Generic 8 Mode Management documentation

product release 8.8.4





Elektrobit Automotive GmbH  
Am Wolfsmantel 46  
91058 Erlangen, Germany  
Phone: +49 9131 7701 0  
Fax: +49 9131 7701 6333  
Email: info.automotive@elektrobit.com

## Technical support

<https://www.elektrobit.com/support>

## Legal disclaimer

Confidential information.

ALL RIGHTS RESERVED. No part of this publication may be copied in any form, by photocopy, microfilm, retrieval system, or by any other means now known or hereafter invented without the prior written permission of Elektrobit Automotive GmbH.

All brand names, trademarks, and registered trademarks are property of their rightful owners and are used only for description.

Copyright 2021, Elektrobit Automotive GmbH.

# Table of Contents

1. Overview of EB tresos AutoCore Generic 8 Mode Management documentation .....	25
2. Supported features .....	26
2.1. Overview .....	26
2.2. Supported BswM features .....	26
2.3. Supported ComM features .....	27
2.4. Supported EcuM features .....	28
2.5. Supported Nm features .....	28
3. ACG8 Mode Management release notes .....	30
3.1. Overview .....	30
3.2. Scope of the release .....	30
3.2.1. Configuration tool .....	30
3.2.2. AUTOSAR modules .....	30
3.2.3. EB (Elektrobit) modules .....	31
3.2.4. MCAL modules and EB tresos AutoCore OS .....	31
3.3. Module release notes .....	31
3.3.1. BswM module release notes .....	31
3.3.1.1. Change log .....	31
3.3.1.2. New features .....	44
3.3.1.3. EB-specific enhancements .....	44
3.3.1.4. Deviations .....	45
3.3.1.5. Limitations .....	48
3.3.1.6. Open-source software .....	48
3.3.2. BswMAs module release notes .....	48
3.3.2.1. Change log .....	49
3.3.2.2. New features .....	54
3.3.2.3. EB-specific enhancements .....	54
3.3.2.4. Deviations .....	54
3.3.2.5. Limitations .....	54
3.3.2.6. Open-source software .....	54
3.3.3. ComM module release notes .....	54
3.3.3.1. Change log .....	55
3.3.3.2. New features .....	64
3.3.3.3. EB-specific enhancements .....	64
3.3.3.4. Deviations .....	65
3.3.3.5. Limitations .....	70
3.3.3.6. Open-source software .....	70
3.3.4. EcuM module release notes .....	70
3.3.4.1. Change log .....	70
3.3.4.2. New features .....	79



3.3.4.3. EB-specific enhancements .....	79
3.3.4.4. Deviations .....	81
3.3.4.5. Limitations .....	90
3.3.4.6. Open-source software .....	91
3.3.5. Nm module release notes .....	91
3.3.5.1. Change log .....	91
3.3.5.2. New features .....	97
3.3.5.3. EB-specific enhancements .....	97
3.3.5.4. Deviations .....	98
3.3.5.5. Limitations .....	100
3.3.5.6. Open-source software .....	101
4. ACG8 Mode Management user guide .....	102
4.1. Overview .....	102
4.2. Background information .....	102
4.3. Configuring the EcuM and the BswM modules .....	103
4.3.1. Overview .....	103
4.3.2. General configuration/integration hints .....	103
4.3.2.1. Migrating the EcuM from ACG-8.5.2 to ACG-8.6.0 .....	104
4.3.2.2. Integration notes for multi-core projects .....	104
4.3.2.3. Integration notes for cross-core wakeup event handling between EcuM and ComM .....	106
4.3.3. Configuring EcuM and BswM for single-core projects .....	107
4.3.3.1. Configuring the startup states .....	107
4.3.3.2. Configuring the shutdown states .....	111
4.3.3.3. Configuring the sleep states .....	114
4.3.3.4. Configuring the wakeup states .....	116
4.3.4. Configuring EcuM and BswM for multi-core projects .....	119
4.3.4.1. Configuring startup on systems with multiple cores .....	119
4.3.4.2. Configuring shutdown on systems with multiple cores .....	120
4.3.4.3. Configuring sleep on systems with multiple cores .....	122
4.3.4.4. Configuring the wakeup state on systems with multiple cores .....	123
4.4. BswM module user guide .....	125
4.4.1. Overview .....	125
4.4.2. Background information .....	125
4.4.2.1. Mode arbitration .....	125
4.4.2.2. Queued mode arbitration .....	127
4.4.2.2.1. Functional overview .....	127
4.4.2.2.2. Configuration options .....	127
4.4.2.2.3. Configuration examples .....	128
4.4.2.2.3.1. Example 1 .....	128
4.4.2.2.3.2. Example 2 .....	128
4.4.2.2.3.3. Example 3 .....	129



4.4.2.3. Multi-core and BSW distribution support .....	129
4.4.2.3.1. Functional overview .....	130
4.4.2.3.2. Configuration options .....	130
4.4.2.4. Post-build selectable support .....	131
4.4.3. Configuring the BswM module .....	131
4.4.3.1. BswM Editor user guide .....	131
4.4.3.1.1. Overview .....	131
4.4.3.1.2. Background information .....	131
4.4.3.1.2.1. Naming conventions .....	132
4.4.3.1.3. Using the BswM Editor .....	133
4.4.3.1.3.1. Starting the BswM Editor .....	134
4.4.3.1.3.2. Creating a new mode request port .....	135
4.4.3.1.3.3. Creating a new mode switch port .....	136
4.4.3.1.3.4. Creating a new event request port .....	137
4.4.3.1.3.5. Creating a new send port .....	138
4.4.3.1.3.6. Creating a new rule .....	139
4.4.3.1.3.7. Creating a new expression .....	140
4.4.3.1.3.8. Creating a new condition .....	141
4.4.3.1.3.8.1. Mode conditions .....	141
4.4.3.1.3.8.2. Event conditions .....	142
4.4.3.1.3.8.3. Legacy event conditions .....	142
4.5. ComM module user guide .....	143
4.5.1. Overview .....	143
4.5.2. Background information .....	143
4.5.2.1. BSW distribution .....	143
4.5.2.1.1. Functional overview .....	144
4.5.3. Configuring the ComM module .....	145
4.5.3.1. Configuring ComM for BSW distribution .....	145
4.6. EcuM module user guide .....	146
4.6.1. Overview .....	146
4.6.2. Background information .....	146
4.6.2.1. EcuM Flex support .....	146
4.6.2.2. EcuM Fixed support .....	148
4.6.2.2.1. Functional overview .....	148
4.6.2.2.2. Configuration options .....	149
4.6.2.3. Multi-core and BSW distribution support .....	149
4.6.2.3.1. Functional overview .....	149
4.6.2.3.2. Configuration options .....	150
4.6.2.4. Post-build selectable support .....	151
4.6.2.4.1. Functional overview .....	151
4.6.2.4.2. Configuration options .....	151
4.7. Nm module user guide .....	152



4.7.1. Overview .....	152
4.7.2. Background information .....	152
4.7.2.1. Network management state information .....	152
4.7.2.1.1. Functional overview .....	152
4.7.2.1.2. Configuration options .....	153
4.7.2.2. BSW distribution .....	153
4.7.2.2.1. Functional overview .....	153
4.7.3. Configuring the Nm module .....	154
4.7.3.1. Configuring Nm for BSW distribution .....	154
5. ACG8 Mode Management module references .....	155
5.1. Overview .....	155
5.1.1. Notation in EB module references .....	155
5.1.1.1. Default value of configuration parameters .....	155
5.1.1.2. Range information of configuration parameters .....	155
5.2. BswM .....	156
5.2.1. Configuration parameters .....	156
5.2.1.1. BswMConfig .....	157
5.2.1.2. BswMArbitration .....	158
5.2.1.3. BswMLogicalExpression .....	160
5.2.1.4. BswMModeCondition .....	161
5.2.1.5. BswMConditionValue .....	162
5.2.1.6. BswMBswMode .....	162
5.2.1.7. BswMCompuScaleModeValue .....	162
5.2.1.8. BswMModeDeclaration .....	163
5.2.1.9. BswMEventRequestPort .....	164
5.2.1.10. BswMEventRequestSource .....	164
5.2.1.11. BswMComMIInitiateReset .....	165
5.2.1.12. BswMModeRequestPort .....	165
5.2.1.13. BswMModeInitValue .....	166
5.2.1.14. BswMBswModeInitValue .....	166
5.2.1.15. BswMCompuScaleModeValue .....	166
5.2.1.16. BswMSwcModeInitValue .....	167
5.2.1.17. BswMModeRequestSource .....	168
5.2.1.18. BswMBswModeNotification .....	170
5.2.1.19. BswMCanSMIndication .....	170
5.2.1.20. BswMComMIIndication .....	171
5.2.1.21. BswMComMPncRequest .....	171
5.2.1.22. BswMDcmApplicationUpdatedIndication .....	172
5.2.1.23. BswMDcmComModeCurrentState .....	172
5.2.1.24. BswMDcmComModeRequest .....	172
5.2.1.25. BswMDcmResetModeRequest .....	173
5.2.1.26. BswMDcmSessionModeRequest .....	173



5.2.1.27. BswMEcuMIndication .....	173
5.2.1.28. BswMEcuMWakeupSource .....	173
5.2.1.29. BswMEthIfPortGroupLinkStateChg .....	174
5.2.1.30. BswMEthSMIndication .....	174
5.2.1.31. BswMFrSMIndication .....	175
5.2.1.32. BswMGenericRequest .....	175
5.2.1.33. BswMJ1939DcmBroadcastStatus .....	176
5.2.1.34. BswMJ1939NmIndication .....	177
5.2.1.35. BswMLinScheduleIndication .....	177
5.2.1.36. BswMLinSMIndication .....	178
5.2.1.37. BswMLinTpModeRequest .....	178
5.2.1.38. BswMNmIfCarWakeUpIndication .....	179
5.2.1.39. BswMNVmJobModeIndication .....	179
5.2.1.40. BswMNVmRequest .....	180
5.2.1.41. BswMRteSwitchAckNotification .....	180
5.2.1.42. BswMSdClientServiceCurrentState .....	181
5.2.1.43. BswMSdConsumedEventGroupCurrentState .....	181
5.2.1.44. BswMSdEventHandlerCurrentState .....	181
5.2.1.45. BswMSwcModeNotification .....	182
5.2.1.46. BswMSwcModeRequest .....	182
5.2.1.47. BswMTimer .....	183
5.2.1.48. BswMWdgMRequestPartitionReset .....	183
5.2.1.49. BswMRule .....	184
5.2.1.50. BswMDataTypeMappingSets .....	186
5.2.1.51. BswMModeControl .....	187
5.2.1.52. BswMAction .....	187
5.2.1.53. BswMAvailableActions .....	188
5.2.1.54. BswMClearEventRequest .....	190
5.2.1.55. BswMComMAllowCom .....	190
5.2.1.56. BswMComMModeLimitation .....	191
5.2.1.57. BswMComMModeSwitch .....	192
5.2.1.58. BswMDeadlineMonitoringControl .....	193
5.2.1.59. BswMEcuMGoDown .....	193
5.2.1.60. BswMEcuMSelectShutdownTarget .....	194
5.2.1.61. BswMEthIfSwitchPortGroupRequestMode .....	195
5.2.1.62. BswMJ1939DcmStateSwitch .....	196
5.2.1.63. BswMJ1939RmStateSwitch .....	197
5.2.1.64. BswMLinScheduleSwitch .....	198
5.2.1.65. BswMNMCControl .....	199
5.2.1.66. BswMPduGroupSwitch .....	200
5.2.1.67. BswMPduRouterControl .....	201
5.2.1.68. BswMRteModeRequest .....	201



5.2.1.69. BswMRteSwitch .....	202
5.2.1.70. BswMSchMSwitch .....	203
5.2.1.71. BswMSdClientServiceModeRequest .....	204
5.2.1.72. BswMSdConsumedEventGroupModeRequest .....	205
5.2.1.73. BswMSdServerServiceModeRequest .....	206
5.2.1.74. BswMSwitchIPduMode .....	206
5.2.1.75. BswMTimerControl .....	207
5.2.1.76. BswMTriggerIPduSend .....	208
5.2.1.77. BswMUserCallout .....	209
5.2.1.78. BswMActionList .....	209
5.2.1.79. BswMActionListItem .....	210
5.2.1.80. BswMRteModeRequestPort .....	212
5.2.1.81. BswMRteModeRequestPortInterfaceRef .....	212
5.2.1.82. BswMSwitchPort .....	213
5.2.1.83. BswMGeneral .....	214
5.2.1.84. BswMQueuedArbitration .....	224
5.2.1.85. BswMUserIncludeFiles .....	225
5.2.1.86. ReportToDem .....	226
5.2.1.87. BswMDefensiveProgramming .....	227
5.2.1.88. CommonPublishedInformation .....	230
5.2.1.89. PublishedInformation .....	233
5.2.2. Application programming interface (API) .....	234
5.2.2.1. Macro constants .....	234
5.2.2.1.1. BSWM_AR_RELEASE_MAJOR_VERSION .....	234
5.2.2.1.2. BSWM_AR_RELEASE_MINOR_VERSION .....	234
5.2.2.1.3. BSWM_AR_RELEASE_REVISION_VERSION .....	234
5.2.2.1.4. BSWM_E_INVALID_PAR .....	234
5.2.2.1.5. BSWM_E_NO_INIT .....	234
5.2.2.1.6. BSWM_E_NULL_POINTER .....	235
5.2.2.1.7. BSWM_E_PARAM_CONFIG .....	235
5.2.2.1.8. BSWM_E_PARAM_POINTER .....	235
5.2.2.1.9. BSWM_E_QUEUE_FULL .....	235
5.2.2.1.10. BSWM_E_REQ_MODE_OUT_OF_RANGE .....	235
5.2.2.1.11. BSWM_E_REQ_USER_OUT_OF_RANGE .....	236
5.2.2.1.12. BSWM_E_WRONG_CONTEXT .....	236
5.2.2.1.13. BSWM_MODULE_ID .....	236
5.2.2.1.14. BSWM_NO_CFGCLASSMIX_REQUIRED .....	236
5.2.2.1.15. BSWM_SID_CANSM_CURRENT_STATE .....	236
5.2.2.1.16. BSWM_SID_COMM_CURRENT_MODE .....	236
5.2.2.1.17. BSWM_SID_COMM_CURRENT_PNC_MODE .....	237
5.2.2.1.18. BSWM_SID_COMM_INITIATE_RESET .....	237
5.2.2.1.19. BSWM_SID_DCM_APPLICATION_UPDATED .....	237



5.2.2.1.20. BSWM_SID_DCM_COMMUNICATIONMODE_CURRENT_STATE .....	237
5.2.2.1.21. BSWM_SID_DCM_REQUEST_COMMUNICATION_MODE .....	237
5.2.2.1.22. BSWM_SID_DCM_REQUEST_RESET_MODE .....	237
5.2.2.1.23. BSWM_SID_DCM_REQUEST_SESSION_MODE .....	238
5.2.2.1.24. BSWM_SID_DEINIT .....	238
5.2.2.1.25. BSWM_SID_ECUM_CURRENT_STATE .....	238
5.2.2.1.26. BSWM_SID_ECUM_CURRENT_WAKEUP .....	238
5.2.2.1.27. BSWM_SID_ETHIF_PORTGROUPLINKSTATECHG .....	238
5.2.2.1.28. BSWM_SID_ETHSM_CURRENT_STATE .....	239
5.2.2.1.29. BSWM_SID_FRSM_CURRENT_STATE .....	239
5.2.2.1.30. BSWM_SID_GET_VERSION_INFO .....	239
5.2.2.1.31. BSWM_SID_INIT .....	239
5.2.2.1.32. BSWM_SID_INVALID .....	239
5.2.2.1.33. BSWM_SID_J1939DCMBROADCASTSTATUS .....	239
5.2.2.1.34. BSWM_SID_J1939NM_STATECHANGENOTIFICATION .....	240
5.2.2.1.35. BSWM_SID_LINSM_CURRENT_SCHEDULE .....	240
5.2.2.1.36. BSWM_SID_LINSM_CURRENT_STATE .....	240
5.2.2.1.37. BSWM_SID_LINTP_REQUESTMODE .....	240
5.2.2.1.38. BSWM_SID_MAINFUNCTION .....	240
5.2.2.1.39. BSWM_SID_NM_CAR_WAKEUP_INDICATION .....	240
5.2.2.1.40. BSWM_SID_NVM_CURRENT_BLOCK_MODE .....	241
5.2.2.1.41. BSWM_SID_NVM_CURRENT_JOB_MODE .....	241
5.2.2.1.42. BSWM_SID_REQUESTMODE .....	241
5.2.2.1.43. BSWM_SID_SD_CLIENTSERVICE_CURRENTSTATE .....	241
5.2.2.1.44. BSWM_SID_SD_CONSUMEDEVENTGROUP_CURRENTSTATE .....	241
5.2.2.1.45. BSWM_SID_SD_EVENTHANDLER_CURRENTSTATE .....	242
5.2.2.1.46. BSWM_SID_TRIGGER_SLAVE_RTE_STOP .....	242
5.2.2.1.47. BSWM_SID_TRIGGER_STARTUP_PHASE2 .....	242
5.2.2.1.48. BSWM_SID_WDGM_REQUEST_PARTITION_RESET .....	242
5.2.2.1.49. BSWM_SW_MAJOR_VERSION .....	242
5.2.2.1.50. BSWM_SW_MINOR_VERSION .....	242
5.2.2.1.51. BSWM_SW_PATCH_VERSION .....	243
5.2.2.1.52. BSWM_VENDOR_ID .....	243
5.2.2.2. Functions .....	243
5.2.2.2.1. BswM_Deinit .....	243
5.2.2.2.2. BswM_GetVersionInfo .....	243
5.2.2.2.3. BswM_Init .....	244
5.2.2.2.4. BswM_IsValidConfig .....	244
5.2.2.2.5. BswM_RequestMode .....	245
5.2.3. Integration notes .....	245
5.2.3.1. Exclusive areas .....	245
5.2.3.1.1. SCHM_BSWM_EXCLUSIVE_AREA .....	245



5.2.3.2. Production errors .....	246
5.2.3.3. Memory mapping .....	246
5.2.3.4. Integration requirements .....	247
5.2.3.4.1. intgr.BswM.EB_INTREQ_BswM_0001 .....	247
5.2.3.4.2. intgr.BswM.EB_INTREQ_BswM_0002 .....	247
5.2.3.4.3. intgr.BswM.EB_INTREQ_BswM_0003 .....	247
5.2.3.4.4. intgr.BswM.EB_INTREQ_BswM_0004 .....	247
5.2.3.4.5. intgr.BswM.EB_INTREQ_BswM_0005 .....	248
5.2.3.4.6. intgr.BswM.EB_INTREQ_BswM_BswMPduGroupSwitch_BswMDeadlineMonitoringControl .....	248
5.2.3.4.7. intgr.BswM.EB_INTREQ_BswMLogicalExpressionsOptimization .....	248
5.2.3.4.8. intgr.BswM.EB_INTREQ_BswMActionsOptimization .....	249
5.2.3.4.9. intgr.BswM.EB_INTREQ_BswMSchMSwitchPostbuild .....	249
5.3. ComM .....	249
5.3.1. Configuration parameters .....	249
5.3.1.1. ComMDefensiveProgramming .....	250
5.3.1.2. ComMConfigSet .....	253
5.3.1.3. ComMChannel .....	254
5.3.1.4. ComMNetworkManagement .....	258
5.3.1.5. ComMUUserPerChannel .....	260
5.3.1.6. ComMPnc .....	260
5.3.1.7. ComMPncComSignal .....	262
5.3.1.8. ComMUser .....	263
5.3.1.9. ComMGeneral .....	264
5.3.1.10. ComMServiceAPI .....	274
5.3.1.11. CommonPublishedInformation .....	275
5.3.1.12. PublishedInformation .....	278
5.3.2. Application programming interface (API) .....	279
5.3.2.1. Type definitions .....	279
5.3.2.1.1. ComM_ASR32_InhibitionStatusType .....	279
5.3.2.1.2. ComM_ASR32_ModeType .....	279
5.3.2.1.3. ComM_ASR32_UserHandleType .....	279
5.3.2.1.4. ComM_ASR40_InhibitionStatusType .....	280
5.3.2.1.5. ComM_ASR40_ModeType .....	280
5.3.2.1.6. ComM_ASR40_UserHandleType .....	280
5.3.2.1.7. ComM_BusSM_RequestComMode_FctPtr .....	280
5.3.2.1.8. ComM_Dcm_Notify_FctPtr .....	280
5.3.2.1.9. ComM_Dcm_Notify_Type .....	281
5.3.2.1.10. ComM_InhibitionStatusType .....	281
5.3.2.1.11. ComM_InitStatusType .....	281
5.3.2.1.12. ComM_ModeType .....	281



5.3.2.1.13. ComM_Multicore_SchmSend_FctPtr .....	281
5.3.2.1.14. ComM_NvDataType .....	282
5.3.2.1.15. ComM_PncModeType .....	282
5.3.2.1.16. ComM_RxSignal_Struct_Type .....	282
5.3.2.1.17. ComM_StateType .....	282
5.3.2.1.18. ComM_TxSignal_Struct_Type .....	283
5.3.2.1.19. ComM_UserHandleType .....	283
5.3.2.2. Macro constants .....	283
5.3.2.2.1. COMM_AR_RELEASE_MAJOR_VERSION .....	283
5.3.2.2.2. COMM_AR_RELEASE_MINOR_VERSION .....	283
5.3.2.2.3. COMM_AR_RELEASE_REVISION_VERSION .....	283
5.3.2.2.4. COMM_BUS_TYPE_CAN .....	284
5.3.2.2.5. COMM_BUS_TYPE_ETH .....	284
5.3.2.2.6. COMM_BUS_TYPE_FR .....	284
5.3.2.2.7. COMM_BUS_TYPE_INTERNAL .....	284
5.3.2.2.8. COMM_BUS_TYPE_LIN .....	284
5.3.2.2.9. COMM_DET_REPORT_ERROR .....	284
5.3.2.2.10. COMM_EIRA_ERA_ACTIVE .....	285
5.3.2.2.11. COMM_EIRA_PASSIVE .....	285
5.3.2.2.12. COMM_E_BUSSM_REQUESTCOMMODE .....	285
5.3.2.2.13. COMM_E_DCM_COMM_FULLCOMMODEENTERED .....	285
5.3.2.2.14. COMM_E_DCM_COMM_NOCOMMODEENTERED .....	285
5.3.2.2.15. COMM_E_DCM_COMM_SILENTCOMMODEENTERED .....	285
5.3.2.2.16. COMM_E_MASTER_DATA_GET .....	285
5.3.2.2.17. COMM_E_MASTER_DATA_SET .....	286
5.3.2.2.18. COMM_E_MODE_LIMITATION .....	286
5.3.2.2.19. COMM_E_MULTIPLE_PNC_ASSIGNED .....	286
5.3.2.2.20. COMM_E_NOT_INITED .....	286
5.3.2.2.21. COMM_E_NO_PNC_ASSIGNED .....	286
5.3.2.2.22. COMM_E_SLAVE_DATA_GET .....	286
5.3.2.2.23. COMM_E_SLAVE_DATA_SET .....	287
5.3.2.2.24. COMM_E_UNINIT .....	287
5.3.2.2.25. COMM_E_WRONG_PARAMETERS .....	287
5.3.2.2.26. COMM_FULL_COMMUNICATION .....	287
5.3.2.2.27. COMM_FULL_COM_NETWORK_REQUESTED .....	287
5.3.2.2.28. COMM_FULL_COM_READY_SLEEP .....	287
5.3.2.2.29. COMM_INIT .....	288
5.3.2.2.30. COMM_INSTANCE_ID .....	288
5.3.2.2.31. COMM_LIMIT_NOCOMM .....	288
5.3.2.2.32. COMM_LIMIT_NOCOMM_MASK .....	288
5.3.2.2.33. COMM_MODULE_ID .....	288
5.3.2.2.34. COMM_NM_FULL_VARIANT .....	288



5.3.2.2.35. COMM_NM_LIGHT_VARIANT .....	289
5.3.2.2.36. COMM_NM_NONE_VARIANT .....	289
5.3.2.2.37. COMM_NM_PASSIVE_VARIANT .....	289
5.3.2.2.38. COMM_NM_SLAVE_ACTIVE_VARIANT .....	289
5.3.2.2.39. COMM_NM_SLAVE_PASSIVE_VARIANT .....	289
5.3.2.2.40. COMM_NO_COMMUNICATION .....	289
5.3.2.2.41. COMM_NO_COM_NO_PENDING_REQUEST .....	289
5.3.2.2.42. COMM_NO_COM_REQUEST_PENDING .....	290
5.3.2.2.43. COMM_NO_INTERNAL_AND_EXTERNAL_REQUESTS .....	290
5.3.2.2.44. COMM_NO_INTERNAL_REQUESTS .....	290
5.3.2.2.45. COMM_NUM_CHANNELS_BYTES .....	290
5.3.2.2.46. COMM_NUM_PNC_BYTES .....	290
5.3.2.2.47. COMM_PREVENT_WAKEUP .....	290
5.3.2.2.48. COMM_PREVENT_WAKEUP_MASK .....	291
5.3.2.2.49. COMM_SID_BUS_SM_BUS_SLEEP_MODE .....	291
5.3.2.2.50. COMM_SID_BUS_SM_INDICATION .....	291
5.3.2.2.51. COMM_SID_COMMUNICATIONALLOWED .....	291
5.3.2.2.52. COMM_SID_DCM_ACTIVE_DIAGNOSTIC .....	291
5.3.2.2.53. COMM_SID_DCM_INACTIVE_DIAGNOSTIC .....	291
5.3.2.2.54. COMM_SID_DEINIT .....	292
5.3.2.2.55. COMM_SID_ECUM_PNC_WAKE_UP_INDICATION .....	292
5.3.2.2.56. COMM_SID_ECUM_WAKE_UP_INDICATION .....	292
5.3.2.2.57. COMM_SID_GET_CURRENT_COM_MODE .....	292
5.3.2.2.58. COMM_SID_GET_CURRENT_PNC_COM_MODE .....	292
5.3.2.2.59. COMM_SID_GET_INHIBITION_STATUS .....	293
5.3.2.2.60. COMM_SID_GET_MAX_COM_MODE .....	293
5.3.2.2.61. COMM_SID_GET_REQUESTED_COM_MODE .....	293
5.3.2.2.62. COMM_SID_GET_STATE .....	293
5.3.2.2.63. COMM_SID_GET_STATUS .....	293
5.3.2.2.64. COMM_SID_GET_VERSION_INFO .....	293
5.3.2.2.65. COMM_SID_INIT .....	294
5.3.2.2.66. COMM_SID_LIMIT_CHANNEL_TO_NO_COM .....	294
5.3.2.2.67. COMM_SID_LIMIT_ECU_TO_NO_COM .....	294
5.3.2.2.68. COMM_SID_MAIN_FUNCTION .....	294
5.3.2.2.69. COMM_SID_MASTER_DATA_GET .....	294
5.3.2.2.70. COMM_SID_MASTER_DATA_SET .....	295
5.3.2.2.71. COMM_SID_NM_BUS_SLEEP_MODE .....	295
5.3.2.2.72. COMM_SID_NM_NETWORK_MODE .....	295
5.3.2.2.73. COMM_SID_NM_NETWORK_START_IND .....	295
5.3.2.2.74. COMM_SID_NM_PREPARE_BUS_SLEEP .....	295
5.3.2.2.75. COMM_SID_NM_RESTART_IND .....	295
5.3.2.2.76. COMM_SID_PREVENT_WAKE_UP .....	296



5.3.2.2.77. COMM_SID_READ_INHIBIT_COUNTER .....	296
5.3.2.2.78. COMM_SID_REQUEST_COM_MODE .....	296
5.3.2.2.79. COMM_SID_RESET_INHIBIT_COUNTER .....	296
5.3.2.2.80. COMM_SID_SET_ECU_GROUP_CLASS .....	296
5.3.2.2.81. COMM_SID_SLAVE_DATA_GET .....	297
5.3.2.2.82. COMM_SID_SLAVE_DATA_SET .....	297
5.3.2.2.83. COMM_SILENT_COM .....	297
5.3.2.2.84. COMM_SILENT_COMMUNICATION .....	297
5.3.2.2.85. COMM_SW_MAJOR_VERSION .....	297
5.3.2.2.86. COMM_SW_MINOR_VERSION .....	297
5.3.2.2.87. COMM_SW_PATCH_VERSION .....	298
5.3.2.2.88. COMM_UNINIT .....	298
5.3.2.2.89. COMM_VENDOR_ID .....	298
5.3.2.2.90. ComM_GetCurrentComMode .....	298
5.3.2.2.91. ComM_GetCurrentPNCComMode .....	298
5.3.2.2.92. ComM_GetInhibitionStatus .....	298
5.3.2.2.93. ComM_GetMaxComMode .....	299
5.3.2.2.94. ComM_GetRequestedComMode .....	299
5.3.2.2.95. ComM_LimitChannelToNoComMode .....	299
5.3.2.2.96. ComM_LimitECUToNoComMode .....	299
5.3.2.2.97. ComM_PreventWakeUp .....	299
5.3.2.2.98. ComM_ReadInhibitCounter .....	299
5.3.2.2.99. ComM_RequestComMode .....	300
5.3.2.2.100. ComM_ResetInhibitCounter .....	300
5.3.2.2.101. ComM_SetECUGroupClassification .....	300
5.3.2.2.102. PNC_FULL_COMMUNICATION .....	300
5.3.2.2.103. PNC_NO_COMMUNICATION .....	300
5.3.2.2.104. PNC_PREPARE_SLEEP .....	300
5.3.2.2.105. PNC_READY_SLEEP .....	301
5.3.2.2.106. PNC_REQUESTED .....	301
5.3.2.3. Objects .....	301
5.3.2.3.1. ComM_BusSmRequestComModeMcFpList .....	301
5.3.2.3.2. ComM_BusTypeOfChannel .....	301
5.3.2.3.3. ComM_ChanState .....	301
5.3.2.3.4. ComM_ChannelInhibitionStatusInit .....	301
5.3.2.3.5. ComM_ChannelMode .....	302
5.3.2.3.6. ComM_ChannelModePrevious .....	302
5.3.2.3.7. ComM_Communication_Allowed .....	302
5.3.2.3.8. ComM_DcmActiveStatus .....	302
5.3.2.3.9. ComM_Dcm_Notify_FpList .....	302
5.3.2.3.10. ComM_EB_NoWakeupsNvStorage .....	302
5.3.2.3.11. ComM_EB_NvRAM .....	303



5.3.2.3.12. ComM_EB_NvROM .....	303
5.3.2.3.13. ComM_InitStatus .....	303
5.3.2.3.14. ComM_NetReqNoNmTimeoutMs .....	303
5.3.2.3.15. ComM_NmVariantOfChannel .....	303
5.3.2.3.16. ComM_PNCMode .....	304
5.3.2.3.17. ComM_PNCRequestedComMode .....	304
5.3.2.3.18. ComM_PNCRequestedComMode0PncVectorAvoidance .....	304
5.3.2.3.19. ComM_PassiveStartupStatus .....	304
5.3.2.3.20. ComM_PncRxActiveEraMask .....	304
5.3.2.3.21. ComM_PncRxEiraMask .....	305
5.3.2.3.22. ComM_PncRxUnfilteredEraMask .....	305
5.3.2.3.23. ComM_PncStateGWE .....	305
5.3.2.3.24. ComM_PncZeroVectorAvoidanceRelease .....	305
5.3.2.3.25. ComM_ReadySleepNoNmTimeoutMs .....	305
5.3.2.3.26. ComM_RequestedComMode .....	306
5.3.2.3.27. ComM_RequestedComModeWithoutPnc .....	306
5.3.2.3.28. ComM_RxActiveEra .....	306
5.3.2.3.29. ComM_RxActiveEraSignalStatus .....	306
5.3.2.3.30. ComM_RxComSignalActiveEraCfg .....	306
5.3.2.3.31. ComM_RxComSignalEiraCfg .....	307
5.3.2.3.32. ComM_RxComSignalUnfilteredEraCfg .....	307
5.3.2.3.33. ComM_RxEira .....	307
5.3.2.3.34. ComM_RxEiraSignalStatus .....	307
5.3.2.3.35. ComM_RxUnfilteredEra .....	307
5.3.2.3.36. ComM_RxUnfilteredEraSignalStatus .....	307
5.3.2.3.37. ComM_StoreNoWakeUpInhibition .....	308
5.3.2.3.38. ComM_TxEiraEraActive .....	308
5.3.2.3.39. ComM_TxEraPassive .....	308
5.3.2.3.40. ComM_UserRequestedComMode .....	308
5.3.2.4. Functions .....	308
5.3.2.4.1. ComM_ASR32_GetCurrentComMode .....	308
5.3.2.4.2. ComM_ASR32_GetInhibitionStatus .....	309
5.3.2.4.3. ComM_ASR32_GetMaxComMode .....	309
5.3.2.4.4. ComM_ASR32_GetRequestedMode .....	310
5.3.2.4.5. ComM_ASR32_LimitChannelToNoComMode .....	310
5.3.2.4.6. ComM_ASR32_LimitECUToNoComMode .....	311
5.3.2.4.7. ComM_ASR32_PreventWakeUp .....	311
5.3.2.4.8. ComM_ASR32_ReadInhibitCounter .....	312
5.3.2.4.9. ComM_ASR32_RequestComMode .....	312
5.3.2.4.10. ComM_ASR32_ResetInhibitCounter .....	313
5.3.2.4.11. ComM_ASR32_SetECUGroupClassification .....	313
5.3.2.4.12. ComM_ASR40_GetCurrentComMode .....	314



5.3.2.4.13. ComM_ASR40_GetCurrentPNCComMode .....	314
5.3.2.4.14. ComM_ASR40_GetInhibitionStatus .....	315
5.3.2.4.15. ComM_ASR40_GetMaxComMode .....	315
5.3.2.4.16. ComM_ASR40_GetRequestedComMode .....	316
5.3.2.4.17. ComM_ASR40_LimitChannelToNoComMode .....	316
5.3.2.4.18. ComM_ASR40_LimitECUToNoComMode .....	317
5.3.2.4.19. ComM_ASR40_PreventWakeUp .....	317
5.3.2.4.20. ComM_ASR40_ReadInhibitCounter .....	318
5.3.2.4.21. ComM_ASR40_RequestComMode .....	318
5.3.2.4.22. ComM_ASR40_ResetInhibitCounter .....	319
5.3.2.4.23. ComM_ASR40_SetECUGroupClassification .....	319
5.3.2.4.24. ComM_BitArrayClear .....	320
5.3.2.4.25. ComM_BitArrayIsBitSet .....	320
5.3.2.4.26. ComM_BitArraySet .....	320
5.3.2.4.27. ComM_BusSM_BusSleepMode .....	320
5.3.2.4.28. ComM_BusSM_ModelIndication .....	321
5.3.2.4.29. ComM_ChannelNoComInhibitionStatus .....	321
5.3.2.4.30. ComM_ChannelWakeUpInhibitionStatus .....	322
5.3.2.4.31. ComM_CommunicationAllowed .....	322
5.3.2.4.32. ComM_DCM_ActiveDiagnostic .....	322
5.3.2.4.33. ComM_DCM_InactiveDiagnostic .....	323
5.3.2.4.34. ComM_DelInit .....	323
5.3.2.4.35. ComM_EcuM_PNCWakeUpIndication .....	323
5.3.2.4.36. ComM_EcuM_WakeUpIndication .....	324
5.3.2.4.37. ComM_EvaluatePnc .....	324
5.3.2.4.38. ComM_GetState .....	324
5.3.2.4.39. ComM_GetStatus .....	325
5.3.2.4.40. ComM_GetVersionInfo .....	325
5.3.2.4.41. ComM_Init .....	326
5.3.2.4.42. ComM_IsValidConfig .....	326
5.3.2.4.43. ComM_Nm_BusSleepMode .....	327
5.3.2.4.44. ComM_Nm_NetworkMode .....	327
5.3.2.4.45. ComM_Nm_NetworkStartIndication .....	328
5.3.2.4.46. ComM_Nm_PreparesBusSleepMode .....	328
5.3.2.4.47. ComM_Nm_RestartIndication .....	328
5.3.2.4.48. ComM_PncMainFunction .....	329
5.3.2.4.49. ComM_PncProcessTimers .....	329
5.3.2.4.50. ComM_ProcessTimers .....	329
5.3.2.4.51. ComM_SetChannelComMInhibitionStatus .....	329
5.3.3. Integration notes .....	330
5.3.3.1. Exclusive areas .....	330
5.3.3.1.1. SCHM_COMM_EXCLUSIVE_AREA_0 .....	330



5.3.3.2. Production errors .....	330
5.3.3.3. Memory mapping .....	330
5.3.3.4. Integration requirements .....	331
5.4. EcuM .....	331
5.4.1. Configuration parameters .....	331
5.4.1.1. CommonPublishedInformation .....	332
5.4.1.2. EcuMConfiguration .....	335
5.4.1.3. EcuMCommonConfiguration .....	336
5.4.1.4. EcuMDefaultShutdownTarget .....	339
5.4.1.5. EcuMDriverInitListOne .....	341
5.4.1.6. EcuMDriverInitItem .....	341
5.4.1.7. EcuMDriverInitListZero .....	344
5.4.1.8. EcuMDriverInitItem .....	344
5.4.1.9. EcuMDriverRestartList .....	346
5.4.1.10. EcuMDriverInitItem .....	346
5.4.1.11. EcuMSleepMode .....	349
5.4.1.12. EcuMWakeupSource .....	350
5.4.1.13. EcuMDemEventParameterRefs .....	352
5.4.1.14. EcuMFixedConfiguration .....	354
5.4.1.15. EcuMDriverInitListThree .....	358
5.4.1.16. EcuMDriverInitItem .....	358
5.4.1.17. EcuMDriverInitListTwo .....	359
5.4.1.18. EcuMDriverInitItem .....	360
5.4.1.19. EcuMFixedUserConfig .....	361
5.4.1.20. EcuMTTII .....	361
5.4.1.21. EcuMWdgM .....	363
5.4.1.22. EcuMFlexConfiguration .....	365
5.4.1.23. EcuMAlarmClock .....	367
5.4.1.24. EcuMFlexUserConfig .....	368
5.4.1.25. EcuMGoDownAllowedUsers .....	369
5.4.1.26. EcuMResetMode .....	370
5.4.1.27. EcuMSetClockAllowedUsers .....	370
5.4.1.28. EcuMShutdownCause .....	370
5.4.1.29. EcuMShutdownTarget .....	371
5.4.1.30. EcuMDefensiveProgramming .....	371
5.4.1.31. EcuMFixedGeneral .....	374
5.4.1.32. EcuMFlexGeneral .....	377
5.4.1.33. EcuMGeneral .....	379
5.4.1.34. EcuMServiceAPI .....	385
5.4.1.35. ReportToDem .....	386
5.4.1.36. PublishedInformation .....	389
5.4.2. Recommended configurations .....	389

5.4.2.1. EcuMRecConfiguration .....	389
5.4.2.1.1. EcuM_Config_0 .....	390
5.4.2.1.2. EcuMFlexConfiguration .....	390
5.4.2.1.3. ECUM_CAUSE_UNKNOWN .....	390
5.4.2.1.4. ECUM_CAUSE_ECU_STATE .....	391
5.4.2.1.5. ECUM_CAUSE_WDGM .....	391
5.4.2.1.6. ECUM_CAUSE_DCM .....	391
5.4.2.1.7. ECUM_RESET MCU .....	391
5.4.2.1.8. ECUM_RESET_WDG .....	391
5.4.2.1.9. ECUM_RESET_IO .....	391
5.4.2.1.10. ECUM_STATE_SLEEP .....	392
5.4.2.1.11. ECUM_STATE_OFF .....	392
5.4.2.1.12. ECUM_STATE_RESET .....	392
5.4.2.1.13. EcuMCommonConfiguration .....	392
5.4.2.1.14. ECUM_WKSOURCE_POWER .....	393
5.4.2.1.15. ECUM_WKSOURCE_RESET .....	393
5.4.2.1.16. ECUM_WKSOURCE_INTERNAL_RESET .....	393
5.4.2.1.17. ECUM_WKSOURCE_INTERNAL_WDG .....	393
5.4.2.1.18. ECUM_WKSOURCE_EXTERNAL_WDG .....	393
5.4.3. Application programming interface (API) .....	394
5.4.3.1. Type definitions .....	394
5.4.3.1.1. EcuM_ASR32_BootTargetType .....	394
5.4.3.1.2. EcuM_ASR32_StateType .....	394
5.4.3.1.3. EcuM_ASR32_UserType .....	394
5.4.3.1.4. EcuM_ASR40_BootTargetType .....	395
5.4.3.1.5. EcuM_ASR40_ShutdownCauseType .....	395
5.4.3.1.6. EcuM_ASR40_StateType .....	395
5.4.3.1.7. EcuM_ASR40_UserType .....	395
5.4.3.1.8. EcuM_BootTargetInternalType .....	395
5.4.3.1.9. EcuM_BootTargetType .....	396
5.4.3.1.10. EcuM_DeterminePbConfigurationRetType .....	396
5.4.3.1.11. EcuM_ResetType .....	396
5.4.3.1.12. EcuM_ShutdownCauseType .....	396
5.4.3.1.13. EcuM_SleepModeConfigType .....	397
5.4.3.1.14. EcuM_StateType .....	397
5.4.3.1.15. EcuM_SyncActionType .....	397
5.4.3.1.16. EcuM_UserType .....	397
5.4.3.1.17. EcuM_WakeupReactionType .....	398
5.4.3.1.18. EcuM_WakeupSourceType .....	398
5.4.3.1.19. EcuM_WakeupStatusType .....	398
5.4.3.2. Macro constants .....	398
5.4.3.2.1. ECUM_ABORT_SYNC .....	398



5.4.3.2.2. ECUM_AR_RELEASE_MAJOR_VERSION .....	399
5.4.3.2.3. ECUM_AR_RELEASE_MINOR_VERSION .....	399
5.4.3.2.4. ECUM_AR_RELEASE_REVISION_VERSION .....	399
5.4.3.2.5. ECUM_BOOT_TARGET_APP .....	399
5.4.3.2.6. ECUM_BOOT_TARGET_BOOTLOADER .....	399
5.4.3.2.7. ECUM_BOOT_TARGET_OEM_BOOTLOADER .....	399
5.4.3.2.8. ECUM_BOOT_TARGET_OEM_BOOTLOADER_INTERNAL_0 .....	400
5.4.3.2.9. ECUM_BOOT_TARGET_OEM_BOOTLOADER_INTERNAL_1 .....	400
5.4.3.2.10. ECUM_BOOT_TARGET_SYS_BOOTLOADER .....	400
5.4.3.2.11. ECUM_BOOT_TARGET_SYS_BOOTLOADER_INTERNAL_0 .....	400
5.4.3.2.12. ECUM_BOOT_TARGET_SYS_BOOTLOADER_INTERNAL_1 .....	400
5.4.3.2.13. ECUM_CAUSE_DCM .....	401
5.4.3.2.14. ECUM_CAUSE_ECU_STATE .....	401
5.4.3.2.15. ECUM_CAUSE_UNKNOWN .....	401
5.4.3.2.16. ECUM_CAUSE_WDGM .....	401
5.4.3.2.17. ECUM_CONTINUE_SYNC .....	401
5.4.3.2.18. ECUM_E_COMMPCWAKEUPINDICATION .....	401
5.4.3.2.19. ECUM_E_COMMWAKEUPINDICATION .....	402
5.4.3.2.20. ECUM_E_INVALID_PAR .....	402
5.4.3.2.21. ECUM_E_MISMATCHED_RUN_RELEASE .....	402
5.4.3.2.22. ECUM_E_MULTIPLE_RUN_REQUESTS .....	402
5.4.3.2.23. ECUM_E_NULL_POINTER .....	402
5.4.3.2.24. ECUM_E_PARAM_POINTER .....	403
5.4.3.2.25. ECUM_E RTE_SWITCH_FAILED .....	403
5.4.3.2.26. ECUM_E_SERVICE_DISABLED .....	403
5.4.3.2.27. ECUM_E_STATE_PAR_OUT_OF_RANGE .....	403
5.4.3.2.28. ECUM_E_UNINIT .....	403
5.4.3.2.29. ECUM_E_UNKNOWN_WAKEUP_SOURCE .....	404
5.4.3.2.30. ECUM_E_UNSUCCESSFUL_CORE_START .....	404
5.4.3.2.31. ECUM_E_WRONG_CONFIG_PARAM .....	404
5.4.3.2.32. ECUM_E_WRONG_CONTEXT .....	404
5.4.3.2.33. ECUM_MODULE_ID .....	404
5.4.3.2.34. ECUM_RAM_CHECK_FAILED .....	405
5.4.3.2.35. ECUM_SID_CLEAR_WK_EV .....	405
5.4.3.2.36. ECUM_SID_DOWN .....	405
5.4.3.2.37. ECUM_SID_GET_BOOT_TAR .....	405
5.4.3.2.38. ECUM_SID_GET_EXP_WK_EV .....	405
5.4.3.2.39. ECUM_SID_GET_LAST_SHUT_TAR .....	405
5.4.3.2.40. ECUM_SID_GET_NXT_RCNT_SHUT_CAUSE .....	406
5.4.3.2.41. ECUM_SID_GET_PEND_WK_EV .....	406
5.4.3.2.42. ECUM_SID_GET_RCNT_SHUT_CAUSE .....	406
5.4.3.2.43. ECUM_SID_GET_SHUT_CAUSE .....	406



5.4.3.2.44. ECUM_SID_GET_SHUT_TAR .....	406
5.4.3.2.45. ECUM_SID_GET_STAT_OF_WK_SRC .....	407
5.4.3.2.46. ECUM_SID_GET_VALID_WK_EV .....	407
5.4.3.2.47. ECUM_SID_GET_VERSION_INFO .....	407
5.4.3.2.48. ECUM_SID_GO_RUN .....	407
5.4.3.2.49. ECUM_SID_HALT .....	407
5.4.3.2.50. ECUM_SID_INIT .....	408
5.4.3.2.51. ECUM_SID_KILL_ALL_RUN_REQUESTS .....	408
5.4.3.2.52. ECUM_SID_MAIN_FUNCTION .....	408
5.4.3.2.53. ECUM_SID_POLL .....	408
5.4.3.2.54. ECUM_SID_RELEASE_POST_RUN .....	408
5.4.3.2.55. ECUM_SID_RELEASE_RUN .....	408
5.4.3.2.56. ECUM_SID_REQUEST_POST_RUN .....	409
5.4.3.2.57. ECUM_SID_REQUEST_RUN .....	409
5.4.3.2.58. ECUM_SID_SEL_BOOT_TAR .....	409
5.4.3.2.59. ECUM_SID_SEL_SHUT_CAUSE .....	409
5.4.3.2.60. ECUM_SID_SEL_SHUT_TAR .....	409
5.4.3.2.61. ECUM_SID_SET_WK_EV .....	410
5.4.3.2.62. ECUM_SID_SHUTDOWN .....	410
5.4.3.2.63. ECUM_SID_STARTUP_TWO .....	410
5.4.3.2.64. ECUM_SID_VALIDATE_WK_EV .....	410
5.4.3.2.65. ECUM_STATE_APP_POST_RUN .....	410
5.4.3.2.66. ECUM_STATE_APP_RUN .....	411
5.4.3.2.67. ECUM_STATE_GO_OFF_ONE .....	411
5.4.3.2.68. ECUM_STATE_GO_OFF_TWO .....	411
5.4.3.2.69. ECUM_STATE_GO_SLEEP .....	411
5.4.3.2.70. ECUM_STATE_OFF .....	411
5.4.3.2.71. ECUM_STATE_PREP_SHUTDOWN .....	411
5.4.3.2.72. ECUM_STATE_RESET .....	412
5.4.3.2.73. ECUM_STATE_RUN .....	412
5.4.3.2.74. ECUM_STATE_SHUTDOWN .....	412
5.4.3.2.75. ECUM_STATE_SLEEP .....	412
5.4.3.2.76. ECUM_STATE_STARTUP .....	412
5.4.3.2.77. ECUM_STATE_STARTUP_ONE .....	413
5.4.3.2.78. ECUM_STATE_STARTUP_TWO .....	413
5.4.3.2.79. ECUM_STATE_WAKEUP .....	413
5.4.3.2.80. ECUM_STATE_WAKEUP_ONE .....	413
5.4.3.2.81. ECUM_STATE_WAKEUPREACTION .....	413
5.4.3.2.82. ECUM_STATE_WAKEUP_TTII .....	414
5.4.3.2.83. ECUM_STATE_WAKEUP_TWO .....	414
5.4.3.2.84. ECUM_STATE_WAKEUP_VALIDATION .....	414
5.4.3.2.85. ECUM_STATE_WAKEUP_WAKESLEEP .....	414



5.4.3.2.86. ECUM_SUBSTATE_MASK .....	414
5.4.3.2.87. ECUM_SUPERSTATE_MASK .....	414
5.4.3.2.88. ECUM_SW_MAJOR_VERSION .....	415
5.4.3.2.89. ECUM_SW_MINOR_VERSION .....	415
5.4.3.2.90. ECUM_SW_PATCH_VERSION .....	415
5.4.3.2.91. ECUM_VAR_BOOTTARGET .....	415
5.4.3.2.92. ECUM_VENDOR_ID .....	415
5.4.3.2.93. ECUM_WKACT_RUN .....	415
5.4.3.2.94. ECUM_WKACT_SHUTDOWN .....	416
5.4.3.2.95. ECUM_WKACT_TTII .....	416
5.4.3.2.96. ECUM_WKSOURCE_ALL_SOURCES .....	416
5.4.3.2.97. ECUM_WKSTATUS_DISABLED .....	416
5.4.3.2.98. ECUM_WKSTATUS_ENABLED .....	416
5.4.3.2.99. ECUM_WKSTATUS_EXPIRED .....	416
5.4.3.2.100. ECUM_WKSTATUS_NONE .....	417
5.4.3.2.101. ECUM_WKSTATUS_PENDING .....	417
5.4.3.2.102. ECUM_WKSTATUS_VALIDATED .....	417
5.4.3.2.103. EcuM_GetBootTarget .....	417
5.4.3.2.104. EcuM_GetLastShutdownTarget .....	417
5.4.3.2.105. EcuM_GetMostRecentShutdown .....	418
5.4.3.2.106. EcuM_GetNextRecentShutdown .....	418
5.4.3.2.107. EcuM_GetShutdownCause .....	418
5.4.3.2.108. EcuM_GetShutdownTarget .....	418
5.4.3.2.109. EcuM_SelectBootTarget .....	418
5.4.3.2.110. EcuM_SelectShutdownCause .....	419
5.4.3.3. Objects .....	419
5.4.3.3.1. EcuM_BootTarget .....	419
5.4.3.4. Functions .....	419
5.4.3.4.1. EcuM_AL_DriverInitOne .....	419
5.4.3.4.2. EcuM_AL_DriverInitZero .....	419
5.4.3.4.3. EcuM_AL_DriverRestart .....	420
5.4.3.4.4. EcuM_AL_Reset .....	420
5.4.3.4.5. EcuM_AL_SetProgrammableInterrupts .....	420
5.4.3.4.6. EcuM_AL_SwitchOff .....	421
5.4.3.4.7. EcuM_AS32_GetBootTarget .....	421
5.4.3.4.8. EcuM_AS32_GetLastShutdownTarget .....	421
5.4.3.4.9. EcuM_AS32_GetShutdownTarget .....	422
5.4.3.4.10. EcuM_AS32_ReleasePOST_RUN .....	423
5.4.3.4.11. EcuM_AS32_ReleaseRUN .....	424
5.4.3.4.12. EcuM_AS32_RequestPOST_RUN .....	424
5.4.3.4.13. EcuM_AS32_RequestRUN .....	425
5.4.3.4.14. EcuM_AS32_SelectBootTarget .....	425



5.4.3.4.15. EcuM_ASR32_SelectShutdownTarget .....	426
5.4.3.4.16. EcuM_ASR40_GetBootTarget .....	427
5.4.3.4.17. EcuM_ASR40_GetLastShutdownTarget .....	427
5.4.3.4.18. EcuM_ASR40_GetMostRecentShutdown .....	428
5.4.3.4.19. EcuM_ASR40_GetNextRecentShutdown .....	429
5.4.3.4.20. EcuM_ASR40_GetShutdownCause .....	430
5.4.3.4.21. EcuM_ASR40_GetShutdownTarget .....	430
5.4.3.4.22. EcuM_ASR40_ReleasePOST_RUN .....	431
5.4.3.4.23. EcuM_ASR40_ReleaseRUN .....	432
5.4.3.4.24. EcuM_ASR40_RequestPOST_RUN .....	432
5.4.3.4.25. EcuM_ASR40_RequestRUN .....	433
5.4.3.4.26. EcuM_ASR40_SelectBootTarget .....	433
5.4.3.4.27. EcuM_ASR40_SelectShutdownCause .....	434
5.4.3.4.28. EcuM_CheckRamHash .....	434
5.4.3.4.29. EcuM_CheckValidation .....	435
5.4.3.4.30. EcuM_CheckWakeup .....	435
5.4.3.4.31. EcuM_ClearWakeupEvent .....	436
5.4.3.4.32. EcuM_DefaultInitListOne .....	436
5.4.3.4.33. EcuM_DefaultInitListZero .....	436
5.4.3.4.34. EcuM_DefaultRestartList .....	437
5.4.3.4.35. EcuM_DeterminePbConfiguration .....	437
5.4.3.4.36. EcuM_DisableWakeupSources .....	437
5.4.3.4.37. EcuM_EnableWakeupSources .....	438
5.4.3.4.38. EcuM_ErrorHook .....	438
5.4.3.4.39. EcuM_GenerateRamHash .....	438
5.4.3.4.40. EcuM_GetExpiredWakeupEvents .....	439
5.4.3.4.41. EcuM_GetPendingWakeupEvents .....	439
5.4.3.4.42. EcuM_GetStatusOfWakeupSource .....	439
5.4.3.4.43. EcuM_GetValidatedWakeupEvents .....	440
5.4.3.4.44. EcuM_GetVersionInfo .....	440
5.4.3.4.45. EcuM_GoDown .....	441
5.4.3.4.46. EcuM_GoHalt .....	441
5.4.3.4.47. EcuM_GoPoll .....	442
5.4.3.4.48. EcuM_GoRun .....	442
5.4.3.4.49. EcuM_Init .....	442
5.4.3.4.50. EcuM_IsValidConfig .....	443
5.4.3.4.51. EcuM_KillAllRUNRequests .....	443
5.4.3.4.52. EcuM_MainFunction .....	443
5.4.3.4.53. EcuM_OnCoreSync .....	444
5.4.3.4.54. EcuM_OnEnterRun .....	444
5.4.3.4.55. EcuM_OnExitPostRun .....	444
5.4.3.4.56. EcuM_OnExitRun .....	445



5.4.3.4.57. EcuM_OnGoOffOne .....	445
5.4.3.4.58. EcuM_OnGoOffTwo .....	445
5.4.3.4.59. EcuM_OnGoSleep .....	446
5.4.3.4.60. EcuM_OnPrepShutdown .....	446
5.4.3.4.61. EcuM_OnRTEStartup .....	446
5.4.3.4.62. EcuM_PostHalt .....	447
5.4.3.4.63. EcuM_PreHalt .....	447
5.4.3.4.64. EcuM_SetWakeupEvent .....	447
5.4.3.4.65. EcuM_Shutdown .....	448
5.4.3.4.66. EcuM_SleepActivity .....	448
5.4.3.4.67. EcuM_StartWakeupSources .....	449
5.4.3.4.68. EcuM_StartupTwo .....	449
5.4.3.4.69. EcuM_StopWakeupSources .....	449
5.4.3.4.70. EcuM_ValidateWakeupEvent .....	450
5.4.4. Integration notes .....	450
5.4.4.1. Exclusive areas .....	450
5.4.4.1.1. SCHM_ECUM_EXCLUSIVE_AREA_0 .....	450
5.4.4.1.2. SCHM_ECUM_EXCLUSIVE_AREA_{coreID} .....	451
5.4.4.2. Production errors .....	451
5.4.4.3. Memory mapping .....	451
5.4.4.4. Integration requirements .....	452
5.4.4.4.1. intgr.EcuM.EB_INTREQ_EcuM_0001 .....	452
5.5. Nm .....	452
5.5.1. Configuration parameters .....	452
5.5.1.1. CommonPublishedInformation .....	453
5.5.1.2. NmChannelConfig .....	456
5.5.1.3. NmBusType .....	464
5.5.1.4. NmGenericBusNmConfig .....	464
5.5.1.5. NmStandardBusNmConfig .....	465
5.5.1.6. NmGlobalConfig .....	466
5.5.1.7. NmGlobalConstants .....	466
5.5.1.8. NmGlobalFeatures .....	467
5.5.1.9. NmGlobalProperties .....	475
5.5.1.10. PublishedInformation .....	476
5.5.2. Application programming interface (API) .....	476
5.5.2.1. Type definitions .....	476
5.5.2.1.1. Nm_BusNmFpType .....	476
5.5.2.1.2. Nm_BusNmType .....	477
5.5.2.1.3. Nm_ModeType .....	478
5.5.2.1.4. Nm_StateType .....	478
5.5.2.1.5. Nm_TS_BusNm_ChkRemoteSleepIndFPtrType .....	478
5.5.2.1.6. Nm_TS_BusNm_GetInfoFPtrType .....	478



5.5.2.1.7. Nm_TS_BusNm_GetStateFPtrType .....	478
5.5.2.1.8. Nm_TS_BusNm_SetUserDataFPtrType .....	479
5.5.2.1.9. Nm_TS_BusNm_StdCbkFPtrType .....	479
5.5.2.1.10. Nm_TS_BusNm_StdFPtrType .....	479
5.5.2.2. Macro constants .....	479
5.5.2.2.1. NM_AR_RELEASE_MAJOR_VERSION .....	479
5.5.2.2.2. NM_AR_RELEASE_MINOR_VERSION .....	479
5.5.2.2.3. NM_AR_RELEASE_REVISION_VERSION .....	479
5.5.2.2.4. NM_BUSNM_CANNM .....	480
5.5.2.2.5. NM_BUSNM_FRNM .....	480
5.5.2.2.6. NM_BUSNM_GENERICNM .....	480
5.5.2.2.7. NM_BUSNM_LINNM .....	480
5.5.2.2.8. NM_BUSNM_UDPNM .....	480
5.5.2.2.9. NM_BUSNM_UNDEF .....	480
5.5.2.2.10. NM_CC_STATE_PREPARE_SHUTDOWN .....	481
5.5.2.2.11. NM_CC_STATE_SHUTDOWN_ABORTED .....	481
5.5.2.2.12. NM_CC_STATE_SHUTDOWN_INITIATED .....	481
5.5.2.2.13. NM_CC_STATE_WAIT_FOR_SHUTDOWN .....	481
5.5.2.2.14. NM_CC_STATE_WAIT_SYNCHRONIZATION .....	481
5.5.2.2.15. NM_E_HANDLE_UNDEF .....	481
5.5.2.2.16. NM_E_NETWORKGWERAREQUEST .....	482
5.5.2.2.17. NM_E_NETWORKRELEASE .....	482
5.5.2.2.18. NM_E_NETWORKREQUEST .....	482
5.5.2.2.19. NM_E_PARAM_POINTER .....	482
5.5.2.2.20. NM_E_PASSIVESTARTUP .....	482
5.5.2.2.21. NM_E_REQUESTBUSSYNCHRONIZATION .....	482
5.5.2.2.22. NM_E_UNINIT .....	483
5.5.2.2.23. NM_MODE_BUS_SLEEP .....	483
5.5.2.2.24. NM_MODE_NETWORK .....	483
5.5.2.2.25. NM_MODE_PREPARE_BUS_SLEEP .....	483
5.5.2.2.26. NM_MODE_SYNCHRONIZE .....	483
5.5.2.2.27. NM_MODULE_ID .....	483
5.5.2.2.28. NM_STATE_BUS_SLEEP .....	484
5.5.2.2.29. NM_STATE_NORMAL_OPERATION .....	484
5.5.2.2.30. NM_STATE_OFFLINE .....	484
5.5.2.2.31. NM_STATE_PREPARE_BUS_SLEEP .....	484
5.5.2.2.32. NM_STATE_READY_SLEEP .....	484
5.5.2.2.33. NM_STATE_REPEAT_MESSAGE .....	484
5.5.2.2.34. NM_STATE_SYNCHRONIZE .....	484
5.5.2.2.35. NM_STATE_UNINIT .....	485
5.5.2.2.36. NM_SW_MAJOR_VERSION .....	485
5.5.2.2.37. NM_SW_MINOR_VERSION .....	485



5.5.2.2.38. NM_SW_PATCH_VERSION .....	485
5.5.2.2.39. NM_VENDOR_ID .....	485
5.5.2.2.40. Nm_RemoteSleepCancelation .....	485
5.5.2.3. Objects .....	486
5.5.2.3.1. Nm_BusNmFp .....	486
5.5.2.4. Functions .....	486
5.5.2.4.1. Nm_BusSleepMode .....	486
5.5.2.4.2. Nm_CheckRemoteSleepIndication .....	486
5.5.2.4.3. Nm_CoordReadyToSleepIndication .....	487
5.5.2.4.4. Nm_DisableCommunication .....	487
5.5.2.4.5. Nm_EnableCommunication .....	488
5.5.2.4.6. Nm_GetCoordinatorState .....	489
5.5.2.4.7. Nm_GetLocalNodIdentifier .....	489
5.5.2.4.8. Nm_GetNodIdentifier .....	490
5.5.2.4.9. Nm_GetPduData .....	491
5.5.2.4.10. Nm_GetState .....	491
5.5.2.4.11. Nm_GetUserData .....	492
5.5.2.4.12. Nm_Init .....	493
5.5.2.4.13. Nm_MainFunction .....	493
5.5.2.4.14. Nm_NetworkMode .....	494
5.5.2.4.15. Nm_NetworkRelease .....	494
5.5.2.4.16. Nm_NetworkRequest .....	495
5.5.2.4.17. Nm_NetworkStartIndication .....	495
5.5.2.4.18. Nm_PassiveStartUp .....	496
5.5.2.4.19. Nm_PrepareBusSleepMode .....	496
5.5.2.4.20. Nm_RemoteSleepCancellation .....	497
5.5.2.4.21. Nm_RemoteSleepIndication .....	497
5.5.2.4.22. Nm_RepeatMessageRequest .....	498
5.5.2.4.23. Nm_SetUserData .....	498
5.5.2.4.24. Nm_SynchronizationPoint .....	499
5.5.2.4.25. Nm_TxTimeoutException .....	499
5.5.3. Integration notes .....	500
5.5.3.1. Exclusive areas .....	500
5.5.3.1.1. SCHM_NM_EXCLUSIVE_AREA_0 .....	500
5.5.3.2. Production errors .....	500
5.5.3.3. Memory mapping .....	500
5.5.3.4. Integration requirements .....	501



# 1. Overview of EB tresos AutoCore Generic 8 Mode Management documentation

Welcome to the EB tresos AutoCore Generic 8 Mode Management (ACG8 Mode Management) product documentation.

This document provides:

- ▶ [Chapter 2, “Supported features”](#): list of features supported by ACG8 Mode Management modules
- ▶ [Chapter 3, “ACG8 Mode Management release notes”](#): release notes for the ACG8 Mode Management modules
- ▶ [Chapter 4, “ACG8 Mode Management user guide”](#): background information and instructions
- ▶ [Chapter 5, “ACG8 Mode Management module references”](#): information about configuration parameters and the application programming interface



## 2. Supported features

### 2.1. Overview

This chapter provides an overview of the ACG8 Mode Management and the features that are currently supported.

[Section 2.2, “Supported BswM features”](#) contains an overview of BswM features.

[Section 2.3, “Supported ComM features”](#) contains an overview of ComM features.

[Section 2.4, “Supported EcuM features”](#) contains an overview of EcuM features.

[Section 2.5, “Supported Nm features”](#) contains an overview of Nm features.

### 2.2. Supported BswM features

- ▶ **Support for BSW distribution:** Support for distribution of multiple BswM instances over multiple partitions.
- ▶ **Support for mode arbitration:** Support for mode arbitration based on configurable rules.
- ▶ **Support for immediate and deferred operation:** Support for processing operations immediately, once they are requested or in the main function cycle.
- ▶ **Support for mode control:** Support for processing configured action lists based on the mode arbitration results.
- ▶ **Support for triggered and conditional action lists:** Support for executing actions every time a rule is evaluated or only when the evaluation result changes.
- ▶ **Support for mode request ports:** Support for ports to be used by SWCs in order to request modes.
- ▶ **Support for mode switch ports:** Support for ports to be used by SWCs in order to be notified about mode switches.
- ▶ **Support for notifications of mode switches:** Support for ports to be used by BswM in order to be notified about mode switches (BSW or SWC).
- ▶ **Support for post-build:** Support for handling post-build loadable and selectable configuration.
- ▶ **Support for link time:** Support for handling configuration at link time.
- ▶ **Support for BswM Timer:** Support for timer handling to delay actions or wait for further mode controls according to AUTOSAR 4.3.0.



## 2.3. Supported ComM features

- ▶ **Support for three communication modes:** Support for three communication modes according to AUTOSAR specifications (COMM\_FULL\_COMMUNICATION, COMM\_SILENT\_COMMUNICATION, COMM\_NO\_COMMUNICATION).
- ▶ **Support for three communication states:** Support for three main communication states with communication capabilities according to AUTOSAR specifications (COMM\_FULL\_COMMUNICATION, COMM\_SILENT\_COMMUNICATION, COMM\_NO\_COMMUNICATION).
- ▶ **Support for communication sub-states:** Support for communication sub-states COMM\_FULL\_COM\_NETWORK\_REQUESTED, COMM\_FULL\_COM\_READY\_SLEEP and COMM\_NO\_COM\_NO\_PENDING\_REQUEST, COMM\_NO\_COM\_REQUEST\_PENDING with communication capabilities according to AUTOSAR specifications.
- ▶ **Support for communication start-up:** Support for starting the communication due to user request of COMM\_FULL\_COMMUNICATION mode.
- ▶ **Support for communication shutdown:** Support for stopping the communication due to user request of COMM\_NO\_COMMUNICATION mode.
- ▶ **Support for communication passive wakeup:** Support for starting the communication due to wakeup event, network start or network restart indication.
- ▶ **Support for shutdown synchronization variants:** Support for shutdown synchronization variants (NONE, LIGHT, PASSIVE, FULL) according to AUTOSAR specifications.
- ▶ **Support for bus wakeup inhibition:** Support for inhibition of communication channels based on user requests.
- ▶ **Support for reading inhibition status:** Support for an interface to read the inhibition status of a channel.
- ▶ **Support for inhibition counter:** Support for a counter to store the number of rejected COMM\_FULL\_COMMUNICATION user requests, including interfaces to read and reset the value of the counter.
- ▶ **Support for limitation to NoCommunication:** Support for forcing the communication mode to NoCommunication even if users still request FullCommunication.
- ▶ **Support for providing the communication mode:** Support for a user interface to read the requested communication mode and current communication mode for a user.
- ▶ **Support for diagnostic communication request:** Support for processing diagnostic communication request and switching communication mode based on the diagnostic request.
- ▶ **Support for state change notification:** Support for notifications to ComM users and mode manager on mode changes.
- ▶ **Support for partial network cluster (PNC):** Support for PNC functionality and state machine according to AUTOSAR specifications.
- ▶ **Support for the Ethernet switch:** Support communication control by an active or passive start-up (PNC wakeup) when Ethernet switch is used.



- ▶ **Support for post-build:** Support for handling post-build loadable and selectable configuration.
- ▶ **Support for BSW distribution:** Support for inter-core communication when different bus types are processed on different cores.

## 2.4. Supported EcuM features

- ▶ **Support for BSW distribution:** Support for distributing EcuM over multiple cores (master-satellite approach with inter-core communication).
- ▶ **Support configurable master core:** Support for configurable core designation of the master EcuM instance.
- ▶ **Support for startup phase:** Support for handling activities before EcuM\_Init, activities in StartPreOs and StartPostOs sequences and support for driver initialization.
- ▶ **Support for shutdown phase:** Support for handling activities in the OffPreOS and OffPostOS sequences.
- ▶ **Support for sleep phase:** Support for handling activities in GoSleep, Halt, Poll and WakeupRestart sequences.
- ▶ **Support for UP phase:** Support for handling and validating wakeup sources and reset reasons.
- ▶ **Support for shutdown targets:** Support for sleep and reset states as shutdown targets.
- ▶ **Support for service interface according to AUTOSAR 3.2 and AUTOSAR 4.0.3:** Support interfaces for requesting and releasing of ECU states, notifying current mode, selecting and accessing the shutdown target and selecting the boot target.
- ▶ **Support for run request protocol:** Support for coordinating if the ECU must be kept alive or is ready to shut down.
- ▶ **Support for post-build:** Support for handling post-build selectable configuration.

## 2.5. Supported Nm features

- ▶ **Support for abstraction from bus-specific NM(s):** Support for interfaces that do not require information about the communication buses used.
- ▶ **Support for NM coordinator:** Support for NM coordinator algorithm to keep coordinated buses alive and shut down coordinated buses.
- ▶ **Support for state change notification:** Support for updating a Com signal with information regarding changes of the states in bus-specific network management module.
- ▶ **Support for car wakeup:** Support for calling a user-defined callback function or giving an indication to BswM when a lower bus-specific NM module indicates the reception of a network management PDU with the car wakeup bit set.



- ▶ **Support for channel-based configurable RepeatMsgInd|NodeDetection|NodeIdEnabled:** Support for a per-channel configuration of parameters NmRepeatMsgIndEnabled, NmNodeDetectionEnabled, NmNodeIdEnabled.
- ▶ **Support for BSW distribution:** Support for inter-core communication when different bus types are processed on different cores.



## 3. ACG8 Mode Management release notes

### 3.1. Overview

This chapter provides the ACG8 Mode Management product specific release notes. General release notes that are applicable to all products are provided in the EB tresos AutoCore Generic documentation. Refer to the general release notes in addition to the product release notes documented here.

### 3.2. Scope of the release

#### 3.2.1. Configuration tool

Your release of EB tresos AutoCore is compatible with the release of the EB tresos Studio configuration tool:

- ▶ EB tresos Studio: 28.2.0 b211016-0103

#### 3.2.2. AUTOSAR modules

The following table lists the AUTOSAR modules that are part of this ACG8 Mode Management release.

Module name	AUTOSAR version and revision	SWS version and revision	Module version	Supplier
<a href="#">BswM</a>	4.0.3 []	1.2.0 [0000]	1.15.7	Elektrobit Automotive GmbH
<a href="#">ComM</a>	4.1.3 []	4.1.1 [0]	5.19.8	Elektrobit Automotive GmbH
<a href="#">EcuM</a>	4.0.3 []	3.0.0 [0000]	5.15.8	Elektrobit Automotive GmbH
<a href="#">Nm</a>	4.0.3 []	3.0.0 [0000]	5.12.9	Elektrobit Automotive GmbH

Table 3.1. Hardware-Independent Modules specified by the AUTOSAR standard



### 3.2.3. EB (Elektrobit) modules

The following table lists all modules which are part of this release but are not specified by the AUTOSAR standard. These modules include tooling developed by EB or they may hold files shared by all other modules.

Module name	Module version	Supplier
<a href="#">BswMAs</a>	1.25.1	Elektrobit Automotive GmbH

Table 3.2. Modules not specified by the AUTOSAR standard

### 3.2.4. MCAL modules and EB tresos AutoCore OS

For information about MCAL modules and OS, refer to the respective documentation, which is available as PDF at `$TRESOS_BASE/doc/3.0_EB_tresos_AutoCore_OS` and `$TRESOS_BASE/doc/5.0_MCAL_modules`<sup>1</sup>. It is also available in the online help in EB tresos Studio. Browse to the folders EB tresos AutoCore OS and MCAL modules.

## 3.3. Module release notes

### 3.3.1. BswM module release notes

- ▶ AUTOSAR R4.0 Rev 3
- ▶ AUTOSAR SWS document version: 1.2.0
- ▶ Module version: 1.15.7.B466224
- ▶ Supplier: Elektrobit Automotive GmbH

#### 3.3.1.1. Change log

This chapter lists the changes between different versions.

##### Module version 1.15.7

2021-10-27

<sup>1</sup>`$TRESOS_BASE` is the location at which you installed EB tresos Studio.



- ▶ Improve Interrupt Block Time by removing ExArea in BswM\_MainFunction when queue mode arbitration is used.

#### **Module version 1.15.6**

2021-06-25

- ▶ Internal module improvement. This module version update does not affect module functionality.

#### **Module version 1.15.5**

2021-03-05

- ▶ Internal module improvement. This module version update does not affect module functionality.

#### **Module version 1.15.4**

2020-10-23

- ▶ Added general support for event request ports and BswMComMInitiateReset ports
- ▶ ASCBSWM-1241 Fixed known issue: BswMSwcModeRequest and BswMSwcModeNotification ports are not considered when defining managed mode groups in the basic software module description
- ▶ ASCBSWM-1252 Fixed known issue: BswM\_CanSM\_CurrentState throws DET error if CANSM\_BSWM\_CHANGE\_BAUDRATE is reported via BswM\_CanSM\_CurrentState

#### **Module version 1.15.3**

2020-06-19

- ▶ ASCBSWM-1201 Fixed known issue: Generation fails if a BswMSwcModeRequest port references a VariableDataPrototype that uses an ApplicationPrimitiveDataType
- ▶ Added support for J1939 mode request ports and actions

#### **Module version 1.15.2**

2020-02-21

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 1.15.1**

2019-10-11



- ▶ ASCBSWM-1160 Fixed known issue: Unused BswMSchMSwitch actions cause an invalid configuration if `IMPLEMENTATION_CONFIG_VARIANT` is VariantPreCompile

#### **Module version 1.15.0**

2019-06-14

- ▶ ASCBSWM-1142 Fixed known issue: Deadline monitoring cannot be disabled when BswMDeadlineMonitoringControl actions are executed before BswMPduGroupSwitch actions
- ▶ Added postbuild selectable support
- ▶ ASCBSWM-1133 Fixed known issue: Changing rule expressions at post-build time might lead to unexpected behavior
- ▶ ASCBSWM-1134 Fixed known issue: Referencing previously unused actions at post-build time leads to unexpected behavior
- ▶ Added support for queued mode arbitration
- ▶ BswM\_Dcm\_RequestCommunicationMode now uses service identifier 0xFE instead of 0x06

#### **Module version 1.14.5**

2019-02-15

- ▶ ASCBSWM-1123 Fixed known issue: Initialization of multiple cores can lead to data corruption and unexpected behaviour

#### **Module version 1.14.4**

2019-01-31

- ▶ ASCBSWM-1113 Fixed known issue: Incorrect allocation of the internal IpduGroupVector leads to an out-of-bounds access in Com
- ▶ ASCBSWM-1110 Fixed known issue: Invalid basic software module description if an immediate BswM-Timer port triggers the execution of a SchMSwitch action

#### **Module version 1.14.3**

2018-10-26

- ▶ Internal module improvement. This module version update does not affect module functionality
- ▶ ASCBSWM-1107 Fixed known issue: Infinite loop if more than 2040 I-PDU groups are configured



### Module version 1.14.2

2018-06-22

- ▶ Added support for Rte Switch Acknowledgements
- ▶ Added memory sections for the BswM software component description
- ▶ Added constraint that BswMPduGroupSwitch and BswMDeadlineMonitoringControl actions can only be configured in one BswM instance
- ▶ Added support for BswM timers
- ▶ ASCBSWM-1074 Fixed known issue: BswM\_PostBuildConfig uses different memory sections for declaration and definition

### Module version 1.14.1

2018-02-16

- ▶ ASCBSWM-967 Fixed known issue: BswM generates invalid code if a mode condition referencing a BswM-NmlfCarWakeUpIndication port doesn't have 0 as a condition value
- ▶ ASCBSWM-981 Fixed known issue: Wrong memory section used for BswM\_PtrModeRequestPortsTable
- ▶ ASCBSWM-995 Fixed known issue: BswM triggers the wrong PDUs if multiple BswMTriggerIPduSend actions exist
- ▶ ASCBSWM-1004 Fixed known issue: Generation fails if BswMGenericRequest ports are used only in some BswM configurations
- ▶ ASCBSWM-997 Fixed known issue: BswM triggers the wrong PDUs if ComIPduHandleId and PduRSourcePduHandleId have different values
- ▶ Enhanced the BSWM\_E\_WRONG\_CONTEXT Det error when BswM API's are called from an invalid partition

### Module version 1.14.0

2017-09-21

- ▶ Major code refactoring for improving RAM and ROM usage
- ▶ Added support for Ethernet Switch Port Groups
- ▶ ASCBSWM-926 Fixed known issue: BswM generator produces invalid code or crashes if cross-partition references are used
- ▶ ASCBSWM-931 Fixed known issue: Inconsistency between generated code and BSWMD when unused BswMSchMSwitch actions exist
- ▶ Added support for Car Wakeup

**Module version 1.13.0**

2017-03-31

- ▶ ASCBSWM-880 Fixed known issue: Single instance BswM fails to compile in a multicore ECU

**Module version 1.12.2**

2017-03-13

- ▶ Symbolic values from user defined header files can be used as initial values for mode request ports or as condition values for mode conditions
- ▶ Mode request port tables generation order is now preserved
- ▶ ASCBSWM-853 Fixed known issue: Possible data corruption if processing nested rules with triggered action list execution is interrupted
- ▶ ASCBSWM-838 Fixed known issue: Premature handling of I-PDU groups when mode control is interrupted
- ▶ ASCBSWM-859 Fixed known issue: Uncompilable code when a mode request port is initialized with a symbolic name value
- ▶ ASCBSWM-870 Fixed known issue: Disabling deadline monitoring control might be ineffective
- ▶ ASCBSWM-346 Fixed known issue: The BswM may reference a data type that is not specified in ASR-4.-0.3 Dcm SWS

**Module version 1.12.1**

2016-12-15

- ▶ ASCBSWM-813 Fixed known issue: Uncompilable code when variant PostBuild and multiple instances are configured
- ▶ Headers for other BSW modules are generated only when the corresponding parameter is enabled
- ▶ Added support for DEM error reporting
- ▶ Simplified the configuration of generic request ports

**Module version 1.12.0**

2016-11-04

- ▶ Added a warning message in case a BswMSwitchPort is not used by any SchMSwitch or RteSwitch action
- ▶ Added validation for the TARGET parameter of a BswMRteModeRequestPortInterfaceRef
- ▶ Added a warning message if a BswMNestedExecutionOnly rule is not referenced by any action list.
- ▶ Improve validation of the BswMRteUsage parameter



- ▶ Enhanced the basic software module description
- ▶ ASCBSWM-795 Fixed known issue: Incorrect behaviour when multiple BswM partitions are on the same core
- ▶ ASCBSWM-800 Fixed known issue: Incorrect mode arbitration if more than 2 BswM configurations exist
- ▶ Fixed issue with exporting unused configuration parameters

#### **Module version 1.11.0**

2016-05-25

- ▶ ASCBSWM-705 Fixed known issue: DET errors are always reported from the first partition
- ▶ ASCBSWM-709 Fixed known issue: Missing semicolon in generated SchMSwitch action functions
- ▶ R-PORT Prototype definition corresponding to a SwcModeRequest port uses its configured initial value
- ▶ Improved configuration validations for logical expressions and rules
- ▶ ASCBSWM-715 Fixed known issue: Uncompilable code is produced when BswMBswRequestedMode is empty
- ▶ ASCBSWM-716 Fixed known issue: Invalid memory access when accessing initial values
- ▶ Optimization when no BswMPduGroupSwitch actions with BswMPduGroupSwitchReinit - TRUE exist
- ▶ ASCBSWM-736 Fixed known issue: Memory access violation if multiple BswM instances are used
- ▶ ASCBSWM-775 Fixed known issue: Invalid BSWMD and SWCD if multiple BswM configurations exist
- ▶ Optimizations for the BswM\_MainFunction execution time
- ▶ DET error logged when BswM API functions are called from an invalid context
- ▶ Improved validation of BswMUserCalloutFunction

#### **Module version 1.10.0**

2016-02-10

- ▶ ASCBSWM-698 Fixed known issue: Invalid handling of duplicate I-PDU groups in case of PduGroupSwitch or DeadlineMonitoringControl actions

#### **Module version 1.9.1**

2016-01-15

- ▶ ASCBSWM-654 Fixed known issue: Wrong definition of BswMPduGroupSwitchActionType and BswMDeadLineMonitoringControlActionType produces invalid code
- ▶ Renamed short names of SERVICE-SW-COMPONENT-TYPE instances



- ▶ ASCBSWM-656 Fixed known issue: BswM generates uncompilable code when using Rte mode request ports
- ▶ ASCBSWM-659 Fixed known issue: Incorrect SchM and BswM headers included in a multi-core setup
- ▶ The symbols of runnable entities on different partitions need to be unique
- ▶ Configuration errors added in case actions using disabled external interfaces are configured
- ▶ Performance optimization when performing mode control
- ▶ Added support for adding a new empty BswM configuration container
- ▶ Removed unused header files BswM\_RTE.h, BswM\_COM.h and User\_Callout.h
- ▶ Added configuration check that all referenced mode declaration groups have corresponding data type mapping sets
- ▶ ASCBSWM-667 Fixed known issue: BswM disables deadline monitoring for I-PDU groups
- ▶ Fixed compiler abstraction mismatch between Rte declarations and BswM definitions
- ▶ ASCBSWM-669 Fixed known issue: Generation error when using Com, PduRControl or LinScheduleSwitch actions in a multicore configuration
- ▶ Added better configuration checks for the parameters inside the EcuMSelectShutdownTarget action
- ▶ ASCBSWM-682 Fixed known issue: No DET error check for request id in case 255 GenericRequest ports are configured
- ▶ Removed restriction that the BswMModeRequesterIds need to be 0 based consecutive

#### Module version 1.9.0

2015-11-10

- ▶ Improved error handling for mode request ports during generation
- ▶ Added support for multiple instantiation
- ▶ Renamed BswM\_LinIf.h to BswM\_LinTp.h

#### Module version 1.8.0

2015-06-24

- ▶ Limited the number of artifacts displayed in the Rule-Port Dependencies window
- ▶ ASCBSWM-581 Fixed known issue: Ports of type BswMDcmApplicationUpdated are reported to be unused
- ▶ Limited the options for configuring the BswM\_Dcm\_ApplicationUpdated()
- ▶ Improved generated code of logical expression states



- ▶ Implemented RfC #54616: BswM\_WdgM\_RequestPartitionReset cannot be treated like the rest of indication services
- ▶ Improved error messages related to mode request ports in the BswM Editor
- ▶ Improved error handling inside the BswM Editor
- ▶ Removed BswM.h inclusion from BswM\_ComM.h
- ▶ Moved the BswM Editor plugin to a separate repository
- ▶ ASCBSWM-617 Fixed known issue: Configuring the same BswMBswModeInitValueMode value in different ports lead to errors during generation
- ▶ Added support for using a variable data prototype reference in a BswMSwcModeRequest port

#### **Module version 1.7.0**

2015-02-20

- ▶ Removed SwcBswMapping constraint related to the existence of a SwitchPort
- ▶ Removed generated mode conditions macros
- ▶ Improved the names of the containers generated by the BswM Editor

#### **Module version 1.6.0**

2015-01-07

- ▶ Added support for CanSM enhanced bus off reporting
- ▶ Changed the default value of BswMRelocatablePbcfgEnable to true
- ▶ Moved generated header type definitions into BswM\_Generated\_Types.h
- ▶ Removed redundant code related to Pb relocatable in BswM.h
- ▶ ASCBSWM-513 Fixed known issue: The rule parameter BswMNestedExecutionOnly is ignored for mode request ports with immediate evaluation
- ▶ Introduced the BswM Editor
- ▶ Modified parameter of BswM\_ExecuteModeControl function to const
- ▶ Added BswM Editor user's guide
- ▶ Added support for BswM\_Dcm\_ApplicationUpdated() API
- ▶ Updated the module version strings within the BswM Editor source code

#### **Module version 1.5.0**

2014-10-02



- ▶ ASCBSWM-453 Fixed known issue: A mode declaration cannot be used to define the initial value of a generic mode request port
- ▶ The EcuM can now initialize the BswM with a NULL\_PTR
- ▶ ASCBSWM-449 Fixed known issue: The inclusion of the BswM post-build header file may lead to a compiler error
- ▶ ASCBSWM-478 Fixed known issue: The configuration parameter BswMSchMEnabled contains invalid XPath expressions
- ▶ ASCBSWM-481 Fixed known issue: Invalid warning caused by the BswM configuration parameter BswME-cuMShutdownTargetRef
- ▶ Reduced the number of statements within the function BswM\_LT\_Init
- ▶ Added INVALID checks for all REF parameters
- ▶ Adapted the use of EcuM wakeup source symbolic name constants
- ▶ Added support for the Ecu configuration container BswMCompuScaleModeValue
- ▶ The BswM action BswMRteModeRequest is now supported
- ▶ Added support for symbolic constants

#### Module version 1.4.0

2014-06-27

- ▶ ASCBSWM-458 Fixed known issue: The BswM module is uninitialized in case of a software reset

#### Module version 1.3.15

2014-06-13

- ▶ ASCBSWM-426 Fixed known issue: Compiler error due to the multiple inclusion of application header files
- ▶ The macro BSWM\_PB\_CONFIG\_NAME is now mapped to the macro BSWM\_CONFIG\_NAME
- ▶ ASCBSWM-444 Fixed known issue: The Det check for requesting\_user in BswM\_RequestMode() is wrong
- ▶ Use of the macro BSWM\_COM\_ACTION\_USED has been replaced where the macros BSWM\_DEADLINE\_MONITORING\_CONTROL\_ACTION\_USED, BSWM\_PDU\_GROUP\_SWITCH\_ACTION\_USED, BSWM\_SWITCH\_IPDU\_MODE\_ACTION\_USED or BSWM\_TRIGGER\_IPDU\_SEND\_ACTION\_USED can be used
- ▶ The macro BSWM\_PDUR\_ACTION\_USED has been replaced with the macro BSWM\_PDU\_ROUTER-CONTROL\_ACTION\_USED
- ▶ The function BswM\_GetVersionInfo now ignores a request when a NULL\_PTR is passed to the function
- ▶ The function BswM\_Dcm\_CommunicationMode\_CurrentState now uses the new data type BswMDcm-ComModeCurrentStatePortType

**Module version 1.3.14**

2014-03-21

- ▶ Changed passing of corresponding numerical values to enumeration constants from the BswM to the Service Discovery module
- ▶ ASCBSWM-397 Fixed known issue: The generation of calls to Rte switch functions is not conform to AUTOSAR
- ▶ Implemented post-build and link-time support
- ▶ Delayed execution of mode arbitration requests is now supported
- ▶ Delayed execution of the Com\_IpduGroupControl command is now supported
- ▶ ASCBSWM-421 Fixed known issue: The BswM generator may produce an invalid warning regarding unused action lists
- ▶ ASCBSWM-422 Fixed known issue: A triggered action list may be executed when the rule's result has not changed
- ▶ ASCBSWM-423 Fixed known issue: BswMLinTpModeRequest mode request ports may be ignored or incorrectly updated

**Module version 1.3.13**

2013-09-17

- ▶ Removed Det error checks for BSWM\_E\_REQ\_USER\_OUT\_OF\_RANGE for all BswM mode request sources except BswM\_RequestMode

**Module version 1.3.12**

2013-08-28

- ▶ ASCBSWM-373 Fixed known issue: A compiler error occurs because of missing MemMap macro

**Module version 1.3.11**

2013-08-09

- ▶ Added a unique short name for required mode declaration group prototypes

**Module version 1.3.10**

2013-07-30



- ▶ Changed to correct the implementation configuration class of several BswM Ecu configuration parameters

### **Module version 1.3.9**

2013-07-23

- ▶ Added an initial value definition of SwcModeRequest ports for the BswM service component description
- ▶ ASCBSWM-341 Fixed known issue: The basic software module description contains an invalid reference to its SwcBswMapping
- ▶ Added Intra Module Checks to avoid integration of incompatible files
- ▶ ASCBSWM-345 Fixed known issue: If only one mode request port is defined for a mode request source that accepts multiple channels, the BswM may assign the status of different channels to the mode request port
- ▶ ASCBSWM-348 Fixed known issue: A Det parameter check within the function BswM\_EcuM\_CurrentWakeup does not recognize an out of range parameter
- ▶ Added configuration parameter for queue length of mode switch ports

### **Module version 1.3.8**

2013-06-18

- ▶ ASCBSWM-283 Fixed known issue: Only the first IPdu will be sent for a BswMTriggerIPduSend action
- ▶ Added Debug and Trace support for the BswM

### **Module version 1.3.7**

2013-05-14

- ▶ ASCBSWM-224 Fixed known issue: The file BswM\_Cfg.h may not conform to the ISO C90 standard
- ▶ Removed the validation of constants passed to the BswM in separate functions
- ▶ Added support for the AUTOSAR Service Discovery module

### **Module version 1.3.6**

2013-04-15

- ▶ ASCBSWM-209 Fixed known issue: The BswM generator may produce duplicate mode switch point declarations
- ▶ ASCBSWM-212 Fixed known issue: Multiple wakeup sources passed to the BswM API function BswM\_EcuM\_CurrentWakeup will not be evaluated



- ▶ ASCBSWM-214 Fixed known issue: The BswM does not generate mode declaration initial values correctly
- ▶ ASCBSWM-198 Fixed known issue: The BswM may reference undefined mode declaration identifiers

#### Module version 1.3.5

2013-03-15

- ▶ ASCBSWM-96 Fixed known issue: The parameter CurrentState of the function BswM\_FrSM\_CurrentState may be validated incorrectly when compiler options are enabled
- ▶ ASCBSWM-202 Fixed known issue: The generation order of certain BswM code fragments is not constant
- ▶ Implemented BswMActions as functions
- ▶ Removed the definition of a default value if no initial value has been defined for a mode request port
- ▶ Changed the validation of constants passed to the BswM into separate functions
- ▶ Changed the data types of BswM variables which store boolean values to be explicitly defined as the standard type "boolean"
- ▶ Removed unnecessary calls to the macro TS\_PARAM\_UNUSED
- ▶ Changed the BswM service component description to define a queue length for each BswMSwitchPort

#### Module version 1.3.4

2013-02-11

- ▶ The BswM actions BswMEcuMSelectShutdownTarget, BswMEcuMGoDown and BswMPduRouterControl are now supported
- ▶ The BswM now supports BswMSchMSwitch actions
- ▶ The BswM now supports BswMBswModeNotification mode request ports
- ▶ Added XPath validation checks for the Ecu parameters BswMArgumentRef and BswMRuleExpressionRef
- ▶ ASCBSWM-171 Fixed known issue: The generated BswM will not compile if a triggered action list is executed from a conditional action list
- ▶ Updated config structure (reference paths) of BswM-ComM channel reference for the introduction of ComMConfigSet container in ComM
- ▶ Removed SvcAs request for SchM exclusive areas since they are no longer used

#### Module version 1.3.3

2013-01-11



- ▶ Mode arbitration is now disabled after BswM\_Deinit has been executed

### Module version 1.3.2

2012-12-14

- ▶ ASCBSWM-86 Fixed known issue: If one or more unused RteSwitchActions exist a null pointer exception may be thrown during the generation of the BswM

### Module version 1.3.1

2012-11-16

- ▶ ASCBSWM-79 Fixed known issue: The function BswM\_WdgM\_RequestPartitionReset incorrectly validates the range of the parameter "Application"
- ▶ ASCBSWM-107 Fixed known issue: If multiple BswMComMIndication mode request ports reference the same CommChannel only one mode request port will be evaluated
- ▶ ASCBSWM-88 Fixed known issue: The function BswM\_LinSM\_CurrentSchedule references the undefined macro BSWM\_LINIF\_SCHEDULE\_TABLE\_MAX
- ▶ ASCBSWM-83 Fixed known issue: If a BswMSwcModeNotification is incorrectly configured the BswM generation will terminate with a null pointer exception
- ▶ ASCBSWM-80 Fixed known issue: The BswM basic software module description contains an invalid reference to the to BswM EcuC module definition
- ▶ ASCBSWM-145 Fixed known issue: The function BswM\_LinSM\_CurrentSchedule incorrectly evaluates Lin network schedule indication requests
- ▶ ASCBSWM-148 Fixed known issue: BswMFrSMIndication mode request sources are not recognized by the BswM generator

### Module version 1.3.0

2012-10-24

- ▶ The top-level structure of the software-component description in the ARXML files changed from /AU-TOSAR/BswM to /AUTOSAR\_BswM
- ▶ ASCBSWM-109 Fixed known issue: BswM generation fails with a Java null pointer exception when a Mode-Type-Request-Map is not complete
- ▶ ASCBSWM-84 Fixed known issue: The BswM does not support a EthSM Channel selection for mode switch ports



- ▶ ASCBSWM-76 Fixed known issue: The verification of EcuM wakeup sources is carried out with incorrect identifiers
- ▶ ASCBSWM-112 Fixed known issue: The generated code for BswMCanSMIndication request port is wrong if more than one CanSMIndication is used

#### **Module version 1.2.1**

2012-06-27

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 1.2.0**

2012-03-23

- ▶ Updating configuration and functionality to 4.0.3, preserving AR 4.0.2 paths to other module configurations

#### **Module version 1.1.1**

2011-10-14

- ▶ Integrated SWS enhancements and clarifications; other minor enhancements

#### **Module version 1.1.0**

2011-09-09

- ▶ AUTOSAR 4.0 version (full)

### **3.3.1.2. New features**

- ▶ No new features have been added since the last release.

### **3.3.1.3. EB-specific enhancements**

This chapter lists the enhancements provided by the module.

- ▶ This module provides no EB-specific enhancements.



### 3.3.1.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

- ▶ BswMMainFunctionPeriod can be configured up to 255 seconds

In order to optimize the BswMTimer implementation, the BswMMainFunctionPeriod parameter has been restricted to a maximum value of 255 seconds.

This should not affect any project as this value is sufficiently high.

- ▶ Unsupported actions

Description:

The following actions are not supported.

- ▶ BswMTriggerSlaveRTEStop
- ▶ BswMTriggerStartUpPhase2

Requirements:

BswM0919\_Conf, BswM0921\_Conf, BswM0920\_Conf,

- ▶ The short-names of the BswM service component description artifacts

Description:

The short-names of the following BswM service component description artifacts are not generated according to the requirement BswM0137: - RequirePort - ProvidePort - RunnableEntity - DataReceiveEvent

Requirements:

BswM0137

- ▶ BswMDcmComMChannelRef in place of BswMDcmComMNetwork

Description:

The configuration parameter BswMDcmComMNetwork has been replaced with a reference-parameter BswMDcmComMChannelRef which references a ComMChannel.

Rationale:

References to ComM channel is much less error-prone than allowing arbitrary strings to be entered as ComM-channel-identifiers. This is the topic of AUTOSAR bugzilla issue [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=53057](http://www.autosar.org/bugzilla/show_bug.cgi?id=53057)

Requirements:

BswM0876\_Conf



- ▶ The parameter `BswMEcuMSleepModeRef` in place of `BswMEcuMShutdownTargetRef`

Description:

The configuration parameter `BswMEcuMShutdownTargetRef` has been replaced with a reference-parameter `BswMEcuMSleepModeRef`.

Rationale:

The Ecu configuration parameters for the shutdown target selection were not complete. `BswMEcuMShutdownTargetRef` could only reference a sleep target and not a reset target. The shutdown target selection has been adapted here: [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=54134](http://www.autosar.org/bugzilla/show_bug.cgi?id=54134)

Requirements:

`BswM0962_Conf`

- ▶ Unsupported APIs `BswM_TriggerStartUpPhase2` and `BswM_TriggerSlaveRTEStop`

Description:

The APIs `BswM_TriggerStartUpPhase2()` and `BswM_TriggerSlaveRTEStop()` are related to Multicore feature whose functionality is to trigger (start or stop) the RTE/SchM on slave core is not supported.

Requirements:

`BswM0140, BswM0142, BswM0145, BswM0141, BswM0143, BswM0144`

- ▶ Removed Det checks for `BSWM_E_REQ_USER_OUT_OF_RANGE`

Description:

The Development Error Tracer checks associated to `BSWM_E_REQ_USER_OUT_OF_RANGE` have been removed since the BswM does not know the valid ranges of the user parameter.

Requirements:

`BswM0092, BswM0094, BswM0096, BswM0100, BswM0102, BswM0108, BswM0111, BswM0151, BswM0155, BswM0114, BswM0098`

- ▶ Optional element `BswMConditionValue`

Description:

The BswM requires that the ECU configuration parameter `BswMConditionValue` be always defined. It is not optional.

Requirements:

`BswM0816_Conf`



- ▶ Disabled parameter `BswMWdgMRequestPartitionResetRef`

Description:

The `BswMWdgMRequestPartitionResetRef` parameter cannot be used to indicate the partition for which the reset request is made for. This is due to the fact the Os module does not have the `OsAppEcucPartitionRef` parameter enabled. As a result, this functionality is broken. In order to correctly provide the partition for which the reset request is made for, the new EB specific parameter `BswMWdgMOsApplicationRef` must be used.

Requirements:

`BswM0917_Conf`

- ▶ `BswM` doesn't provide the declaration for `BswM_MainFunction`

Description:

The `BswM` will not provide the declaration of the `BswM_MainFunction` anymore. In case the `BswM` has only one instance the declaration of `BswM_MainFunction` will be provided by the Schm. In case of multiple instantiation, each partition will have its own `BswM_1_{Partition}_MainFunction`.

Requirements:

`BswM0053, BswM0075, BswM.Impl.Api.BswM_MainFunction_Partition`

- ▶ `BswMPartitionRef` references an `OsApplication` instead of a `EcucPartition`

Description:

The `BswMPartitionRef` parameter doesn't reference an `EcucPartition` but an `OsApplication` due to Os restrictions.

Requirements:

`ECUC_BswM_00984`

- ▶ `BswM` always requires the SchM interface

Description:

The `BswMSchMEnabled` parameter will be disabled and it's default value shall be 'true'. The rationale is that the SchM interface is always needed by the `BswM`.

Requirements:

`BswM00947_Conf`

- ▶ `BswM` provides the header files for other modules only if `BswM<mod>` is set to TRUE

Description:

Page 47 of 501



The newer version of this requirement (from Autosar 4.2) is preferred because it reduces the number of header files which are provided by the BswM.

Requirements:

BswM0026

- ▶ BswM\_UserType and BswM\_ModeType types are not configuration dependent.

Description:

According to BswM0041, the range of the BswM\_UserType and BswM\_ModeType types should be configuration dependent. Instead, the current implementation defines these types as uint16 to simplify the implementation. BswM0041 also mentions BswM\_ConfigType but this is already covered by BswM0042.

Requirements:

BswM0041

- ▶ BswMRequestedModeMax is not used by the BswM.

Description:

As the BswM\_ModeType is now fixed at uint16, it doesn't make any sense for the user to configure the BswMRequestedModeMax anymore.

Requirements:

BswM0875\_Conf

### 3.3.1.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- ▶ For this module no limitations are known.

### 3.3.1.6. Open-source software

BswM does not use open-source software.

## 3.3.2. BswMAs module release notes

- ▶ Module version: 1.25.1.B466224



- ▶ Supplier: Elektrobit Automotive GmbH

### 3.3.2.1. Change log

This chapter lists the changes between different versions.

#### Module version 1.25.1

2021-06-25

- ▶ Add Open source statement in the release notes.

#### Module version 1.24.1

2021-03-05

- ▶ Added support for element names conflicted with grammar keywords.

#### Module version 1.23.1

2020-12-18

- ▶ BswM Editor shows errors after displaying paths that ends with a slash sign

#### Module version 1.22.1

2020-10-22

- ▶ Added support for BswMJ1939DcmBroadcastStatus, BswMJ1939NmlIndication, BswMJ1939DcmStateSwitch and BswMJ1939RmStateSwitch.
- ▶ Added support for BswMEventRequestPorts.
- ▶ Added support for BswMComMInitiateReset.
- ▶ Added support for event conditions.
- ▶ Added support for BswMClearEventRequest.
- ▶ Fixed cross references (BswMSwitchPortRef, BswMRteSwitchPortRef, BswMTimerRef etc.) serialization bug.
- ▶ Investigate static analysis issue

#### Module version 1.21.1

2020-06-19



- ▶ Added optional inline name to Action element in the grammar.
- ▶ Added names conflicts resolving dialog.
- ▶ Added optional inline name to Expression and ModeCondition element in the grammar.
- ▶ Added optional inline name to ActionList and ActionListItem element in the grammar.

### **Module version 1.20.1**

2020-02-21

- ▶ Fixed loading of clean configuration containing one freshly created port.

### **Module version 1.20.0**

2019-10-11

- ▶ Added support for ModeCondition with invalid path reference.
- ▶ Added support for BswMNVJobModelIndication ports with invalid NvMServices value.
- ▶ Added support for ActionLists with invalid ActionListItems
- ▶ Added support for invalid logical operators in complex expressions.
- ▶ Added support for invalid references in logical expressions.
- ▶ Added support for invalid timer references in timer controls.
- ▶ Only nested rule is suggested for the NestedRule element.
- ▶ When non nested rule is added as NestedRule element a "Rule must be nested" validation error is shown.
- ▶ Fixed issue with ports not being proposed by content assist when they were defined under the rule.

### **Module version 1.19.0**

2019-06-14

- ▶ Added suggestions for timer values in logical expressions.
- ▶ Updated grammar for rule's initState value
- ▶ For GenericPort parameter BswMRequestModeMax is not mandatory any more.
- ▶ Added suggestion for logical expression rules. Added suggestion for Timer Value in Timer Control action. Added dependency view to rule which runs TimerControl. Remove suggestion of initialValue for BswM-Timer.
- ▶ Updated grammar for RteSwitchPortRef field of BswMRteSwitchAckNotification
- ▶ Added invalid DataMapping handling
- ▶ Added possibility to save invalid reference paths



- ▶ Updated grammar for DataMapping field of BswMBswModeNotification, BswMSwcModeNotification and BswMSwcModeRequest
- ▶ Added support for invalid values of queue length inside SwitchPort
- ▶ Added support for invalid values of BswMModeRequesterId and BswMRequestedModeMax in GenericRequestPort
- ▶ Added support for invalid values of SwitchIPduModeValue in SwitchIPduModeAction
- ▶ Added possibility to save switch ports with empty reference path.
- ▶ Added support for rules with invalid expression path.
- ▶ Added support for rules with invalid path to TrueActionList or FalseActionList.
- ▶ Added support for invalid BswMTimerControl actions.
- ▶ Added support for empty path references.
- ▶ Added support for invalid enum values.
- ▶ Added support for invalid SendPort and SwitchPort references.
- ▶ ASCBSWMAS-117 Upon serialization (i.e. saving the BswM configuration back to the Ecu data model when the BswM Editor is closed and saved), if the user had deleted any artifacts from the BswM configuration, then only identifiable elements (ports and rules) will be deleted from the BswM configuration model. For elements which are not identifiable (actions, action lists, conditions, expressions), user will be asked for confirmation whether those object also should be removed.

#### **Module version 1.18.0**

2019-02-15

- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 1.17.0**

2018-10-26

- ▶ Added support for Timer according to ASR4.3.
- ▶ Added support for the BswMLegacyDcmEnabled parameter.

#### **Module version 1.16.0**

2018-06-22

- ▶ Added support for RTE mode switch acknowledgments: added a new type of mode request port (BswMRteSwitchAckNotification).
- ▶ Added QF-Tests for UI.



### Module version 1.15.0

2017-12-15

- ▶ Added design documentation.
- ▶ Added buttons in the BswM Editor toolbar for adding and removing editor tabs/BswM configurations.
- ▶ Added requirements for the BswM Assistant.
- ▶ Added support for ethernet switch port groups: added a new type of mode request port (BswMEthIfPort-GroupLinkStateChg) and a new type of action (BswMEthIfSwitchPortGroupRequestMethod).
- ▶ ASCBSWMAS-103 Fixed known issue: Switch Ports which are referenced elsewhere in the configuration, in actions of type BswMRteSwitch or BswMSchMSwitch, will end up duplicated in the Ecu data model upon closing the BswM Editor (there will be 2 containers with the same name, defining the same switch port).
- ▶ Added Unit Tests for supported BswM artifacts, loading and saving BswM configurations.
- ▶ Added support for ReportFailedToDemRef.
- ▶ ASCBSWMAS-135 Fixed known issue: If there are 2 rules defined in a BswM configuration, where rule #1 is defined before rule #2 and rule #1 contains a reference to rule #2 in one of its action lists (rule #2 is referenced as a nested rule in one of the action lists of rule #1), then the BswM configuration will fail to load in the BswM Editor (the corresponding BswM Editor tab is empty after opening the BswM Editor). Also fixed: upon saving a BswM configuration which fails to load, the identifiable artifacts from that configuration will be removed.
- ▶ Added support for Car Wakeup: added a new type of mode request port (BswMNmlfCarWakeUpIndication) and a new boolean parameter under BswMGeneral (NM Enabled).
- ▶ Added remaining Unit Tests that were still missing.

### Module version 1.14.0

2017-05-05

### Module version 1.13.0

2017-03-31

- ▶ For configuration artifacts from the original configuration which have not been changed in the BswM editor, their shortnames and other attributes are preserved when the BswM Editor is closed and saved.
- ▶ When there are configuration artifacts which are inconsistent and cannot be loaded into the BswM Editor, then if the BswM Editor is saved those artifacts are no longer removed from the BswM configuration.

### Module version 1.12.0

2016-11-04



- ▶ Improved validation in case of PduGroupSwitch and DeadlineMonitoringControl actions
- ▶ Improved validation in case of rules without any true or false action lists set

#### **Module version 1.11.0**

2016-05-25

- ▶ ASCBSWMAS-61 Improved validations for data type mapping sets
- ▶ ASCBSWMAS-60 Fixed known issue: Data type mapping sets are not correctly serialized if BswM-DataTypeMappingSetRef is not already enabled

#### **Module version 1.10.0**

2016-02-08

#### **Module version 1.9.1**

2016-01-15

- ▶ Removed the QuickOutline warning showing up in Tresos at plugin load
- ▶ ASCBSWMAS-38 Fixed known issue: Invalid configuration in case Rte actions are used in configuration which isn't named 'BswMConfig'
- ▶ ASCBSWMAS-23 Fixed known issue: Editor crashes randomly when using content assist
- ▶ ASCBSWMAS-41 Fixed known issue: the BswM Editor now enables the BswMUserIncludeFiles container if user includes are configured and automatically removes duplicate entries
- ▶ ASCBSWMAS-44 Fixed known issue: Editor doesn't display an error message when a referenced mode request port is removed
- ▶ ASCBSWMAS-45 Improved validation in case of EcuMSelectShutdownTarget action
- ▶ ASCBSWMAS-47 Improved validation in case of LinScheduleSwitch action

#### **Module version 1.9.0**

2015-11-10

- ▶ Added support for multiple instantiation

#### **Module version 1.8.0**

2015-06-24



- ▶ Added support for the BswMSwcModeRequestVariableDataPrototypeRef

#### **Module version 1.7.0**

2015-05-28

- ▶ Moved the BswM Editor to the new BswMAs repository

#### **3.3.2.2. New features**

- ▶ No new features have been added since the last release.

#### **3.3.2.3. EB-specific enhancements**

This module is not part of the AUTOSAR specification.

#### **3.3.2.4. Deviations**

This module is not part of the AUTOSAR specification.

#### **3.3.2.5. Limitations**

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- ▶ For this module no limitations are known.

#### **3.3.2.6. Open-source software**

The BswmAs module does not use open-source software.

### **3.3.3. ComM module release notes**

- ▶ AUTOSAR R4.1 Rev 3
- ▶ AUTOSAR SWS document version: 4.1.1
- ▶ Module version: 5.19.8.B466224
- ▶ Supplier: Elektrobit Automotive GmbH



### 3.3.3.1. Change log

This chapter lists the changes between different versions.

#### Module version 5.19.8

2021-10-27

- ▶ ASCCOMM-1416 Fixed known issue: Messages with all partial network bytes set to zero are sent on the bus for a channel with attribute ComMPncGatewayType set to COMM\_GATEWAY\_TYPE\_PASSIVE and ComM0PncVectorAvoidance set to TRUE

#### Module version 5.19.7

2021-05-28

- ▶ Added Support for handling of wakeup events for communication channels from different cores
- ▶ Added support for LIN slave in ComM

#### Module version 5.19.6

2021-03-05

- ▶ Added support for ComM\_GetCurrentPNCComMode
- ▶ Improved configuration constraints for ComMPNCGatewayType
- ▶ Added configuration constraints for managed ComM channels

#### Module version 5.19.5

2020-10-23

- ▶ Improved ActiveWakeupBit handling when PNC is requested within received NM message
- ▶ Added support for Ecu Reset initiated by ComM

#### Module version 5.19.4

2020-06-19

- ▶ ASCCOMM-1233 Fixed known issue: Compilation error is reported when multicore is enabled, number of configured PNCs is less than two and only one ComM channel is used

#### Module version 5.19.3

2020-02-21



- ▶ ASCCOMM-1198 Fixed known issue: A ComM channel configured with ComMNmVariant set to NONE and ComMBusType different than COMM\_BUS\_TYPE\_INTERNAL always goes to state COMM\_NO\_COMMUNICATION

#### **Module version 5.19.2**

2019-10-11

- ▶ Removed nested compiler abstraction macro
- ▶ Changed Max Pncl from 63 to 511
- ▶ ASCCOMM-1181 Fixed known issue: ComM generates a non-compliant ARXML file for AUTOSAR 4
- ▶ ASCCOMM-1182 Fixed known issue: Inhibition status is not accessible via ComM\_ChannelWakeuface
- ▶ ASCCOMM-1187 Fixed known issue: Rte is not correctly informed about full communication requesters when wake-up inhibition or mode limitation is used

#### **Module version 5.19.1**

2019-06-14

- ▶ ASCCOMM-1138 Fixed known issue: A passive wakeup can be ignored for channels that have entered COMM\_FULL\_COM\_NETWORK\_REQUESTED and mode limitation is enabled
- ▶ ASCCOMM-1147 Fixed known issue: ComM could enter COMM\_FULL\_COM\_READY\_SLEEP while a user request is set
- ▶ ASCCOMM-1148 Fixed known issue: ComM could enter COMM\_NO\_COMMUNICATION while a user request is set
- ▶ ASCCOMM-1163 Fixed known issue: ComM fails to compile if multi-core is enabled and all ComM channels use the ComMNmVariant NONE or LIGHT
- ▶ Added Support for ComM0PncVectorAvoidance
- ▶ ASCCOMM-1166 Fixed known issue: ComM channel may fail to shut down when ComMDcmUsage and ComMModeLimitationEnabled are enabled

#### **Module version 5.19.0**

2019-02-15

- ▶ Changed structure of generated templates for post-build selectable support

#### **Module version 5.18.1**

2019-01-31



- ▶ Improved robustness check for references, optional parameters property and enable parameters property
- ▶ ASCCOMM-1068 Fixed known issue: ComM PNC identifiers are not correctly imported from system description

#### **Module version 5.18.0**

2018-10-26

- ▶ Implemented Multi-core support
- ▶ ASCCOMM-1087 Fixed known issue: Zero size array is generated for a gateway without ERA signals received on active channels
- ▶ ASCCOMM-1090 Fixed known issue: ComM generates an invalid basic software module description if no configuration set is provided
- ▶ ASCCOMM-1072 Fixed known issue: Wrong Pn information is sent on the bus if <Bus>Nm configures PnInfoOffset greater than 1
- ▶ Support for VLANs by means of managed and managing ComM channels
- ▶ ASCCOMM-1101 Fixed known issue: ComM\_RteNotificationDoActFkp is not defined in correct memory section

#### **Module version 5.17.5**

2018-06-22

- ▶ ComMMode-ModeDeclarationGroup add support for ALPHABETIC\_ORDER
- ▶ Added memory sections for the ComM software component description
- ▶ Updated ComM ComMPnId to be according to Asr4.3 Standard
- ▶ ASCCOMM-1004 Fixed known issue: Gateway ECU transmits incorrect PN information

#### **Module version 5.17.4**

2018-02-16

- ▶ ASCCOMM-992 Fixed known issue: The network management variant NONE is not available for LIN channels
- ▶ ASCCOMM-995 Fixed known issue: Root configuration alias always generated with ComMConfigSet\_0

#### **Module version 5.17.3**

2017-12-15



- ▶ Improved Limit to COMM\_NO\_COMMUNICATION mode
- ▶ Inverted logic for AUTOSAR 4.0.2 and AUTOSAR 3.x symbolic names
- ▶ Post-build selectable support
- ▶ EcuM PNC wakeup indication don't wakeup all PNC if ComMSynchronousWakeUp is set TRUE
- ▶ ASCCOMM-985 Fixed known issue: Variable ComMPncMainfunctionRunning is not assigned to a memory section

#### **Module version 5.17.2**

2017-09-22

- ▶ Implemented Comm PNC wake-up indication extension
- ▶ ASCCOMM-935 Fixed known issue: Comm PNC state machine can change the state based on outdated PN information

#### **Module version 5.17.1**

2017-06-30

- ▶ Added support for Ethernet Switch Port Groups

#### **Module version 5.17.0**

2017-03-31

- ▶ ASCCOMM-864 Fixed known issue: Code generation error for specific configuration
- ▶ ASCCOMM-893 Fixed known issue: Comm state machine may get locked if its interfaces are called from different contexts
- ▶ Improved Comm could reject user requests even if the requests should be accepted in the current state
- ▶ ASCCOMM-897 Fixed known issue: The value of ComMPncPrepareSleepTimer could be inconsistent
- ▶ Removed duplicated REFINED\_MODULE\_DEF

#### **Module version 5.16.0**

2016-10-31

- ▶ Generate RTE\_MODE symbols using explicit values instead of lexicographical order
- ▶ ASCCOMM-872 Fixed known issue: Comm\_DCM\_ActiveDiagnostic and Comm\_DCM\_InactiveDiagnostic can produce out of bounds memory access

**Module version 5.15.0**

2016-05-24

- ▶ ASCCOMM-838 Fixed known issue: Inconsistent state between ComM and BusNm
- ▶ ASCCOMM-840 Fixed known issue: ServiceNeedsWizard generates NvM block NVM\_BLOCK\_COMM independant of ComMGlobalNvMBlockDescriptor
- ▶ ASCCOMM-842 Fixed known issue: Code does not compile in case a single PNC is configuured and ComMPncGatewayEnabled is set to true

**Module version 5.14.0**

2016-02-10

- ▶ Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeader-File`

**Module version 5.13.0**

2015-11-06

- ▶ Internal module improvement. This module version update does not affect module functionality

**Module version 5.12.0**

2015-07-28

- ▶ ASCCOMM-795 Fixed known issue: Invalid xdm checks regarding the need of ComMPncComSignal elements

**Module version 5.11.0**

2015-06-24

- ▶ Improved initialization of RxEiraMerged

**Module version 5.10.0**

2015-02-20

- ▶ Limit the ComMNmVariant values according to NmPassiveModeEnabled
- ▶ ASCCOMM-761 Fixed known issue: Invalid configuration in case of a single passive channel



- ▶ Fixed compiler warning regarding assignment of a variable

### **Module version 5.9.0**

2015-01-07

- ▶ Recompute the block-size calculation of the ComM ServiceNeeds
- ▶ Replace enums with specific defines

### **Module version 5.8.0**

2014-10-02

- ▶ ASCCOMM-681 Fixed known issue: Invalid filtering of EIRA Tx Signals
- ▶ ASCCOMM-694 Fixed known issue: PNC, which was not requested by a ComM user, keeps the bus awake
- ▶ ASCCOMM-697 Fixed known issue: Invalid handling of multiple EIRA Signals
- ▶ Fixed an inconsistency in the AUTOSAR specifications 4.0.3 and 4.1 regarding partial networking that can lead to a livelock situation that prevents buses going to sleep according to Rfc #64979
- ▶ ASCCOMM-726 Fixed known issue: Wrong description of ComMPncReleaseChannel
- ▶ DCC compiler warnings
- ▶ Redesign channel state machine and PNC state machine for ComM
- ▶ Protect variables using Exclusive Areas where necessary
- ▶ Partial Network Gateway Coordinator
- ▶ ASCCOMM-746 Fixed known issue: Compile error occurs in case a single PNC is configured

### **Module version 5.7.0**

2014-04-25

- ▶ ASCCOMM-671 Fixed known issue: PNC, which was not requested by a ComM user, keeps the bus awake
- ▶ ASCCOMM-663 Fixed known issue: PreventWakeUp inhibition cannot be set while channel is on FULL\_-COM
- ▶ ASCCOMM-692 Fixed known issue: Fix typo in label of ComMPncComSignalDirection parameter

### **Module version 5.6.1**

2014-01-27

- ▶ ASCCOMM-652 Fixed known issue: Enabling parameter ComMRelocatablePbcfgEnable leads to a compiler error if BswM is also configured for PostBuild



## Module version 5.5.2

2014-01-14

- ▶ ASCCOMM-625 Fixed known issue: `ComM_GetMaxComMode()` returns an incorrect mode when wake-up inhibition is active
- ▶ ASCCOMM-626 Fixed known issue: ComM channel main function gets stuck in an endless loop if the total number of Rx EIRA Com signals is greater than 255
- ▶ ASCCOMM-628 Fixed known issue: Concurrent access to internal variables leads to unexpected behavior
- ▶ Implemented support of VARIANT-POST-BUILD for ComM
- ▶ ASCCOMM-638 Fixed known issue: Wrong type definition of `ComM_ASR40_UserHandleType` in AUTOSAR 3.2 projects

## Module version 5.5.1

2013-10-11

- ▶ ASCCOMM-581 Fixed known issue: Parameters `ComMPncComSignalRef` and `ComMBusSMRequestRetryCount` throw errors when *Config time feature* is enabled
- ▶ ASCCOMM-577 Fixed known issue: Possible deadlock of a PNC in `PNC_READY_SLEEP` state
- ▶ ASCCOMM-514 Fixed known issue: A channel which is woken-up passively may shut down unexpectedly
- ▶ ASCCOMM-602 Fixed known issue: Mode Switch Points are not generated for users connected to a channel via PNCs

## Module version 5.5.0

2013-06-26

- ▶ Added AUTOSAR 3.2 support of Rte Interface and SWCD
- ▶ ASCCOMM-441 Fixed known issue: Multiple Com signals cannot be used to communicate PNC status information
- ▶ Improved the robustness of the channel/PNC state machine designs by revising the event handling; removed configuration parameter `ComMEventQueueSize`
- ▶ ASCCOMM-467 Fixed known issue: Invalid error during configuration: The length of the referenced Rx `ComSignal` does not match the size of EIRA stored in ComM
- ▶ ASCCOMM-499 Fixed known issue: A channel with `ComMNmVariant = PASSIVE` cannot be mapped to a PNC
- ▶ ASCCOMM-516 Fixed known issue: If the values of `ComMPncIds` are not configured in ascending order, requesting or releasing a PNC may cause a state change in a different PNC
- ▶ ASCCOMM-521 Fixed known issue: The same channel cannot be referenced by a user via different PNCs



- ▶ ASCCOMM-492 Fixed known issue: ComM does not leave `SILENT_COMMUNICATION` mode
- ▶ ASCCOMM-428 Fixed known issue: If a user requests full communication after a network channel is released, ComM and Nm may get out of synchronization
- ▶ ASCCOMM-510 Fixed known issue: `ComM_GetRequestedComMode()` returns `COMM_NO_COMMUNICATION` even though the last requested mode is `COMM_FULL_COMMUNICATION`
- ▶ ASCCOMM-548 Fixed known issue: Unexpected module behavior if `ComM_Init()` is preempted by another ComM API
- ▶ ASCCOMM-543 Fixed known issue: A faulty user request that is incorrectly stored may keep the bus awake after mode limitation is disabled
- ▶ ASCCOMM-550 Fixed known issue: Compiler warnings due to too large shift counts on 16-bit architecture

### Module version 5.4.0

2013-02-19

- ▶ ASCCOMM-456 Fixed known issue: ComM never requests `NO_COMMUNICATION` at BusSM modules in case `FULL_COMMUNICATION` mode cannot be reached
- ▶ ASCCOMM-445 Fixed known issue: User request of `COMM_FULL_COMMUNICATION` is not granted if inhibition gets disabled
- ▶ Added sanity checks to ensure that the value of configuration parameter `ComMDcmUsage` matches existence of the Dcm module in the project
- ▶ ASCCOMM-463 Fixed known issue: Compiler warnings due to undefined symbol `COMM_DCM_ENABLED` in `ComM_Dcm.h`
- ▶ Corrected the schema structure by adding mandatory container `ComMConfigSet`
- ▶ Memory allocation keywords were implemented in compliance to AUTOSAR 4.0.3
- ▶ ASCCOMM-465 Fixed known issue: Compiler errors when symbolic names according to AUTOSAR 4.-0.3 are used
- ▶ ASCCOMM-390 Fixed known issue: Channel with Nm Variant set to `LIGHT` does not go to `COMM_NO_COMMUNICATION` mode if Nm Light Timeout is configured as 0

### Module version 5.3.0

2012-10-12

- ▶ ASCCOMM-361 Fixed known issue: ComM state machine never leaves the substate `COMM_NO_COMM_REQUEST_PENDING` if BusSM module does not indicate `COMM_FULL_COMMUNICATION` or a user requests `COMM_NO_COMMUNICATION`
- ▶ ASCCOMM-384 Fixed known issue: Communication channel is shut down even if *Full Communication* is requested when PNC support is enabled and only one channel is configured



- ▶ ASCCOMM-351 Fixed known issue: ComM prohibits inhibiting a channel when active diagnostic session is indicated for another channel
- ▶ ASCCOMM-371 Fixed known issue: Memory access violation when length of the `ComSignal` to receive EIRA is greater than the size of EIRA or `ComMPncId` is greater than the number of PNCs configured
- ▶ Removed obsolete DET error reporting of error code `COMM_E_ERROR_IN_PROV_SERVICE`
- ▶ ComM-Dcm interface made optional
- ▶ ASCCOMM-300 Fixed known issue: Software component might not get notified about mode switch
- ▶ ASCCOMM-356 Fixed known issue: Undefined symbol names for `ComM_ModeType`
- ▶ The top-level structure of the software-component description in the ARXML files changed from `/AU-TOSAR/ComM` to `/AUTOSAR_ComM`
- ▶ ASCCOMM-342 Fixed known issue: ComM state machine gets stuck in `FULL_COMMUNICATION` mode
- ▶ ASCCOMM-398 Fixed known issue: ComM may fall back to state *No communication* during processing of a *Full communication* request due to pending bus sleep mode event

### Module version 5.2.1

2012-06-27

- ▶ ASCCOMM-350 Fixed known issue: Wake up inhibition gets active for a channel although it is in state `COMM_FULL_COMMUNICATION` in contrast to requirement [ComM219]
- ▶ ASCCOMM-364 Fixed known issue: Memory section not closed (correctly for the variable `ComM_Pnc-Timer`)
- ▶ Implementation of support for bus type `Eth`
- ▶ ASCCOMM-362 Fixed known issue: Declaration of ComM Com call back function is unavailable if RTE usage is ON

### Module version 5.2.0

2012-03-23

- ▶ ASCCOMM-268 Fixed known issue: Timings `ComMTMinFullComModeDuration` and `ComMNmLight-Timeout` might vary between different channels and also be shorter than expected time
- ▶ ASCCOMM-267 Fixed known issue: Implement the return feature `COMM_E_UNINIT` in all ComM APIs
- ▶ ASCCOMM-289 Fixed known issue: Function `ComM_RequestComMode()` incorrectly returns `E_OK` during an active mode limitation
- ▶ ASCCOMM-297 Fixed known issue: ComM state machine gets stuck and therefore may never release the run request from EcuM
- ▶ ASCCOMM-308 Fixed known issue: Usage of Nm related inclusions/functions without using Nm



- ▶ ASCCOMM-314 Fixed known issue: ComM doesn't grant requested communication modes if inhibitions are deactivated

### Module version 5.1.0

2011-08-29

- ▶ Initial AUTOSAR 4.0 version

#### 3.3.3.2. New features

- ▶ No new features have been added since the last release.

#### 3.3.3.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- ▶ Code size optimization

To minimize the code size for simple module configurations the ComM makes use of preprocessor conditionals removing if-branches, loops, and array look-ups wherever possible.

Rationale:

This enhancement implements partly the HIS requirements concerning code size: HisComM0001, HisComM0002, HisComM0003, HisComM0004, HisComM0005.

- ▶ Configurable parameter `ComMPncReleaseChannel` used for immediately stopping transmission of Nm messages after all PNCs of a channel are released

In case `ComMPncReleaseChannel` is `COMM_NO_INTERNAL_REQUESTS`, all Full Communication requests are released after all PNCs of a channel leave the `PNC_REQUESTED` state.

In order to maintain compatibility with the previous version, the default value of the `ComMPncReleaseChannel` is `COMM_NO_INTERNAL_AND_EXTERNAL_REQUESTS`. In this case, all Full Communication requests are released when all PNCs of a channel enter the `PNC_NO_COMMUNICATION` state.

Rationale:

AUTOSAR 4.0.3 does not specify the exact moment when the PNC Full Communication requests shall be released. In order to stay forward compatible, the previous implementation used the new requirements from AUTOSAR 4.1 which specified that all Full Communication requests would be released in the `PNC_NO_COMMUNICATION` state.



However, in case of external requests, this approach sometimes leads to livelock situation where the bus never goes to sleep and all ECU's stay awake. In this case, the best solution is to release the Full Communication requests after leaving the `PNC_REQUESTED` state. The `ComMPncReleaseChannel` parameter allows to switch between the two behaviors.

This issue is currently discussed on the AUTOSAR Bugzilla Rfc #63643.

- ▶ Extended the range for PNC ids

Description:

The range for PNC ids changed from 8..63 to 8..255

Rationale:

Since in BusNms the `PnInfoOffset` changed to 1..31 and because the PNC id is the absolute position in the PDU, to be able to use offsets greater than 8 bytes PNC id range has been extended.

#### 3.3.3.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

- ▶ Communication mode indication and BusSM mode

Description:

According to ComM083 and ComM084, a call of `ComM_GetCurrentComMode()` shall be propagated to the underlying Bus State Manager (`BusSM_GetCurrentComMode()`) and the mode retrieved from the BusSM shall be indicated to the upper layer as the current communication mode. In contrast to these requirements, ComM does not propagate a call of `ComM_GetCurrentComMode()` to the underlying Bus State Manager in case multicore is enabled. Instead, ComM indicates the communication mode based on the stored communication mode which is previously indicated by the underlying Bus State Manager.

Rationale:

This approach is chosen due to the fact that ComM and BusSm can be on different cores and a call to `BusSM_GetCurrentComMode()` would lock also the core on which the BusSM is.

Requirements:

ComM083, ComM084

- ▶ Types `ComM_InitStatusType` and `ComM_PncModeType`

Description:



The types `ComM_InitStatusType` and `ComM_PncModeType` are not enumerations as prescribed by the AUTOSAR SWS but of type `uint8`.

#### Rationale:

The use of the `uint8` type saves RAM and ROM because an enumeration value would take up 2 or 4 bytes per value depending on the CPU architecture.

#### Requirements:

ComM863 refinements: `ComM.Types_Implicit1_Refine` and `ComM.Types_Implicit6_Refine`

- ▶ No multiple network requests in state `COMM_FULL_COM_NETWORK_REQUESTED`

#### Description:

According to requirements ComM869 and ComM870, if a user requests `COMM_FULL_COMMUNICATION` or the DCM indicates *Active Diagnostic* in substate `COMM_FULL_COM_NETWORK_REQUESTED` and configuration parameter `ComMNmVariant` is set to `FULL`, the ComM module shall invoke `Nm_NetworkRequest()` for the corresponding NM channel(s). The deviation is present for AUTOSAR 4.0.3

In contrast to this, `Nm_NetworkRequest()` is not called if a user requests `COMM_FULL_COMMUNICATION` or the DCM indicates *Active Diagnostic* and the channel is already in substate `COMM_FULL_COM_NETWORK_REQUESTED`. Instead, `Nm_NetworkRequest()` is called only once when the substate `COMM_FULL_COM_NETWORK_REQUESTED` is entered after a user requests `COMM_FULL_COMMUNICATION` or the DCM indicates *Active Diagnostic* in state `COMM_NO_COM_NO_PENDING_REQUEST`.

#### Rationale:

The network is already requested once when the substate `COMM_FULL_COM_NETWORK_REQUESTED` is entered after a user requests `COMM_FULL_COMMUNICATION` or the DCM indicates *Active Diagnostic* in state `COMM_NO_COM_NO_PENDING_REQUEST`. Therefore it does not make sense to request the network again each time a user requests `COMM_FULL_COMMUNICATION` unless due to ComM980. Besides, there is no textual requirement demanding network to be requested *on entering* substate `COMM_FULL_COM_NETWORK_REQUESTED`. This issue is reported to AUTOSAR in [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=59575](http://www.autosar.org/bugzilla/show_bug.cgi?id=59575).

#### Requirements:

ComM869, ComM870

- ▶ Inclusion of header files `CanSM.h` and `Dcm.h`

#### Description:



According to requirement ComM506, ComM shall include the header files `CanSM.h` and `Dcm.h`. In contrary to this, ComM does not include the header files `CanSM.h` and `Dcm.h` but includes `CanSM_ComM.h` and `Dcm_Cbk.h` instead. The deviation is present for AUTOSAR 4.0.3.

#### Rationale:

Requirement CANSM008 in the AUTOSAR specification of the CAN State Manager specifies that `CanSM.h` does not provide the declarations of API functions `CanSM_GetCurrentComMode()` and `CanSM_RequestComMode()`. Additionally, requirement CANSM009 specifies that the header file `CanSM_ComM.h` exports the APIs dedicated to the ComM module.

Similarly, requirement [Dcm110] in the AUTOSAR specification of the Diagnostic Communication Manager specifies that `Dcm.h` contains only those types, functions and parameters that are visible to SWCs. Additionally, the callback APIs `Dcm_ComM_NoComModeEntered()`, `Dcm_ComM_SilentComModeEntered()` and `Dcm_ComM_FullComModeEntered()` are specified in section 8.4 *Callback Notifications* of the same document with the description that the function prototypes of these callback APIs are provided in the file `Dcm_Cbk.h`.

This issue has been fixed in R4.1.1 of the Specification of Communication Manager where requirement SWS\_ComM\_00506 specifies that ComM shall include the header files `Dcm_Cbk.h` and `CanSM_ComM.h`.

#### Requirements:

##### ComM506

- ▶ Behavior of `ComM_Nm_BusSleepMode()`

#### Description:

A call of `ComM_Nm_BusSleepMode()` shall perform the transition of the hardware and transceiver to bus-sleep mode in a synchronous way according to the AUTOSAR ComM SWS. In contrast to that the EB ComM performs this operation asynchronously during the main function call. Deviation is present for AUTOSAR 4.0.3

#### Rationale:

The synchronous handling of these actions require locking of global resources, is error-prone, and inefficient with respect to run-time and code size.

#### Requirements:

##### ComM392

- ▶ Availability of `ComM_SetECUGroupClassification()`

#### Description:



The API function `ComM_SetECUGroupClassification()` is only available if either the configuration parameter `ComMWakeupInhibitionEnabled` or `ComMModeLimitationEnabled` is set to true. This contradicts requirement ComM552 in which no availability condition is defined.

Rationale:

The API function `ComM_SetECUGroupClassification()` will never be needed for module configurations with `ComMWakeupInhibitionEnabled = false` and `ComMModeLimitationEnabled = false`.

Requirements:

ComM552

- ▶ Correction: Diagnostic requests ignored for *PASSIVE* channels

Description:

A call of `ComM_DCM_ActiveDiagnostic()` in state `COMM_NO_COM_NO_PENDING_REQUEST` or `COMM_SILENT_COMMUNICATION` or `COMM_FULL_COM_READY_SLEEP` shall trigger the necessary transitions towards `COMM_FULL_COM_NETWORK_REQUESTED` state according to the requirements ComM876, ComM878, and ComM883 respectively. In contrast to these requirements, the call of `ComM_DCM_ActiveDiagnostic()` for a channel with `ComMNmVariant = PASSIVE` is simply ignored and hence does not trigger any expected transitions. Deviation is present for AUTOSAR 4.0.3

Rationale:

*Active Diagnostic* is not possible on a *PASSIVE* channel since it is not possible to keep the channel awake. This problem is reported to AUTOSAR in [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=56661](http://www.autosar.org/bugzilla/show_bug.cgi?id=56661).

Requirements:

ComM876, ComM878, ComM883

- ▶ No AUTOSAR Debugging support

Description:

ComM is not instrumented for the usage with AUTOSAR Debugging. Deviation is present for AUTOSAR 4.0.3

Rationale:

Debugging support according to the AUTOSAR standard is not planned.

Requirements:

ComM850, ComM851



- ▶ Symbolic port name support

Description:

The port names provided by the ComM are not named by their numeric index of the configured ComM user or ComM channel as suggested by the AUTOSAR ComM SWS in section 7.14. Instead the ports are postfixed by the symbolic name of the configured ComM user or ComM channel.

Rationale:

With symbolic names port names do not change when ports are deleted or inserted and renumbered. Therefore ports must not be reconnected.

Requirements:

ComM662

- ▶ Limited HIS Optimizations

Description:

According to HIS requirements (HisComM0001, HisComM0002, HisComM0003, HisComM0004 and HisComM0005), if only one single `ComMChannel` is configured, the module code should be optimized. That is, in the above condition the ComM module should do the following:

- ▶ Make use of the `ComMChannel` configuration directly in the code.
- ▶ Not make branches at runtime depending on this configuration data.
- ▶ Avoid to separately store this configuration data.
- ▶ Not contain any code for the user to channel mapping like switch blocks, mapping tables, callback lists.
- ▶ Not use the indirection to access data indexed by the user ID.

In contrast to these requirements, the above optimizations are not done for the parameters `ComMNoWake-up`, `ComMNoFullCom`, and `ComMNoCom`.

Requirements:

HisComM0001, HisComM0002, HisComM0003, HisComM0004 and HisComM0005

- ▶ Variant `VARIANT-PRE-COMPIL` is not supported.

Description

The ComM supports the variant `VARIANT-POST-BUILD` only. The implementation of variant `VARIANT-PRE-COMPIL` is not required since all functionality of `VARIANT-PRE-COMPIL` is implemented by `VARIANT-POST-BUILD`.

- ▶ Two identical service ids across requirements from 4.0.3 and R20-11.



#### Description

ComM\_GetState service id defined in 4.0.3 is 0x34 (ComM872). The same id is for ComM\_BusSM\_-BusSleepMode in R20-11 (SWS\_ComM\_91000). Function ComM\_GetState was removed in R20-11 but implementation shall remain compatible with 4.0.3.

#### 3.3.3.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- ▶ For this module no limitations are known.

#### 3.3.3.6. Open-source software

ComM does not use open-source software.

### 3.3.4. EcuM module release notes

- ▶ AUTOSAR R4.0 Rev 3
- ▶ AUTOSAR SWS document version: 3.0.0
- ▶ Module version: 5.15.8.B466224
- ▶ Supplier: Elektrobit Automotive GmbH

#### 3.3.4.1. Change log

This chapter lists the changes between different versions.

##### Module version 5.15.8

2021-10-27

- ▶ Added EcuM\_Externals.h file required by ASR 4.4.0.

##### Module version 5.15.7

2021-06-25

- ▶ Added support for Handling of wakeup events for ComM communication channels from different cores.



### Module version 5.15.6

2021-03-05

- ▶ Internal module improvement. This module version update does not affect module functionality.

### Module version 5.15.5

2020-10-23

- ▶ Internal module improvement. This module version update does not affect module functionality.

### Module version 5.15.4

2020-06-19

- ▶ Internal module improvement. This module version update does not affect module functionality.

### Module version 5.15.3

2020-02-21

- ▶ Internal module improvement. This module version update does not affect module functionality.

### Module version 5.15.2

2019-10-11

- ▶ ASCECUM-1169 Fixed known issue: Linker error may occur because of incomplete memory mapping of EcuM\_DriverInitListOneCall() and EcuM\_DriverRestartListCall().
- ▶ ASCECUM-1181 Fixed known issue: EcuM may fail to start the OS because an invalid OsAppMode is used.
- ▶ Removed AUTOSAR 3.x symbolic name value macros and inverted logic to only provide AUTOSAR 4.0.-2 compliant macros if ECUM\_PROVIDE\_LEGACY\_SYMBOLIC\_NAMES is defined.

### Module version 5.15.1

2019-06-14

- ▶ ASCECUM-1160 Fixed known issue: Linker error may occur because of incomplete memory mapping of EcuM\_SchM\_Init() and EcuM\_SchM\_Deinit().
- ▶ ASCECUM-1161 Fixed known issue: Wrong MemMap INIT\_POLICY for EcuM global variables may cause undefined behavior because of unexpected return value.



### **Module version 5.15.0**

2019-03-28

- ▶ Added support for Post-build variant handling.

### **Module version 5.14.4**

2019-02-15

- ▶ Internal module improvement. This module version update does not affect module functionality.

### **Module version 5.14.3**

2018-10-26

- ▶ Internal module improvement. This module version update does not affect module functionality.
- ▶ Added configuration parameter to call `StartCore()` also for the core that boots up the system

### **Module version 5.14.2**

2018-06-22

- ▶ Added memory sections for the EcuM software component description
- ▶ ASCECUM-590 Fixed known issue: Validated wakeup events will be ignored when set before entering the SLEEP state
- ▶ ASCECUM-1088 Fixed known issue: Wrong EcuM instance ID reported to Det if defensive programming is enabled
- ▶ ASCECUM-1011 Fixed known issue: Uncompilable code if EcuM wakeup sources are configured only with a ComM PNC reference

### **Module version 5.14.1**

2018-02-16

- ▶ Internal module improvement. This module version update does not affect module functionality.

### **Module version 5.14.0**

2017-09-22

- ▶ Added support for Ethernet Switch Port Groups
- ▶ Added support for BswM notifications of each EcuM state change.

**Module version 5.13.0**

2017-03-31

- ▶ ASCECUM-923 Fixed known issue: EcuM does not inform BswM about wakeup state changes if the StartupTwo sequence is interrupted

**Module version 5.12.1**

2017-03-10

- ▶ ASCECUM-908 Fixed known issue: The single-core EcuM never invokes EcuM\_AL\_DriverRestart
- ▶ ASCECUM-899 Fixed known issue: EcuM wrongfully enters the Sleep state if the GoSleep sequence is interrupted

**Module version 5.12.0**

2016-11-04

- ▶ Enhanced the basic software module description
- ▶ ASCECUM-876 Fixed known issue: The slave-core EcuM may not perform all of its activities during ECU shutdown
- ▶ StartCore will not be called for the already started initial core
- ▶ The multicore EcuM supports now a core synchronization safety mechanism
- ▶ Fixed issue with exporting unused configuration parameters

**Module version 5.11.0**

2016-05-25

- ▶ ASCECUM-852 Fixed known issue: The single-core EcuM does not report the RamHash result to Dem
- ▶ ASCECUM-855 Fixed known issue: DET errors are always reported from the first active core

**Module version 5.10.1**

2016-04-15

- ▶ ASCECUM-848 Fixed known issue: The multicore EcuM does not complete the ECU shutdown and sleep operations
- ▶ The multicore EcuM supports now a configurable designation of its master core

**Module version 5.10.0**

2016-02-10

- ▶ ASCECUM-821 Fixed known issue: The definition of the ECUM\_SLAVE\_SYNC\_MASK may contain invalid characters
- ▶ Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeader-File`

**Module version 5.9.1**

2016-01-15

- ▶ Adapted the usage of the OsResource RES\_SCHEDULER.
- ▶ ASCECUM-806 Fixed known issue: Compilation fails when the OS module doesn't provide the OS\_CORE\_ID\_MASTER macro.
- ▶ ASCECUM-803 Fixed known issue: EcuM master-slave synchronization is missing during EcuM\_GoPoll()
- ▶ Handled return value of SchM Sender-Receiver APIs used in EcuM APIs

**Module version 5.9.0**

2015-11-10

- ▶ Added support for master-satellite EcuM multi-core

**Module version 5.8.0**

2015-06-24

- ▶ EcuM.c has been changed from a generated file to a static file.
- ▶ Implemented the EcuM multi-core startup and shutdown sequences.
- ▶ ASCECUM-764 Fixed known issue: EcuM Fixed Modes are available even if EcuM Fixed Support is disabled.

**Module version 5.7.0**

2015-02-20

- ▶ Fixed warnings present during SystemDescriptionImport within EcuM files

**Module version 5.6.0**

2015-01-07



- ▶ Fixed MISRA violations

#### **Module version 5.5.0**

2014-10-02

- ▶ ASCECUM-687 Fixed known issue: The EcuM API function `EcuM_SelectShutdownTarget` may not be defined
- ▶ The attribute `postBuildChangeable` has been defined for all EcuC containers that can have multiple entries
- ▶ Created service needs requests for Dem events
- ▶ Defined `RunnableEntityArgument` elements for functions related to the EcuM Run Request API.
- ▶ ASCECUM-685 Fixed known issue: Symbolic name constants for EcuM wakeup sources are incorrectly generated by the EcuM.

#### **Module version 5.4.9**

2014-06-13

- ▶ ASCECUM-664 Fixed known issue: Link-time errors for EcuM data types may occur if compiler optimizations are used

#### **Module version 5.4.8**

2014-04-25

- ▶ ASCECUM-676 Fixed known issue: Wakeup sources in the `PENDING` state at the time of an ECU reset cannot be restarted
- ▶ ASCECUM-688 Fixed known issue: The header file `EcuM_Cbk.h` is not included by the `EcuM.h`

#### **Module version 5.4.7**

2014-03-21

- ▶ ASCECUM-653 Fixed known issue: EcuM may generate the wrong definition for the constant `ECUM_NORMALMCUMODE`
- ▶ Updated the basic software module description for Debug & Trace support
- ▶ ASCECUM-661 Fixed known issue: Port-defined argument values are not defined for the `EcuM_StateRequest` API
- ▶ ASCECUM-668 Fixed known issue: The EcuM notifies the wrong mode when the state `PREP_SHUTDOWN` is entered



- ▶ ASCECUM-670 Fixed known issue: Invalid warning about the parameter `EcuMModuleInitConfigStr` when the `PreInit` service is configured
- ▶ ASCECUM-672 Fixed known issue: The event `ECUM_E_ALL_RUN_REQUESTS_KILLED` is reported to DEM even if it is not enabled in the EcUM configuration
- ▶ ASCECUM-673 Fixed known issue: The EcUM may not start wakeup sources that have been previously cleared

#### Module version 5.4.6

2013-09-17

- ▶ ASCECUM-640 Fixed know issue: The EcUM source code may not compile when EcUM fixed support is enabled
- ▶ Added description of production errors to the module reference documentation

#### Module version 5.4.5

2013-08-20

- ▶ Added a range check for the configuration parameter `EcuMValidationTimeout`
- ▶ ASCECUM-632 Fixed know issue: The EcUM passes the wrong Rte modes when the ASR32 and ASR40 service APIs are enabled
- ▶ ASCECUM-635 Fixed know issue: A preprocessor error may be generated if an application header file is included before the `EcuM.h`

#### Module version 5.4.4

2013-06-21

- ▶ EcUM wakeup source status changes are now implemented within an internal state machine
- ▶ Refactored the function `EcuM_GoHaltOrPoll` into the functions `EcuM_GoHalt`, `EcuM_GoPoll`, `EcuM_GoSleep`, and `EcuM_WakeupRestart`
- ▶ Refactored the handling of `RUN` and `POST_RUN` requests within the `EcuM_MainFunction` into the function `EcuM_HandleRunRequest`
- ▶ ASCECUM-601 Fixed known issue: When the EcUM is in a substate of `RUN`, events for non-communication channels will not be ignored
- ▶ Added missing Det checks for `ECUM_E_UNINIT`
- ▶ Removed references of symbolic name parameters by the ECU configuration short name
- ▶ Added AUTOSAR 3.2 support of Rte Interface and SWCD
- ▶ Added AUTOSAR 3.2 support of Rte Interface and SWCD



- ▶ Added Debug & Trace support for the EcuM

#### Module version 5.4.3

2013-05-15

- ▶ ASCECUM-580 Fixed known issue: A validated wakeup event will not be returned by `EcuM_GetValidatedWakeupEvent` after the `EcuM_MainFunction` has been executed
- ▶ The EcuM mode switch port `currentMode` now defines a queue length
- ▶ ASCECUM-582 Fixed known issue: The function `EcuM_SelectShutdownTarget` does not ignore invalid parameters when development error detection is disabled
- ▶ ASCECUM-581 Fixed known issue: The function `EcuM_ClearWakeupEvent` does not clear validated and expired wakeup events
- ▶ ASCECUM-569 Fixed known issue: Wakeup events that are set while transitioning to the `SLEEP` state are incorrectly set to expired
- ▶ ASCECUM-583 Fixed known issue: Wakeup event handling during the initialization of the EcuM is not supported
- ▶ ASCECUM-578 Fixed known issue: The EcuM may incorrectly signal the BswM that a pending wakeup event has been cleared
- ▶ ASCECUM-571 Fixed known issue: When the `SLEEP` phase is entered, the EcuM will signal the BswM that all wakeup sources are in the state `ECUM_WKSTATUS_NONE`
- ▶ Merged the files `EcuM_Callouts.c` and `EcuM_CalloutList.c` into the file `EcuM_Callout_Stubs.c`
- ▶ Default implementations of `EcuMDriverInitListZero`, `EcuMDriverInitListOne`, and `EcuMDriverRestartList` are now defined within the file `EcuM_Cfg.c`

#### Module version 5.4.2

2013-04-16

- ▶ ASCECUM-528 Fixed known issue: ECU cannot be woken up by peripherals other than communication channels
- ▶ ASCECUM-566 Fixed known issue: The EcuM may not compile as it may reference undefined constants within the file `EcuM_Cfg.c`
- ▶ Reworked the inclusion of the header file `Rte_EcuM_Types.h`
- ▶ ASCECUM-567 Fixed known issue: The BswM is not notified when the validation of a wakeup source fails

#### Module version 5.4.1

2013-03-18



- ▶ ASCECUM-546 Fixed known issue: The EcuM service component description may contain invalid reference values
- ▶ ASCECUM-552 Fixed known issue: The EcuM may report compiler warnings as it may not include `Rte_Main.h` when Rte support is enabled

#### Module version 5.4.0

2013-02-21

- ▶ Changed the reference path of `ComMChannel` in parameter `EcuMComMChannelRef` to `/AUTOSAR/EcucDefs/ComM/ComMConfigSet/ComMChannel`
- ▶ ASCECUM-512 Fixed known issue: `EcuM_StartWakeupSource` called multiple times
- ▶ ASCECUM-526 Fixed known issue: Compilation error due to undefined symbol for `McuModeSetting-Conf`
- ▶ ASCECUM-529 Fixed known issue: The EcuM generates the identifiers of symbolic name parameters incorrectly

#### Module version 5.3.1

2013-01-18

- ▶ ASCECUM-519 Fixed known issue: If the configured Wake-up Source IDs are not consecutive some wake-up events may not be correctly processed

#### Module version 5.3.0

2012-10-12

- ▶ ASCECUM-454 Fixed known issue: `SwcBswMapping` is located at the wrong location in BSWMD
- ▶ ASCECUM-458 Fixed known issue: Validity check of parameter `EcuMComMChannelRef` causes exception if reference is invalid
- ▶ ASCECUM-481 Fixed known issue: `EcuM_Bswmd.arxml` and `EcuM_swc_internal.arxml` are inconsistent to each other
- ▶ ASCECUM-476 Fixed known issue: Wakeup is not notified to ComM when only a single channel is configured in ComM
- ▶ The top-level structure of the software-component description in the ARXML files changed from `/AUTOSAR/EcuM` to `/AUTOSAR_EcuM`
- ▶ Removed support for multiple `EcuMConfiguration` containers
- ▶ Added a compatibility interface for basic EcuM Fixed support
- ▶ EcuM updated to AUTOSAR 4.0. Rev 3



## Module version 5.2.0

2012-03-16

- ▶ EBAECUM-269 Fixed known issue: In order to adhere to the ecuc\_sws\_6007 in AUTOSAR 4.0 Rev 2 EcuM Fixed parameter shall be added to VSMD
- ▶ Preconfigured the mapping of Mcu reset reasons to EcuM wakeup sources
- ▶ Changed the interpretation of the configuration parameter `EcuMWakeupSourceId` from a bit array value to a bit position value

## Module version 5.1.0

2011-08-26

- ▶ Initial AUTOSAR 4.0 version

### 3.3.4.2. New features

- ▶ No new features have been added since the last release.

### 3.3.4.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- ▶ `EcuM_Init()` performs DET checks

Description:

The implementation throws the DET error `ECUM_E_NULL_POINTER` and cancels the execution of `EcuM_Init()` if `EcuM_DeterminePbConfiguration()` returns `NULL`.

Rationale:

By this check a missing post-build time configuration will not lead to dereferencing a `NULL` pointer.

- ▶ Enhanced production error reporting

Description:

An enhanced production error reporting mechanism has been introduced. This allows to configure the following options independently for each Dem event:

- ▶ Report production errors to the Diagnostics Event Manager (Dem).
- ▶ Report production errors to the Development Error Tracer (Det) as development errors.
- ▶ Do not report production errors at all.



If a production error is redirected towards the `Det`, you may configure the reported `Det` error-ID.

Rationale:

This enhancement implements the HIS requirements concerning fault operation and error detection: His-Gen0007, HisGen0008 and HisGen0009.

- ▶ Support for non-default header file inclusion for initialization lists

Description:

The vendor-specific configuration parameter `EcuMModuleHeaderFile` allows to specify dedicated include files for each entry in the initialization lists.

Rationale:

Per default the AUTOSAR configuration allows the EcuM only to include the regular API headers of the AUTOSAR modules given by the parameter `EcuMModuleId`. To call `LinTp_Init()` which is declared in `LinIf.h` or to be able to integrate non-AUTOSAR modules it is necessary to specify header files with arbitrary names.

- ▶ Support for initializations of non-AUTOSAR modules in initialization lists

Description:

The vendor-specific configuration parameter `EcuMModuleInitConfigStr` allows to specify to give a configuration pointer to the initialization function of non-AUTOSAR modules.

Rationale:

Per default the AUTOSAR configuration allows only to pass pointers to module configuration referenced by `EcuMModuleConfigurationRefs` to initialization functions. With `EcuMModuleInitConfigStr` the user is able to overwrite the default behavior for any module.

- ▶ Provision of `ShutdownHook()` function

Description:

EcuM provides a minimalistic implementation of the Os callback function `ShutdownHook()`. If the EcuM shall provide this implementation it can be activated with the EB-specific configuration parameter `EcuMProvideShutdownHook`.

Rationale:

If the Os is integrated with the EcuM in a standard way, then the implementation of the `ShutdownHook()` is always the same. Using this feature will save time at integration as it is no longer necessary to provide this standard implementation explicitly.

- ▶ Support for initialization of modules with upper multiplicity greater than 1



#### Description:

EcuM supports initialization of modules which have an upper multiplicity greater than one. The initialization service shall be generated in the following format: `EcuModuleID_VendorId_VendorApiInfix_EcuModuleService>()`.

#### Rationale:

Modules with an upper multiplicity greater than 1 are usually some driver modules, e.g. Fr. For these modules initialization APIs are available in the following format: `EcuModuleID_VendorId_VendorApiInfix_EcuModuleService>()`.

### 3.3.4.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

- ▶ Time Triggered Increased Inoperation

#### Description:

The feature *Time Triggered Increased Inoperation* (TTII) is not supported.

- ▶ Configuration Consistency check (reference to product description: ASCPD-145)

#### Description:

Configuration consistency check is not implemented and the related DEM error `ECUM_E_CONFIGURATION_DATA_INCONSISTENT` is not supported.

#### Requirements:

`ECUM163_Conf`

- ▶ No support for alarm clock functionality (reference to product description: ASCPD-143)

#### Description:

All API functions related to wakeup alarm handling are not supported. These are `EcuM_SetRelWake-upAlarm()`, `EcuM_SetAbsWake-upAlarm()`, `EcuM_AbortWake-upAlarm()`, `EcuM_GetCurrent-Time()`, `EcuM_GetWake-upTime()`, `EcuM_SetClock()`.

#### Requirements:

`EcuM4002`, `EcuM4069`, `EcuM4086`, `EcuM4087`, `EcuM4088`, `EcuM4070`, `EcuM4071`, `EcuM4072`, `EcuM4073`, `EcuM4089`, `EcuM4009`, `EcuM4010`, `EcuM3013`, `EcuM4054`, `EcuM4055`, `EcuM4056`, `EcuM4057`, `EcuM4058`, `EcuM4059`, `EcuM4060`, `EcuM3019`, `EcuM4061`, `EcuM4062`, `EcuM4063`,



EcuM4064, ECUM199\_Conf, ECUM200\_Conf, ECUM184\_Conf, ECUM186\_Conf, ECUM188\_Conf, ECUM195\_Conf, ECUM201\_Conf, ECUM146\_Conf, ECUM197\_Conf, ECUM198\_Conf

- ▶ No support for defensive behavior (reference to product description: ASCPD-144)

Description:

The EcuM does not observe the defensive behavior of the calling component and ignores the function argument `caller` of the function `EcuM_GoDown()`. Hence the configuration of allowed users of `EcuM_GoDown()` (Related requirements: ECUM196\_Conf, ECUM206\_Conf, ECUM207\_Conf) is also not implemented.

Requirements:

EcuM4047, ECUM196\_Conf

- ▶ No support for multiple Driver Init Lists.

Description:

The EcuM module does not provide multiple configuration parameters for `EcuMDriverInitListZero` and `EcuMDriverInitListOne` in order for the MCAL modules to be initialized only by the master core.

Requirements:

EcuM4013

- ▶ No Det error for unspecified shutdown causes in `EcuM_SelectShutdownCause()`

Description:

`EcuM_SelectShutdownCause()` does not report DET error `ECUM_E_INVALID_PAR` for unspecified/not configured shutdown causes.

Rationale:

Shutdown causes other than default causes can be added by EcuM configuration but this implementation only supports default shutdown causes.

Requirements:

EcuM3009

- ▶ No wakeup status notification to ComM and BswM in `EcuM_SetWakeupEvent()` and `EcuM_ValidateWakeupEvent()` `EcuM_WakeupRestart()` `EcuM_GoHalt()`

Description:

The wakeup source status changes are not indicated to ComM and BswM from within the calls to the functions `EcuM_SetWakeupEvent()`, `EcuM_ValidateWakeupEvent()`, `EcuM_WakeupRestart()`



`EcuM_GoHalt()` Instead the notification calls to BswM and ComM modules are delayed and performed from `EcuM_MainFunction()`.

Rationale:

The function `EcuM_SetWakeupEvent()` may be called from an init function like those of the CanTrcv. There the calls of `CanTrcv_Init() -> EcuM_SetWakeupEvent() -> ComM_EcuM_WakeUpIndication() -> ComM_EcuM_PNCWakeUpIndication()` and `BswM_EcuM_CurrentWakeUp()` are performed before the ComM and BswM are initialized. The notifications are ignored by the ComM and BswM and a DET error may be raised. The solution is implemented as outlined in AUTOSAR Bugzilla issue [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=53338](http://www.autosar.org/bugzilla/show_bug.cgi?id=53338).

Requirements:

EcuM2645, SWS\_EcuM\_04112

- ▶ Shutdown target configuration

Description:

In contrast to requirements ECUM178\_Conf, ECUM179\_Conf, shutdown targets cannot be configured other than predefined targets `ECUM_STATE_OFF`, `ECUM_STATE_RESET`, `ECUM_STATE_SLEEP`

Rationale:

EcuM shutdown targets other than `ECUM_STATE_OFF`, `ECUM_STATE_RESET`, `ECUM_STATE_SLEEP` are not used in EcuM. The APIs which uses shutdown targets are of `EcuM_StateType` and these APIs only allow this three shutdown targets. This is addressed in AUTOSAR Bugzilla issue [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=51379](http://www.autosar.org/bugzilla/show_bug.cgi?id=51379).

Requirements:

ECUM178\_Conf, ECUM179\_Conf

- ▶ Reset loop detection

Description:

No support for reset loop detection.

Rationale:

Apart from the configuration parameter ECUM171\_Conf: `EcuMResetLoopDetection`, the specification of ECU State Manager R4.0 Rev 3 does not have requirements regarding action items for this feature. AUTOSAR Bugzilla Ref: [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=50899](http://www.autosar.org/bugzilla/show_bug.cgi?id=50899)

Requirements:

ECUM171\_Conf



- ▶ The parameter `EcuMResetReasonRef` is optional

Description:

In contrast to requirement ECUM128\_Conf, the multiplicity of the parameter `EcuMResetReasonRef` has been changed from 1 to 0..1.

Rationale:

For communication channel wakeup, the wakeup source may not be mapped to an MCU reset reason.

Ref: AUTOSAR Bugzilla [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=52629](http://www.autosar.org/bugzilla/show_bug.cgi?id=52629)

Requirements:

ECUM128\_Conf

- ▶ Error reporting to Dem is optional

Description:

According to the requirement EcuM2985, detection of production errors shall not be switched off. In contrast to this requirement, production errors are switched off by the configuration parameter `EcuMIncludeDem`

Rationale:

In AUTOSAR SWS ECU StateManager of AUTOSAR 4.0 Rev 3, the configuration parameter `EcuMIncludeDem` is used to enable/disable the availability of Dem. Also the multiplicity of a similar parameter `EcuMDemEventParameterRefs` is mentioned as 0 to 1. These specification errors are handled in the AUTOSAR Bugzilla [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=50236](http://www.autosar.org/bugzilla/show_bug.cgi?id=50236) and [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=55543](http://www.autosar.org/bugzilla/show_bug.cgi?id=55543) respectively.

Requirements:

EcuM2985

- ▶ No debugging support (reference to product description: ASCPD-68)

Description:

The feature *Debugging support* is not supported.

Requirements:

EcuM4035, EcuM4036, EcuM4037

- ▶ No provision of `EcuM_Generated_Types.h`

Description:



According to the requirement EcuM2992, the ECU Manager module implementation shall provide a `EcuM_Generated_Types.h` file which contains generated type declarations that fulfill the forward declarations in `EcuM.h`. In contrast to this requirement, `EcuM_Generated_Types.h` is not provided.

Rationale:

All the types are statically defined in this implementation.

Requirements:

EcuM2992

- ▶ No hash value computation

Description:

In contrast to section 7.3.4 Checking Configuration Consistency, the hash value computation is not performed and also it is not stored in configuration parameter `EcuMConfigConsistencyHash`.

Rationale:

The hash value computation feature has not been implemented in this release.

Requirements:

EcuM2794, EcuM2795, EcuM2796, EcuM2798, EcuM2799, ECUM102\_Conf

- ▶ No Alias for `EcuMShutdownTarget`

Description:

In contrast to requirement EcuM2959, there is no alias defined for the parameter `EcuMShutdownTarget`.

Rationale:

According to requirement [EcuM624] only `ECUM_STATE_RESET`, `ECUM_STATE_SLEEP`, and `ECUM_STATE_OFF` can be chosen as valid shutdown targets. Also the parameter (`EcuMShutdownTargetId`) for defining the alias is removed in the AUTOSAR 4.1 specification (through RFC #51379). Thus it is not possible (also not valid) to define an alias for `EcuMShutdownTarget`. This is the topic of AUTOSAR Bugzilla [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=58918](http://www.autosar.org/bugzilla/show_bug.cgi?id=58918).

Requirements:

EcuM2959

- ▶ No Default wakeup source reporting

Description:



The requirements EcuM4040 and EcuM2601 implies that when a hardware cannot detect a specific wake-up source for a reset/restart reason, then the ECU Manager module shall report `ECUM_WKSOURCE_RESET` as a default wakeup source. In contrast to these requirements, `ECUM_WKSOURCE_RESET` is not reported as a default wakeup source during `STARTUP` or `WAKEUP`, if a reset/restart reason is not found (by the hardware). This is the topic of AUTOSAR Bugzilla [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=58920](http://www.autosar.org/bugzilla/show_bug.cgi?id=58920).

#### Rationale:

This feature has not been implemented in this release.

#### Requirements:

EcuM2601, EcuM4040\_Implicit\_2

- ▶ DET initialization is not dependent on modules configured in EcuM

#### Description:

Contrary to the requirement EcuM2783, DET initialization does not depend on any of the modules configured in the EcuM. DET will be initialized only if EcuM includes DET and an init service entry of DET in any initialization list exists.

#### Rationale:

This functionality is not supported.

- ▶ AUTOSAR 3.2 support: The client-server interface `EcuM_ApplicationMode` is not provided

#### Description:

Contrary to the AUTOSAR 3.2 specification the client-server interface `EcuM_ApplicationMode` requirement that contains ports to `EcuM_SelectApplicationMode` (EcuM2833) and `EcuM_GetApplicationMode` (EcuM2834) is not provided.

#### Rationale:

This functionality is not supported.

- ▶ EcuM\_Types.h header file does not include Rte Types related header file

#### Description:

Contrary to the requirement EcuM3025 the file `EcuM_Types.h` does not include the file `Rte_EcuM_Type.h`.

#### Rationale:

When adding AUTOSAR 3.2 support to EcuM, the BSW's view of the default API has to be different to the Rte's/SWC's view of the default API. To resolve this issue distinct translation units for the BSW interface



and the Rte/SWC interface have been introduced. Thus `EcuM_Types.h` does not include `Rte_EcuM_Type.h`.

Requirements:

EcuM3025

- ▶ Disable Interrupts and Enable Interrupts is not available for multicore.

Description:

Within the Poll Sequence, WakeupRestart Sequence, all interrupts should be disabled before the `Mcu_SetMode(Mcu_ModeType)` API is called, and enabled back again.

Rationale:

This functionality is not supported.

Requirements:

`EcuM.MasterCore.Poll.Sequence`, `EcuM.SlaveCore.Poll.Sequence`, `SWS_EcuM_04026`

- ▶ Master / Slave Core Shutdown Synchronization during shutdown

Description:

The SWS specifies that a shutdown flag should be set, which can be read by all EcuMs of all slave cores. Then BSW de-initialization on slave core takes places, and the shutdown flag should be unset.

Rationale:

If a flag would be used, this means that there should be a global variable access, from different core contexts, which will result in an Os error and which may lead to cache coherency issues.

Requirements:

`SWS_EcuM_04019`, `SWS_EcuM_04021`

- ▶ Expected interfaces from CanSM are not available

Description:

The SWS specifies that EcuM requires interfaces from the CanSM module in order to fulfil its core functionality.

Rationale:

Currently EcuM does not make use of the `CanSM_StartWakeupSource(Std_ReturnType, NetworkHandleType)` and `CanSM_StopWakeupSource(Std_ReturnType, NetworkHandleType)` APIs for wakeup handling.



#### Requirements:

SWS\_EcuM\_02858

- ▶ Additional mandatory expected interfaces

#### Description:

The SWS does not specify anymore some mandatory interfaces for the EcuM which are needed to fulfil its core functionality: 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. DisableAllInterrupts - Depending on the selected value from the Rte configuration parameter InterruptBlockingFunction, when the EcuM exits an exclusive area using the SchM, it needs the DisableAllInterrupts interface. EnableAllInterrupts - Depending on the selected value from the Rte configuration parameter InterruptBlockingFunction, when the EcuM enters an exclusive area using the SchM, it needs the EnableAllInterrupts interface.

#### Rationale:

Currently the EcuM is using the mentioned interfaces but they have been removed from the Autosar 4.-3 SWS document as being mandatory interfaces.

#### Requirements:

SWS\_EcuM\_02858

- ▶ No support for EcuMOSResource

#### Description:

The functionality related to this parameter is not supported by the current implementation. Considering that it is disabled, its multiplicity has been changed from 1..\* to 0.

#### Requirements:

ECUM183\_Conf

- ▶ Parameter EcuMFlexModuleConfigurationRef refers also to other BSW modules init structure

#### Description:

Contrary to the requirement ECUM182\_Conf that is related to the parameter EcuMFlexModuleConfigurationRef, EcuM refers also to the init structure of other BSW modules.

#### Rationale:



Reference to the init structure of BswM, CanTrcv, Cdd, Eth, EthTrcv, FrMultipleConfiguration and Ocu is also needed in parameter `EcuMFlexModuleConfigurationRef`. This is addressed in AUTOSAR 4.-1.3 Software Specification of EcuM.

Reference to the init structure of Dio is also needed in parameter `EcuMFlexModuleConfigurationRef` since Dio will be initialized by EcuM during the startup phase. This is addressed in AUTOSAR Bugzilla issue [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=51175](http://www.autosar.org/bugzilla/show_bug.cgi?id=51175).

Reference to the init structure of Fr is also needed in parameter `EcuMFlexModuleConfigurationRef` since Fr could also be initialized by EcuM during the startup phase. This is addressed in AUTOSAR Bugzilla issue [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=53326](http://www.autosar.org/bugzilla/show_bug.cgi?id=53326).

Reference to the init structure of the following MCAL modules: GtmCfgConfigSet , McIConfigSet , Spm-cConfigSet, which are not specified in the EcuM Software Specification document is also needed in the `EcuMFlexModuleConfigurationRef` parameter.

Requirements:

`ECUM182_Conf`

- ▶ Multi-Core GoSleep Sequence is similar to the Single-Core GoSleep Sequence

Description:

Activities in the GoSleep Sequence have been updated in Autosar 4.3.0 SWS of EcuM after bugzilla RfC #63167 has been implemented. However these changes were only applied to the Single-Core diagram (Activities in the GoSleep Sequence - Figure 12) while the Multi-Core diagrams of the GoSleep Sequence have not been updated. They should of been updated aswell since this RfC solves a bug which applies to both Single-Core and Multi-Core use-cases. This implementation provides the solution from RfC #63167 not only for the Single-Core scenario but also for the Multi-Core scenario.

Requirements:

`SWS_EcuM_04023`, `SWS_EcuM_04027`

- ▶ Multi-Core Poll Sequence is similar to the Single-Core Poll Sequence

Description:

Activities in the Poll Sequence have been updated in Autosar 4.3.0 SWS of EcuM after bugzilla RfC #63167 has been implemented. However these changes were only applied to the Single-Core diagram (Activities in the Poll Sequence - Figure 14) while the Multi-Core diagrams of the Poll Sequence have not been updated. They should of been updated aswell since this RfC solves a bug which applies to both Single-Core and Multi-Core use-cases. This bugzilla RfC fixes the blocking loop of the Poll Sequence which should react also to validated wakeup events, not only to pending wakeup events. This implementation provides the solution from RfC #63167 not only for the Single-Core scenario but also for the Multi-Core scenario.



Requirements:

SWS\_EcuM\_04025, SWS\_EcuM\_04029, EcuM3020

- ▶ EcuM\_MemMap.h is included instead of MemMap.h

Description:

Due to the architecture of EcuM\_MemMap.h, the preprocessing time is significantly decreased compared to MemMap.h. See ASCPROJECT-2931 for a detailed performance comparison. Also see SWS\_MemMap\_00032 which specifies using [Mip]\_MemMap.h.

Requirements:

EcuM2862

- ▶ No support for EcuMWakeupSourcePolling

Description:

The functionality related to this parameter is not supported by the current implementation.

Requirements:

ECUM153\_Conf

- ▶ EcuMMainFunctionPeriod cannot be set to 0 or infinity

Description:

The AUTOSAR specifications (ranging from 4.0.3 to 4.3.1 to R20-11) state that this parameter's range is 0 to infinity. EcuMMainFunctionPeriod cannot be 0 or else EcuM\_MainFunction would never be executed. EcuMMainFunctionPeriod cannot realistically implement infinity as the upper range limit as it doesn't make sense. Therefore, the parameter's range should be limited to a practical range like 0.001 to 255.

Requirements:

ECUM121\_Conf

### 3.3.4.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- ▶ **Shutdown synchronization**

Description:



The slave-core EcuM instance should de-initialize its BswM and SchM before sending a signal to indicate that core is ready for shutdown. However only the BswM is de-initialized before the signal is sent.

Rationale:

Synchronization data must be sent before the SchM is de-initialized because the EcuM uses the SchM Sender-Receiver interface for core synchronization. If SchM is de-initialized first, then this interface cannot be used anymore.

#### ► API Reentrancy status in a Multi-Core context

Description:

The following EcuM APIs are specified by the EcuM SWS as Non-Reentrant: - EcuM\_SetWakeupEvent() - EcuM\_GetPendingWakeupEvents() - EcuM\_GetValidatedWakeupEvents() - EcuM\_GetExpiredWakeupEvents() - EcuM\_ClearWakeupEvent() - EcuM\_StartupTwo() - EcuM\_MainFunction().

However, in a Multi-Core context, these APIs are reentrant, since they can be called from different core contexts.

### 3.3.4.6. Open-source software

EcuM does not use open-source software.

## 3.3.5. Nm module release notes

- AUTOSAR R4.0 Rev 3
- AUTOSAR SWS document version: 3.0.0
- Module version: 5.12.9.B466224
- Supplier: Elektrobit Automotive GmbH

### 3.3.5.1. Change log

This chapter lists the changes between different versions.

#### Module version 5.12.9

2021-10-27



- ▶ Changed dependency between ComControlEnabled and PassiveModeEnabled from error message to a warning

#### **Module version 5.12.8**

2021-06-25

- ▶ Added NM\_STATE\_OFFLINE state type

#### **Module version 5.12.7**

2021-03-05

- ▶ Internal module improvements. This module version update does not affect module functionality

#### **Module version 5.12.6**

2020-10-23

- ▶ Improved Active wakeup Bit functionality

#### **Module version 5.12.5**

2020-06-19

- ▶ Improved calculate value functionality in tresos for parameter NmChannelId
- ▶ ASCNM-617 Fixed known issue: Nm\_RemoteSleepIndication and Nm\_RemoteSleepCancellation cannot be used when multicore is enabled, Nm coordinator is not used and NmProvideRemoteSleepCallbacks is set
- ▶ ASCNM-623 Fixed known issue: Linker error is reported for Nm if multi-core is enabled while development error detection and coordinator support are disabled

#### **Module version 5.12.4**

2020-02-21

- ▶ Changed default value for parameter NmComControlEnabled to false

#### **Module version 5.12.3**

2019-10-11



- ▶ ASCNM-597 Fixed known issue: Linker errors are reported due to incorrect memory mapping

#### **Module version 5.12.2**

2019-06-14

- ▶ Internal module improvements. This module version update does not affect module functionality.

#### **Module version 5.12.1**

2019-02-15

- ▶ Internal module improvements. This module version update does not affect module functionality.

#### **Module version 5.12.0**

2018-10-26

- ▶ Implemented Multi-core support
- ▶ ASCNM-566 Fixed known issue: In certain configurations, Nm does not support communication control service for UdpNm

#### **Module version 5.11.4**

2018-06-22

- ▶ Added extra callback support for Nm\_RemoteSleepIndication and Nm\_RemoteSleepCancellation functions.

#### **Module version 5.11.3**

2018-02-16

- ▶ ASCNM-502 Fixed known issue: The API Nm\_SetUserData() may cause a null pointer exception
- ▶ Updated description for parameters NmCarWakeUpRxEnabled and NmCarWakeUpCallback
- ▶ Removed AUTOSAR 3.x compliant symbolic name value macros and updated the logic to only provide AUTOSAR 4.0.2 compliant macros

#### **Module version 5.11.2**

2017-09-22



- ▶ Added NmNodeDetectionEnabled, NmNodeIdEnabled and NmRepeatMsgIndEnabled as per channel configurable.

#### **Module version 5.11.1**

2017-06-30

- ▶ Implemented support for car wake up
- ▶ ASCNM-473 Fixed known issue: Coordinated shutdown is not aborted while waiting for valid synchronization
- ▶ ASCNM-478 Fixed known issue: Nm gets locked in synchronization state. Fix is available only by enabling and configuring NmWaitForShutdownTime.

#### **Module version 5.11.0**

2017-03-31

- ▶ Updated the path of the REFINED\_MODULE\_DEF parameter

#### **Module version 5.10.0**

2016-05-24

- ▶ Internal module improvements. This module version update does not affect module functionality: ASC-NM-431, ASCNM-432

#### **Module version 5.9.0**

2016-02-10

- ▶ Added support for Debug & Trace with custom header file configurable via parameter BaseDbgHeader-File

#### **Module version 5.8.0**

2015-11-06

- ▶ Internal module improvement. This module version update does not affect module functionality: ASC-NM-409, ASCNM-411, ASCNM-412, ASCNM-416, ASCNM-417, ASCNM-417, ASCNM-418, ASCNM-419

#### **Module version 5.7.0**

2015-02-20



- ▶ Add global NmPassiveModeEnabled parameter

#### **Module version 5.6.0**

2014-10-02

- ▶ Internal module improvement. This module version update does not affect module functionality: ASC-NM-402

#### **Module version 5.5.0**

2014-04-25

- ▶ Internal module improvement. This module version update does not affect module functionality: ASC-NM-397, ASCNM-395, ASCNM-393, ASCNM-389, ASCNM-388

#### **Module version 5.4.1**

2013-10-18

- ▶ ASCNM-355 Fixed known issue: `Nm_MainFunction` is not available if *Coordinator Support* is disabled
- ▶ Updated the memory mapping macro names in template file `Nm_ExtraCallbacks.c`
- ▶ Added support for function tracing via AUTOSAR Debugging

#### **Module version 5.4.0**

2013-06-26

- ▶ ASCNM-323 Fixed known issue: Compiler error when `NmComUserDataSupport` is set to true
- ▶ ASCNM-324 Fixed known issue: Memory access violation when channels are not configured to be zero-based and consecutive
- ▶ ASCNM-321 Fixed known issue: *Nm Coordinator* does not shutdown the channel if parameter `NmShutdownDelayTimer` is set to zero
- ▶ ASCNM-291 Fixed known issue: Compiler errors may occur if parameter `NmStandardBusType` has the same value in all channels
- ▶ ASCNM-320 Fixed known issue: `Nm_StateChangeNotification` cannot be modified as it is not provided as a template function
- ▶ ASCNM-354 Fixed known issue: Compiler errors may occur if parameter `NmStateReportEnabled` is set to false for all channels and parameter `NmStateChangeIndEnabled` is set to true



### Module version 5.3.0

2013-02-15

- ▶ ASCNM-300 Fixed known issue: A compiler error occurs when symbolic names according to AUTOSAR R4.0.3 are used
- ▶ Changed the reference path of ComMChannel in parameter NmComMChannelRef to /AUTOSAR/EcucDefs/ComM/ComMConfigSet/ComMChannel
- ▶ Implemented memory allocation keywords in compliance to AUTOSAR 4.0.3
- ▶ ASCNM-276 Fixed known issue: The Nm may access an incorrect or non-existent channel when channels are not configured to be zero-based and consecutive

### Module version 5.2.0

2012-10-12

- ▶ Added support for *State Change Notification*
- ▶ ASCNM-261 Fixed known issue: A compiler error occurs when Nm\_Init() is called without any arguments and NmCoordinatorSupportEnabled and NmDevErrorDetect are both disabled
- ▶ Changed the top-level structure of the software-component description in the arxml files from /AUTOSAR/Nm to /AUTOSAR\_Nm
- ▶ Removed the type Nm\_ReturnType and the associated return values from NmStack\_Types.h

### Module version 5.1.1

2012-06-27

- ▶ ASCNM-235 Fixed known issue: NM has to support \_NON\_ zero-based and \_NON\_ consecutive Nm channel IDs
- ▶ Added templates for the extra callback notifications for OEM-specific extensions
- ▶ Added enumeration type Nm\_BusNmType in header file NmStack\_types.h

### Module version 5.1.0

2012-03-16

- ▶ ASCNM-199 Fixed known issue: Nm does not support FrNm when communication control service Nm-ComControlEnabled is activated
- ▶ ASCNM-217 Fixed known issue: Signature of Nm\_GetUserData() is wrong
- ▶ ASCNM-228 Fixed known issue: Nm throws compilation error when bus type is not selected as CAN or FlexRay



- ▶ Updated to AUTOSAR 4.0.3 version

#### Module version 5.0.0

2011-08-22

- ▶ Initial AUTOSAR 4.0 version

#### 3.3.5.2. New features

- ▶ No new features have been added since the last release.

#### 3.3.5.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- ▶ Additional callout in callback function `Nm_StateChangeNotification()`

Description:

The callback function `Nm_StateChangeNotification()` forwards the state changes to the additional callout function `Nm_StateChangeNotificationCallout()`. Parameter `NmStateChangeNotificationCallout` controls this behavior.

This additional callout function can be implemented in order to handle the notified state changes from the BusNm modules.

- ▶ Function tracing support via AUTOSAR Debugging

Description:

The module Nm supports tracing of function entry and exit via the EB Dbg module.

Function tracing records following parameters for each function:

- ▶ function name
- ▶ values of the function arguments
- ▶ point in time of function invocation
- ▶ point in time of function termination
- ▶ return value of the function
- ▶ ComUserDataSupport configuration in modules Nm, CanNm, FrNm, UdpNm

Description:



The value of `NmComUserDataSupport`, should be equal to `FrNmComUserDataSupport`, `CanNmCo-  
mUserDataSupport`, `UdpNmComUserDataSupport` in case the modules are added into configuration  
(`FrNm`, `CanNm`, `UdpNm`). A XPath check is added for the node `NmComUserDataSupport` in `Nm.xdm.m4`.

### 3.3.5.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

- ▶ No support for LINNM (reference to product description: ASCPD-105)

Description:

The *NM Coordinator* functionality does not support LINNM as a standard bus type.

Requirements:

`Nm220_Conf`

- ▶ Coordination of nested subbusses (reference to product description: ASCPD-106)

The *NmCoordinatorSyncSupport* functionality is not supported in current implementation.

Description:

The requirement `Nm256` demands that a *NM Ccoordinator* shall support two or more *NM Coordinators* connected to the same NM cluster.

The requirement `Nm257` describes that one or more passive *NM Coordinators* send Nm messages only if the node has a network management request pending or a connected network. In this case the connected network is not ready to sleep and is coordinated actively by that coordinator.

The requirements under section 7.2.8 *Coordination of nested subbusses* in the AUTOSAR specification (Rev 4.0.3) describe the detailed functionality required to coordinate the nested subbusses that use an active or passive *Nm Coordinator*.

In contrast to these requirements, coordination of nested subbusses that use active or passive *NM Coordinators* is not supported.

Requirements:

`Nm256`, `Nm257`, `Nm258`, `Nm259`, `Nm260`, `Nm261`, `Nm262`, `Nm263`, `Nm264`, `Nm265`, `Nm266`, `Nm267`,  
`Nm237_Conf` and `Nm236_Conf`

- ▶ Tracing of variables is not supported via AUTOSAR Debugging

Description:



The Nm module does not provide support for tracing global variables.

Requirements:

Nm189, Nm190, Nm191, Nm240

- ▶ Scope of evaluation of coordination algorithm

Description:

The requirement Nm171 describes that the evaluation of coordination algorithm shall be carried out in all API calls of the Nm including the main function of Nm. In contrast to this requirement, the evaluation and actions of coordination algorithm are not performed in all API calls of the Nm and only performed in `Nm_MainFunction()` and in some API calls (e.g. `Nm_NetworkRelease()` and `Nm_RemoteSleepIndication`).

Rationale:

The requirement Nm171 is implementation-specific. See the Bugzilla entry [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=58474](http://www.autosar.org/bugzilla/show_bug.cgi?id=58474), where the description of a similar requirement Nm170 is corrected as follows:  
[SWS\_NM\_00170]: There are no limitation where the actions performed by the coordinator algorithm shall take place. This can be done either by the Nm main function (`Nm_MainFunction()`) or module indication/callbacks. So it is clear that it is up to the implementer to perform the actions of coordination algorithm on some APIs. It may not necessary to perform the actions of coordination algorithm in all the APIs of Nm. Additionally note that Nm170 and Nm171 are close together.

Requirements:

Nm171

- ▶ Additional provision of callback API `Nm_StateChangeNotificationCallout()`

Description:

The requirement Nm095 demands that the Nm module shall not provide interface functions beyond those specified in the AUTOSAR specification. In contrast to this requirement, an additional (user-specific) call-out function `Nm_StateChangeNotificationCallout()` is provided to notify state changes within the <BusNm> modules.

Rationale:

It is at least one customer requirement which is not specified in the AUTOSAR specification.

Requirements:

Nm095



- ▶ No call of `<BusNm>_NetworkRequest()` for the network which throws error on `<BusNm>_NetworkRelease()`

Description:

The requirement Nm236 states that if Nm receives `E_NOT_OK` on invocation of `<BusNm>_NetworkRelease()`, Nm shall abort the shutdown and call `<BusNm>_NetworkRequest()` for all the networks in the coordinated clusters. This includes the own network which throws the error on `<BusNm>_NetworkRelease()`. In contrast to this requirement, `<BusNm>_NetworkRequest()` is not called for the own network which throws `E_NOT_OK` on invocation of `<BusNm>_NetworkRelease()`.

Additionally `E_NOT_OK` is the correct return value as Nm236 (in AUTOSAR 4.0 Rev 3) talks about `NM_E_NOT_OK`. See [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=56744](http://www.autosar.org/bugzilla/show_bug.cgi?id=56744)

Rationale:

The use case where `<BusNm>_NetworkRelease()` returns `E_NOT_OK` is invalid. See the RFC [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=60589](http://www.autosar.org/bugzilla/show_bug.cgi?id=60589).

Requirements:

Nm236

### 3.3.5.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- ▶ Implementation-specific parameter range limitations

Description:

Parameter `NmChannelId`: Range limited from 0..255 to 0...254.

Rationale:

This limitation allows for a more efficient implementation.

- ▶ No support for link time configuration

Description:

The Nm Interface module can only be configured at pre-compile time. Link time configuration is not supported.

Rationale:



Source code can be optimized with respect to code size and execution speed more aggressively if only pre-compile time configuration is supported.

Requirements:

Nm195, Nm247

### **3.3.5.6. Open-source software**

Nm does not use open-source software.



## 4. ACG8 Mode Management user guide

### 4.1. Overview

This user guide describes the concepts and the configuration of the modules:

- ▶ EcuM
- ▶ BswM
- ▶ ComM
- ▶ Nm

This user guide is intended for readers who have good knowledge of AUTOSAR and about the purpose of the ACG8 Mode Management modules.

### 4.2. Background information

In this user guide you learn about the mode management modules EcuM and BswM. You also find information regarding specific features of ComM and Nm.

For general information about ComM, Nm and related <bus>Nm modules, see section *Network management and state management stack* in chapter *Concepts* in the EB tresos AutoCore Generic 8 documentation.

The mode management modules EcuM and BswM provide options that support the **BSW distribution** feature for use in a multi-core project environment. For more information about the general concepts related to multi-core and BSW distribution in AUTOSAR, and the support provided in EB tresos AutoCore Generic, see section *Multi-core support* in chapter *Concepts* in the EB tresos AutoCore Generic 8 documentation.

For additional background information, see the module-specific user guides:

- ▶ [Section 4.4, “BswM module user guide”](#) for the BswM module
- ▶ [Section 4.6, “EcuM module user guide”](#) for the EcuM module
- ▶ [Section 4.7, “Nm module user guide”](#) for the Nm module



## 4.3. Configuring the EcuM and the BswM modules

### 4.3.1. Overview

The following sections provide you with instructions on how to configure the EcuM and BswM modules.

**NOTE**



**Before configuring the EcuM module, execute the Service Needs Calculator**

Many parameters of the EcuM module are set automatically by the **Service Needs Calculator**. Therefore, execute the **Service Needs Calculator** before configuring the EcuM module.

Instructions on how to execute the **Service Needs Calculator** are available in the EB tresos AutoCore Generic 8 Base documentation.

In this chapter you find only instructions on how to set those parameters that are not set by the **Service Needs Calculator**. For the best possible results, read through the instructions below step-by-step in the order presented.

This guide is intended for users who have basic knowledge of the following:

- ▶ Integration of AUTOSAR BSW modules
- ▶ Startup behavior of an ECU
- ▶ State management of an ECU

### 4.3.2. General configuration/integration hints

Two template files for all EcuM callout functions are generated in the folder `workspace/<yourProject-Name>/output/generated/templates` when you generate your project. To include these files into the build process, copy the template files after generation into your application folder. If necessary, adapt the template files to your specific needs. Further information about generating and building your application can be found in the build environment chapter of the EB tresos AutoCore Generic 8 Base documentation.

As illustrated in [Figure 4.12, “Sample containers, references and multiplicity of a BswM rule”](#), each `BswMRule` references exactly one `BswMLogicalExpression` and one `BswMActionList`. This is so, because only the `BswMRuleTrueActionList` is used. Therefore, these two references are not listed each time in the configuration examples.

You must configure the interfaces that are provided by the BswM module. At least the following three interfaces which are configured by the parameter `BswMModeRequestPort` have to be configured.



BswMModeRequestPort	BswMModeRequestSource
BswM_ReqPort	BswMGenericRequest
EcuM_WakeupSourcePort_Power	BswMEcuMWakeupSource
BswM_CurrentModePort	BswMComMIndication

Table 4.1. BswM interface configuration (BswMModeRequestPort)

**WARNING****Map BswM\_MainFunction() to a standard task (not extended)**

If Rte\_Start() is called from BswM context, map the BswM\_MainFunction() to a standard task (not extended). Otherwise, a deadlock occurs.

#### 4.3.2.1. Migrating the EcuM from ACG-8.5.2 to ACG-8.6.0

##### 1. Migrating configuration parameters

- ▶ Import the EcuM module.

For details on how to import configuration data, see the EB tresos Studio user's guide.

- ▶ For each EcuMDriverInitItem container in EcuMDriverInitListOne and EcuMDriverRestartList, you need to configure a Boolean parameter EcuMEnableDriver.

If enabled, the corresponding EcuMDriverInitItem service is initialized by the EcuM. This parameter is enabled by default for each new EcuMDriverInitItem container.

##### 2. Migrating API changes

- ▶ The access to the EcuM post-build data structure changed.

Therefore, the EcuM\_PBcfg.h header file must be included in order for the callout function EcuM\_DeterminePbConfiguration() to return a pointer to the EcuM post-build data structure.

- ▶ The definitions of the following callout functions changed: EcuM\_AL\_DriverInitOne() and EcuM\_AL\_DriverRestart().

EcuM provides an example of the definitions in the generated template file EcuM\_Callout\_Stubs.c.

#### 4.3.2.2. Integration notes for multi-core projects

Multi-core is enabled in the Os by configuring the OsNumberOfCores parameter to a value higher than 1 and by assigning Os Applications to different core IDs. The initial core from which the application starts will always be the core-ID 0 or the OS\_CORE\_ID\_MASTER (as specified by the Os SWS).

The master EcuM instance needs to be connected with each one of the slave EcuM instances. In order to achieve this, connect the EcuM slave ports with the master EcuM instance by adding new configuration para-



meters to the `Rte` configuration under the **BSW Module Instances** list, related to the master `EcuM` instance. The `EcuM` ports need to be connected in the following manner:

1. Add a new container for each slave `EcuM` instance in the **RteBswRequiredSenderReceiverConnection** tab with the following connections:

- ▶ `RteBswProvidedVariableDataPrototypeRef`: add a reference to the `EcuM_SlaveCoreReadyPort`. Each slave core shall write to this variable data prototype to signal that they are ready. The slave core shall pass their `coreId`.
- ▶ `RteBswRequiredVariableDataPrototypeRef`: add a reference to the `EcuM_SlaveCoreSyncPort`. The master core shall read from this variable to determine if the slave cores are ready.
- ▶ `RteBswProvidedDataModInstRef`: add a reference to the slave `EcuM` instance that you need to synchronize with the master `EcuM` instance.

Each slave `EcuM` instance needs to add connection with the master `EcuM` instance, by connecting the `EcuM` master ports with each `EcuM` slave instances. In the `Rte` configuration, under the **BSW Module instances** list, in the container of each slave `EcuM` instance, the `EcuM` ports need to be connected in the following manner:

1. Add a new container in the **RteBswRequiredSenderReceiverConnection** tab with the following connections:
  - ▶ `RteBswProvidedVariableDataPrototypeRef`: add a reference to the `EcuM_MasterCoreReadyPort`. The master core shall write to this variable data prototype to signal the slave cores that it is ready.
  - ▶ `RteBswRequiredVariableDataPrototypeRef`: add a reference to the `EcuM_MasterCoreSyncPort`. The slave cores shall read from their port to determine if the master core is ready.
  - ▶ `RteBswProvidedDataModInstRef`: add a reference to the master `EcuM` instance in order to synchronize it with the current slave `EcuM` instance.
2. Also, the ports related to the sleep mode need to be connected, for this add a new container in the **RteBswRequiredSenderReceiverConnection** tab with the following connections:
  - ▶ `RteBswProvidedVariableDataPrototypeRef`: add a reference to the `EcuM_SetSleepMode`. The master core shall write to the `SetSleepMode` port to inform the slave cores of the currently selected sleep mode.
  - ▶ `RteBswRequiredVariableDataPrototypeRef`: add a reference to the `EcuM_GetSleepMode`. Each slave core shall read from their `GetSleepMode` port to determine the current sleep mode.
  - ▶ `RteBswProvidedDataModInstRef`: add a reference to the master `EcuM` instance in order to synchronize it with the current slave `EcuM` instance.

To set up multiple instances of the `BswM` module proceed as follows:

1. Configure an OS-Application for each partition on which a `BswM` instance is required in the `Os` module configuration.



2. Assign the OS-Applications to the desired cores.
3. In the `BswM` module configuration, add a `BswM` instance configuration for each partition as needed.
4. In each `BswM` instance configuration, enable the `BswMPartitionRef` parameter and select the OS-Application reference that you configured earlier in the `OS` module.
5. Finally, set up the BSW partitioning in the `Rte`.

#### 4.3.2.3. Integration notes for cross-core wakeup event handling between EcuM and ComM

EcuM notifies ComM about each validated wakeup event referencing a ComM channel or ComM PNC. When the multi-core functionality is enabled and EcuM receives such a validated wakeup event, EcuM should inform the ComM module only in the context of the ComM's core instance. Since EcuM has multiple instances, one for each core, a mechanism is needed to facilitate cross-core communication between EcuM and ComM.

Multi-core is enabled in the EcuM by enabling the parameter `EcuMEnableMulticore` and in ComM by enabling the parameter `ComMMultiCoreSupport`. The EcuM-ComM cross-core communication is enabled with the parameter `EcuMCrossCoreComMWakeup`.

The cross-core communication is based on the SchM client/server interface. The EcuM's required client/server entries must be connected with the ComM's provided client/server entries, generated for each EcuM wakeup event referencing a ComM channel or ComM PNC, for each EcuM core instance.

In order to configure this, add new entries to the `Rte` configuration under the BSW module instance list, related to each EcuM instance.

Connect these client/server entries as follows:

1. Add a new container for each EcuM instance in the **RteBswRequiredClientServerConnection** tab with the following connections:
  - ▶ `RteBswProvidedClientServerEntryRef`: add a reference to the `ProvidedCSEntry_ComM_EcuM_WakeUpIndication_{ComMChannelName}_Core_{OsCoreIndex}`. These entries are generated by the ComM module.
  - ▶ `RteBswRequiredClientServerEntryRef`: add a reference to the `RequiredCSEntry_ComM_EcuM_WakeUpIndication_{ComMChannelName}_Core_{OsCoreIndex}`. These entries are generated by the EcuM module.
  - ▶ `RteBswProvidedClientServerEntryModInstRef`: add a reference to the ComM master module instance.
2. Configure exactly the same mapping, corresponding to all EcuM instances, also in the ComM master module instance.



## 4.3.3. Configuring EcuM and BswM for single-core projects

### 4.3.3.1. Configuring the startup states

The startup state of the BswM and the EcuM is initiated after power on or reset. [Figure 4.1, “The startup state of the EcuM and of the BswM”](#) gives an overview of the startup states, including the static EcuM states and the configurable BswM states.

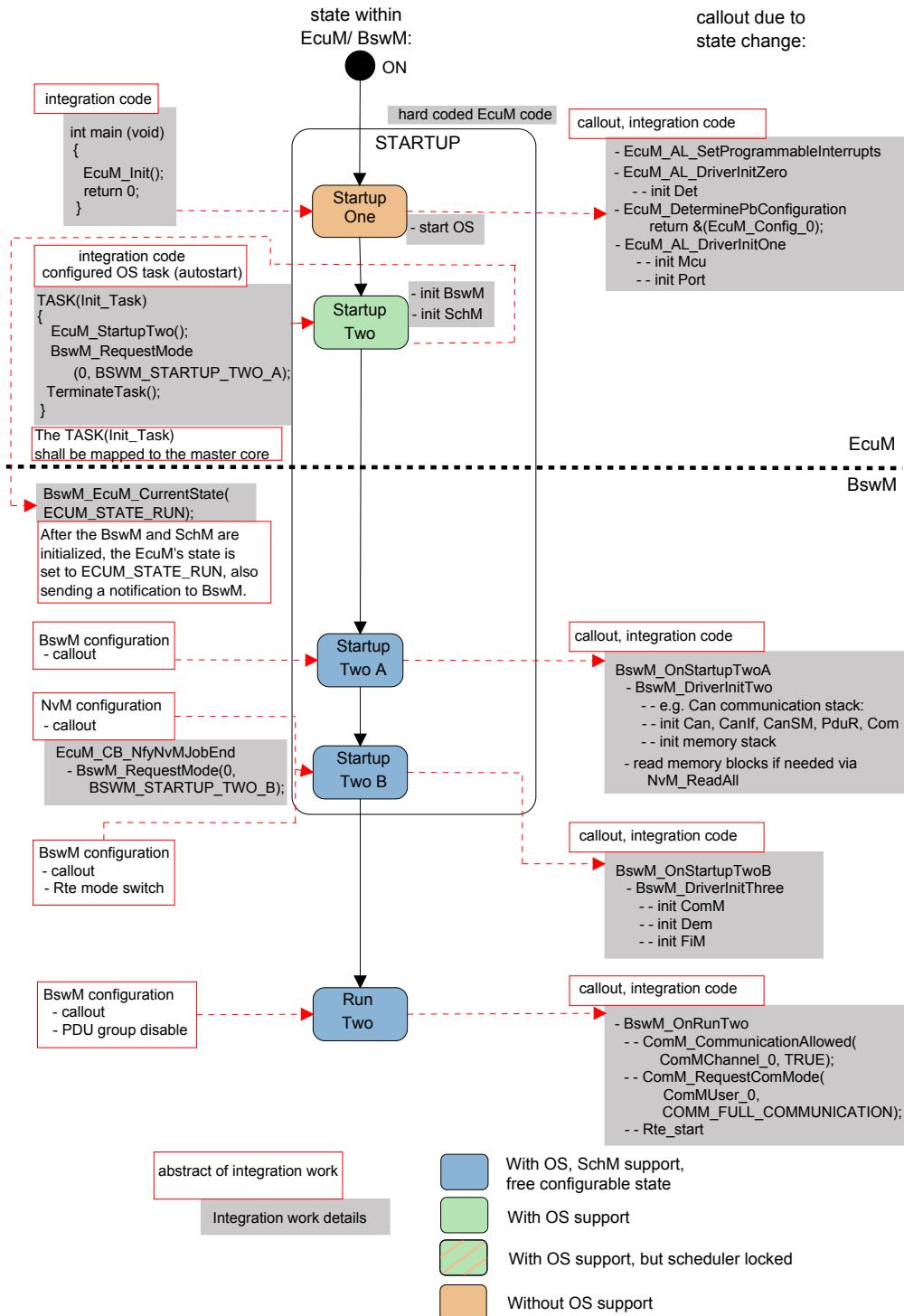
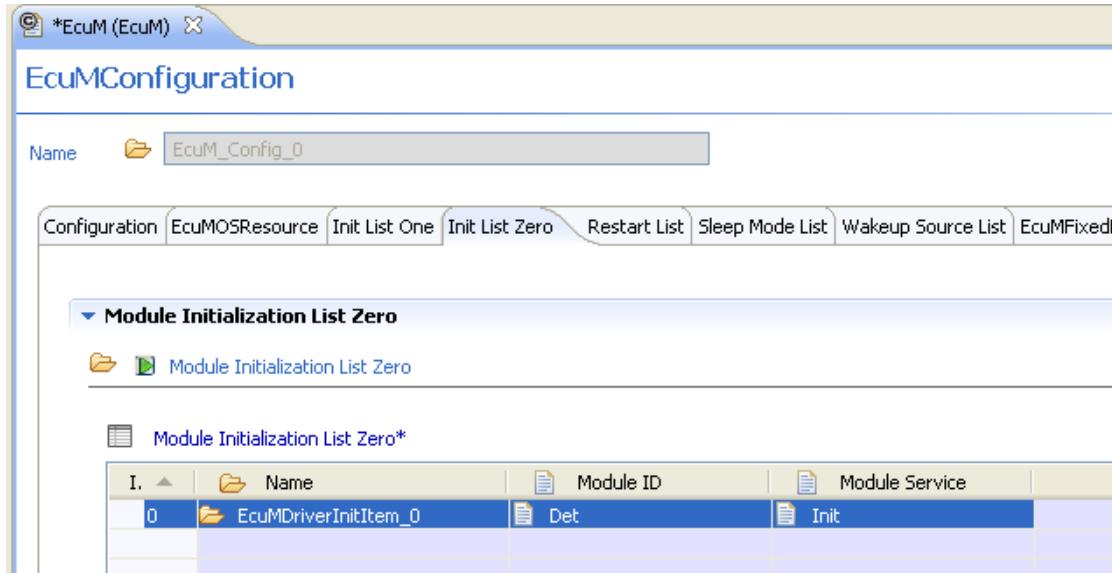


Figure 4.1. The startup state of the EcuM and of the BswM

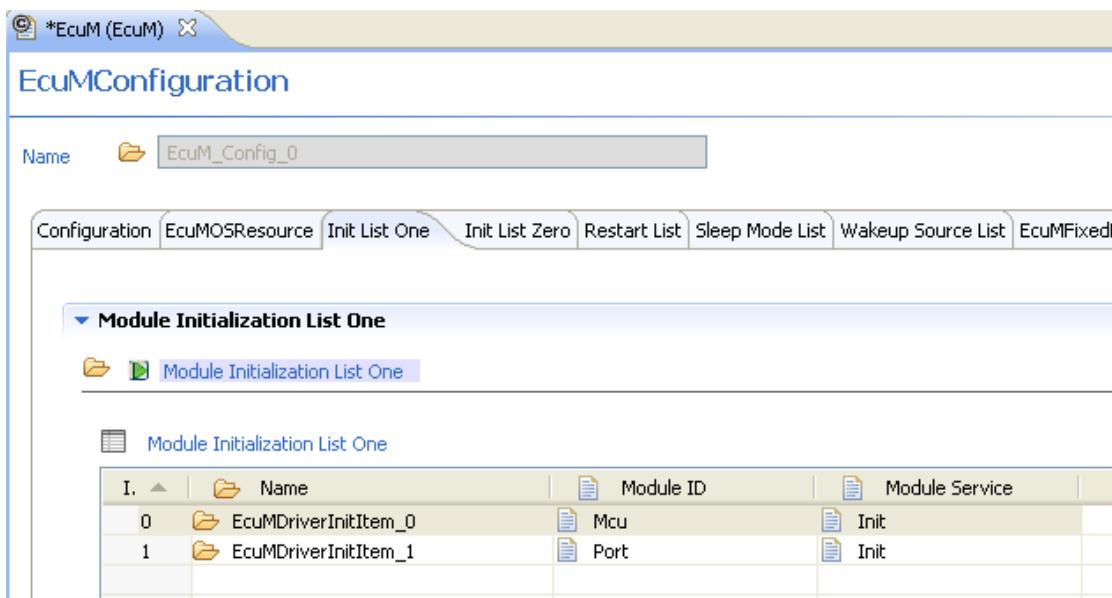
After executing the startup code, the main function is called. Therefore, you must provide the main function and its implementation as integration code as shown in [Figure 4.1, “The startup state of the EcuM and of the BswM”](#).



- ▶ Configure the **Module Initialization List Zero**:



- ▶ Configure the **Module Initialization List One**:



The EcuM enters the state `ECUM_STATE_STARTUP_ONE` and calls the callout functions shown in [Figure 4.1, “The startup state of the EcuM and of the BswM”](#).

- ▶ Use the template files as described in [Section 4.3.2, “General configuration/integration hints”](#) as base for your callout implementation. If you have configured the *Module Initialization List Zero* and the *Module Initialization List One* as described above, the template file includes all configured init function calls.

Now the EcuM starts the Os.

- ▶ Configure the *Init\_Task* within the Os and enable the parameter `OsTaskAutostart` for this task.



The `EcuM` starts the `Os` at the end of the `startup` one state. After the `Os` has been started, all tasks with the parameter `OsTaskAutostart` set to `TRUE` are executed. Therefore the `Init_Task` is executed and the integrator must provide the implementation according to [Figure 4.1, “The startup state of the `EcuM` and of the `BswM`”](#).

The `EcuM` initializes the BSW Scheduler (`SchM`) and the `BswM` modules. The `Init_Task` includes a call to the `BswM` module for requesting a new mode:

```
BswM_RequestMode(0, BSWM_STARTUP_TWO_A);
```

The state handling is now given to the `BswM` module.

The general configuration structure of the `BswM` is illustrated by [Figure 4.12, “Sample containers, references and multiplicity of a `BswM` rule”](#). The general settings of some configuration parameters are summarized in [Section 4.3.2, “General configuration/integration hints”](#).

- ▶ In addition to the general settings, configure the `BswM` in the following way for the `startup` state:

<b>BswMRule</b>	<b>BswMModeCondition</b>	<b>BswMAction</b>
BswMRule_StartupTwoA	BswM_Cond_StartupTwoA <ul style="list-style-type: none"> <li>▶ <code>BswMConditionMode: BswM_ReqPort</code></li> <li>▶ <code>BswMConditionValue: BswMBswMode</code></li> <li>▶ <code>BswMBswModeSourceType: BSWM_GENERIC_REQUEST</code></li> <li>▶ <code>BswMBswRequestedMode: BSWM_STARTUP_TWO_A</code></li> </ul>	BswM_Act_StartupTwoA <ul style="list-style-type: none"> <li>▶ <code>BswMAvailableActions: BswMUserCallout</code></li> <li>▶ <code>BswMUserCalloutFunction: BswM_OnStartupTwoA()</code></li> </ul>
BswMRule_StartupTwoB	BswM_Cond_StartupTwoB <ul style="list-style-type: none"> <li>▶ <code>BswMConditionMode: BswM_ReqPort</code></li> <li>▶ <code>BswMConditionValue: BswMBswMode</code></li> <li>▶ <code>BswMBswModeSourceType: BSWM_GENERIC_REQUEST</code></li> <li>▶ <code>BswMBswRequestedMode: BSWM_STARTUP_TWO_B</code></li> </ul>	BswM_Act_StartupTwoB <ul style="list-style-type: none"> <li>▶ <code>BswMAvailableActions: BswMUserCallout</code></li> <li>▶ <code>BswMUserCalloutFunction: BswM_OnStartupTwoB()</code></li> </ul> BswM_Act_Request_RunTwo <ul style="list-style-type: none"> <li>▶ <code>BswMAvailableActions: BswMUserCallout</code></li> <li>▶ <code>BswMUserCalloutFunction: BswM_RequestMode(0, BSWM_RUN_TWO)</code></li> </ul>
BswMRule_RunTwo	BswM_Cond_RunTwo	BswM_Act_RunTwo



BswMRule	BswMModeCondition	BswMAction
	<ul style="list-style-type: none"> <li>▶ BswMConditionMode: BswM_ReqPort</li> <li>▶ BswMConditionValue: BswMBswMode</li> <li>▶ BswMBswModeSourceType: BSWM_GEN-ERIC_REQUEST</li> <li>▶ BswMBswRequestedMode: BSWM_RUN_TWO</li> </ul>	<ul style="list-style-type: none"> <li>▶ BswMAvailableActions: BswMUserCallout</li> <li>▶ BswMUserCalloutFunction: BswM_OnRunTwo ()</li> </ul> <p>BswM_Act_RunTwo_Pdu</p> <ul style="list-style-type: none"> <li>▶ BswMAvailableActions: BswMPduGroupSwitch</li> <li>▶ BswMEnabledPduGroupRef: your configured PDU groups</li> </ul> <p>BswM_Act_NotifyRteRun</p> <ul style="list-style-type: none"> <li>▶ BswMAvailableActions: BswMRteSwitch</li> <li>▶ BswMRteSwitchInterfaceRef: reference to MODE-SWITCH-INTERFACE</li> <li>▶ BswMSwitchedMode: reference to MODE-DECLARATION-GROUP</li> </ul>

#### 4.3.3.2. Configuring the shutdown states

The shutdown state is initiated after a request from a software component or from the ComM module. [Figure 4.2, “The shutdown states”](#) gives an overview of the shutdown states, including the static EcuM state and the configurable BswM states:

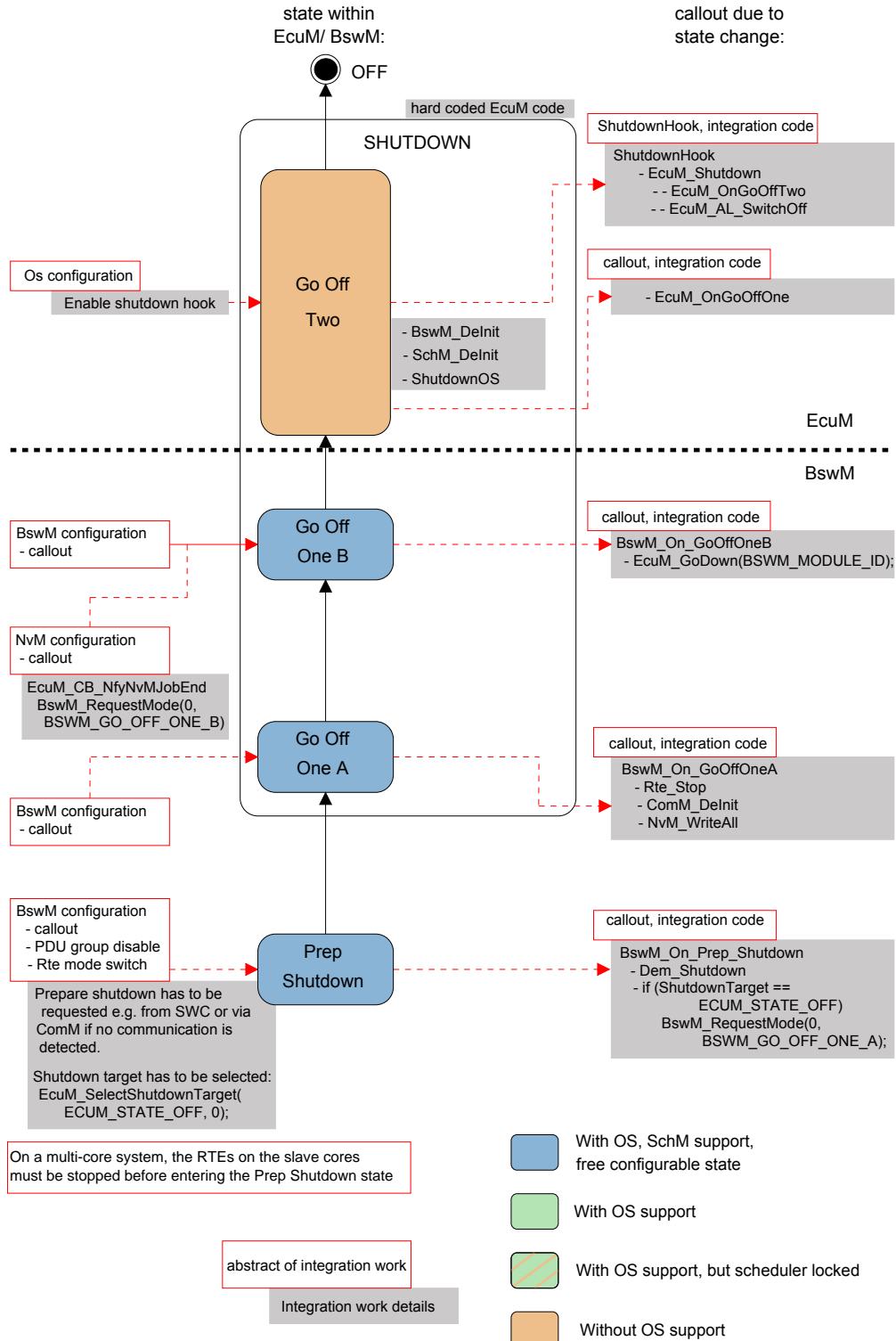


Figure 4.2. The shutdown states



When requesting shutdown, set the shutdown target to `ECUM_STATE_OFF`. The callout function `BswM_On_Prep_Shutdown` evaluates this parameter and requests go off one A from the BswM module. The BswM configuration is summarized in the table below and the needed callout functions are shown in [Figure 4.2, “The shutdown states”](#).

In the callout function `BswM_On_GoOffOneB`, the BswM module delegates the shutdown sequence to the EcuM module by calling the function `EcuM_GoDown`. The EcuM de-initializes the BswM module and the BSW Scheduler (`SchM`), and shuts down the OS.

To ensure that the shutdown hook is called, enable the parameter `OsShutdownHook` in the OS configuration.

Configure the BswM parameters in the following way for the shutdown state:

BswMRule	BswMModeCondition	BswMAction
BswMRule_No-Com	<p><code>BswM_Cond_NoCom</code></p> <ul style="list-style-type: none"> <li>▶ <code>BswMConditionMode: BswM_CurrentModePort</code></li> <li>▶ <code>BswMConditionValue: BswMBswMode</code></li> <li>▶ <code>BswMBswModeSourceType: BSWM_COMM_INDICATION</code></li> <li>▶ <code>BswMBswRequestedMode: COMM_NO_COMMUNICATION</code></li> </ul>	<p><code>BswM_Act_PrepShutdown</code></p> <ul style="list-style-type: none"> <li>▶ <code>BswMAvailableActions: BswMUserCallout</code></li> <li>▶ <code>BswMUserCalloutFunction: BswM_On_Prep_Shutdown()</code></li> </ul> <p><code>BswM_Act_NotifyRteRun</code></p> <ul style="list-style-type: none"> <li>▶ <code>BswMAvailableActions: BswMRteSwitch</code></li> <li>▶ <code>BswMRteSwitchInterfaceRef: reference to MODE-SWITCH-INTERFACE</code></li> <li>▶ <code>BswMSwitchedMode: reference to MODE-DECLARATION-GROUP</code></li> </ul> <p><code>BswM_Act_PrepShutdown_Pdu</code></p> <ul style="list-style-type: none"> <li>▶ <code>BswMAvailableActions: BswMPduGroupSwitch</code></li> <li>▶ <code>BswMDisablePduGroupRef: enter your configured PDU groups</code></li> </ul>
BswMRule_GoOffOneA	<p><code>BswM_Cond_GoOffOneA</code></p> <ul style="list-style-type: none"> <li>▶ <code>BswMConditionMode: BswM_ReqPort</code></li> <li>▶ <code>BswMConditionValue: BswMBswMode</code></li> <li>▶ <code>BswMBswModeSourceType: BSWM_GENERIC_REQUEST</code></li> <li>▶ <code>BswMBswRequestedMode: BSWM_GO_OFF_ONE_A</code></li> </ul>	<p><code>BswM_Act_GoOffOneA</code></p> <ul style="list-style-type: none"> <li>▶ <code>BswMAvailableActions: BswMUserCallout</code></li> <li>▶ <code>BswMUserCalloutFunction: BswM_OnGoOffOneA()</code></li> </ul>



BswMRule	BswMMModeCondition	BswMAction
BswMRue_GoOf-fOneB	<p>BswM_Cond_GoOffOneB</p> <ul style="list-style-type: none"> <li>▶ BswMConditionMode: BswM_ReqPort</li> <li>▶ BswMConditionValue: BswMBswMode</li> <li>▶ BswMBswModeSourceType: BSWM_GEN-ERIC_REQUEST</li> <li>▶ BswMBswRequestedMode: BSWM_GO_OFF_ONE_B</li> </ul>	<p>BswM_Act_GoOffOneB</p> <ul style="list-style-type: none"> <li>▶ BswMAvailableActions: BswMUserCallout</li> <li>▶ BswMUserCalloutFunction: BswM_OnGoOf-fOneB ()</li> </ul>

#### 4.3.3.3. Configuring the sleep states

The `sleep` state is initiated after a request from a software component or from the `ComM` module. [Figure 4.3, “The sleep states”](#) gives an overview of the `sleep` states, including the static `EcuM` states and the configurable `BswM` states.

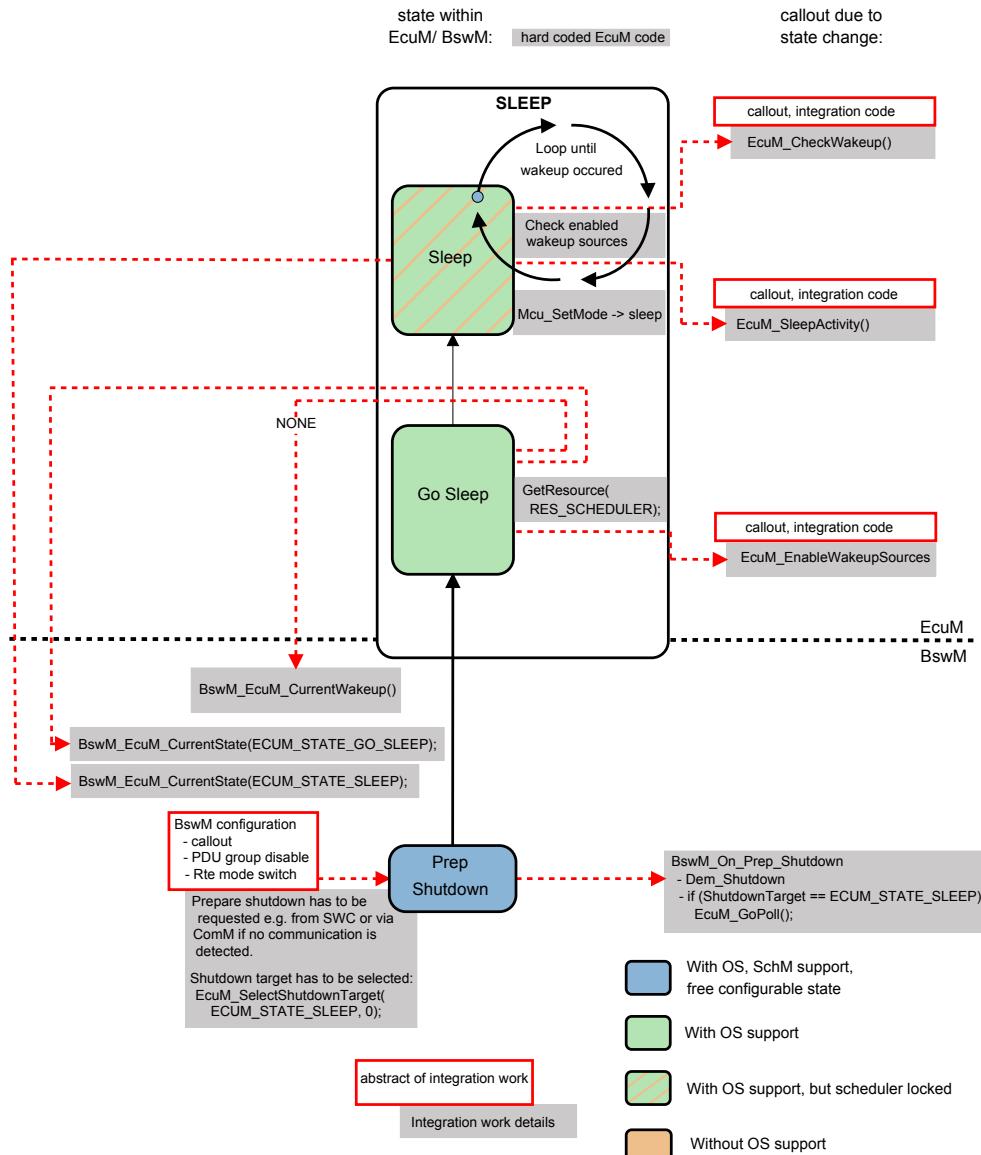


Figure 4.3. The sleep states

To configure the sleep state, you do not need to configure the BswM module, because the prepare shutdown configuration as described in [Section 4.3.3.2, “Configuring the shutdown states”](#) is also used for the sleep state.

- ▶ When requesting sleep, set the shutdown target to ECUM\_STATE\_SLEEP. The callout function BswM\_On\_Prep\_Shutdown evaluates this parameter and calls the EcuM function EcuM\_GoPoll.
- ▶ Alternatively, call the EcuM function EcuM\_GoHalt.
- ▶ To configure the EcuM module for the sleep mode, configure the sleep mode list by registering all wake-up sources for this sleep mode, as shown in [Figure 4.4, “Configuring the Wakeup Source List”](#).

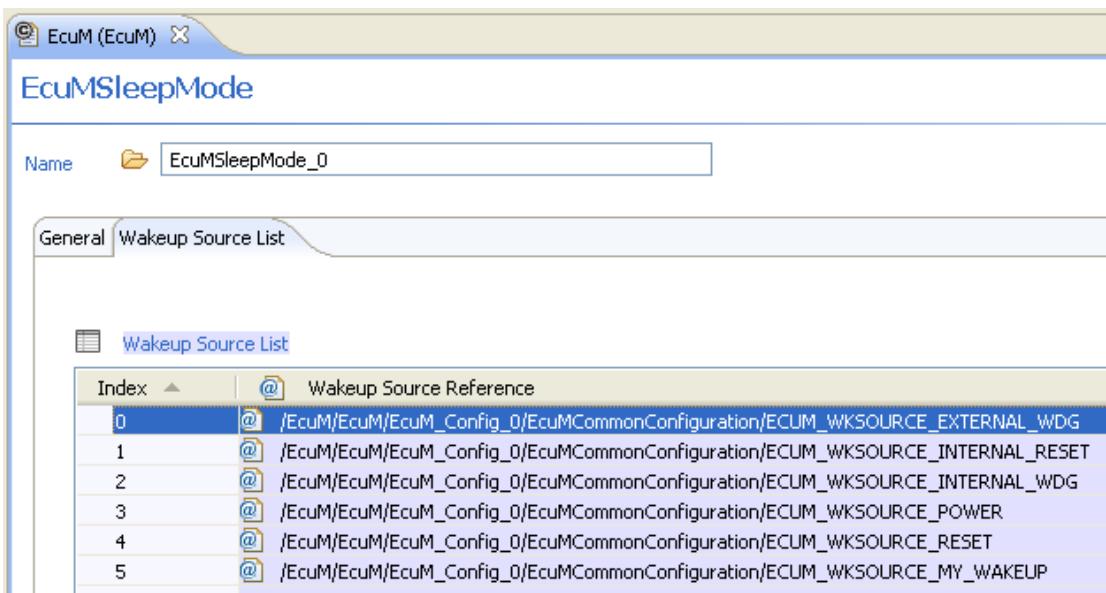


Figure 4.4. Configuring the **Wakeup Source List**

#### 4.3.3.4. Configuring the **wakeup states**

The **wakeup state** is initiated after a pre-configured wakeup event occurred. [Figure 4.5, “The \*\*wakeup states\*\*”](#) gives an overview of the **wakeup states**, including the static **EcuM states** and the configurable **BswM states**.

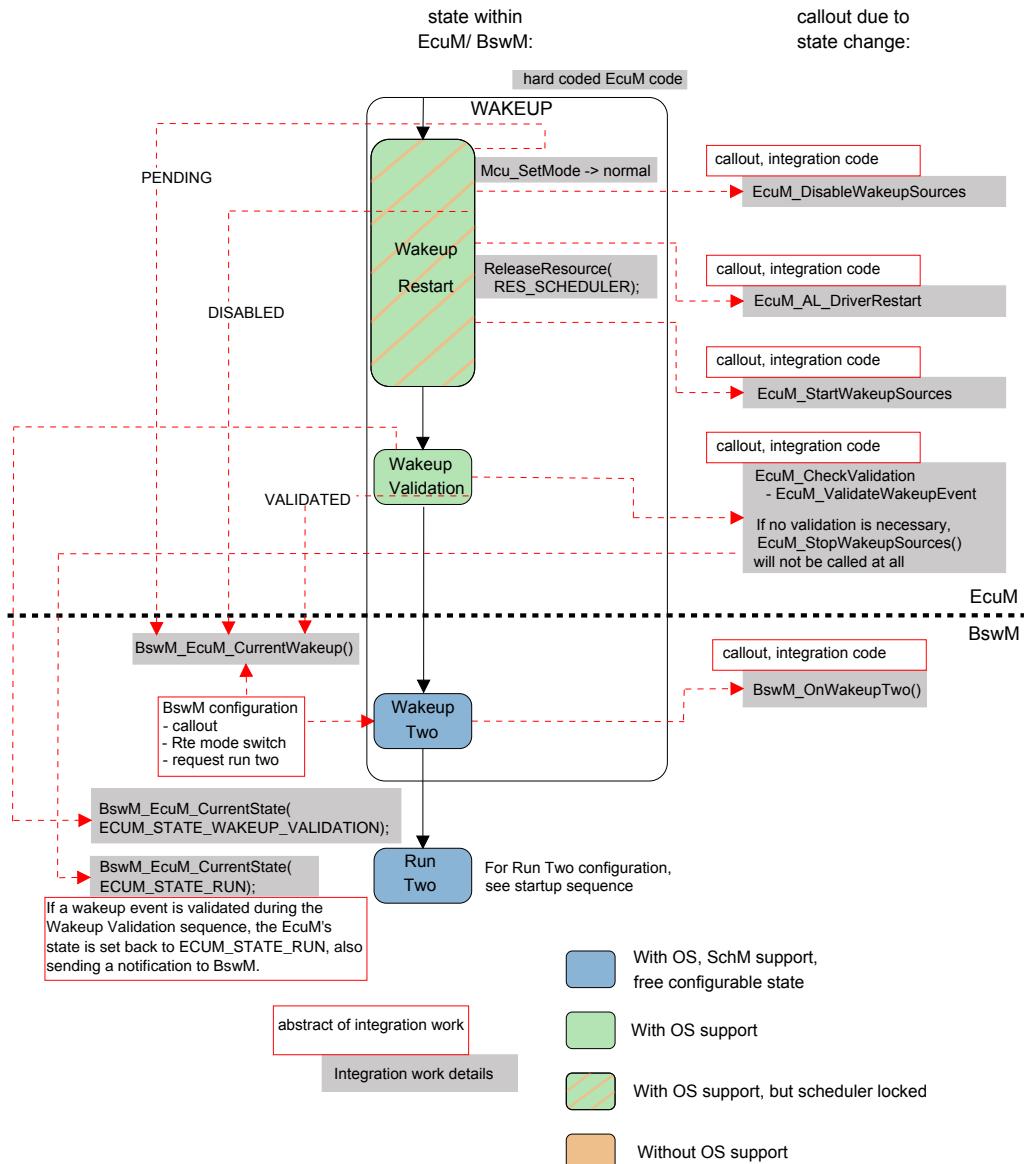


Figure 4.5. The wakeup states

During sleep mode, the EcuM reacts on wakeup events which are configured for the sleep mode. Therefore, the wakeup events have to be configured in the EcuM *Wakeup Source List* as shown in [Figure 4.6, “Configuring the Sleep Mode List”](#). Additionally, the wakeup events must be referenced from the *Sleep Mode List* configuration as shown in [Figure 4.4, “Configuring the Wakeup Source List”](#).



## EcuMConfiguration

The screenshot shows the EcuM Configuration software interface. At the top, there is a navigation bar with tabs: 'Name', 'EcuM\_Config\_0', 'Restart List', 'Sleep Mode List' (which is currently selected), 'Wakeup Source List', 'EcuMFixedModuleConfi', 'EcuMComMCommunication', 'Init List Three', 'Init List Two', and 'EcuMFix'. Below the navigation bar is a table titled 'Wakeup Source List'. The table has columns: 'I.', 'Name', 'Reset Reason', 'WKS ID', and a delete icon. The data in the table is as follows:

I.	Name	Reset Reason	WKS ID
0	ECUM_WKSOURCE_POWER	0x0	0x1
1	ECUM_WKSOURCE_RESET	0x2	0x2
2	ECUM_WKSOURCE_INTERNAL_RESET	0x1	0x4
3	ECUM_WKSOURCE_INTERNAL_WDG	0x1	0x8
4	ECUM_WKSOURCE_EXTERNAL_WDG	0x3	0x10
5	ECUM_WKSOURCE_MY_WAKEUP	0xffffffff	0x20

Figure 4.6. Configuring the **Sleep Mode List**

If a wakeup event occurs, the `EcuM` module signals this to the `BswM` module by calling `BswM_EcuM_CurrentWakeups` with the wakeup status `PENDING`. After setting the `Mcu` to normal processing speed, the wakeup sources are disabled. This disablement is signaled to the `BswM` module by calling `BswM_EcuM_CurrentWakeups` with the wakeup status `DISABLED`. In this configuration example, these two indications are not further processed by the `BswM` module.

After validating a wakeup event successfully, the `BswM_EcuM_CurrentWakeups` is called again, with the wakeup status `VALIDATED`. This indication brings the `BswM` into the state `wakeup_two` and hands over the further processing of the wakeup handling to the `BswM` module.

Configure the `BswM` in the following way for the `wakeup` state:

<b>BswMRule</b>	<b>BswMModeCondition</b>	<b>BswMAction</b>
BswMRule_Wake-upTwo	<p><code>BswM_Cond_WakeupTwo</code></p> <ul style="list-style-type: none"> <li>▶ <code>BswMConditionMode: EcuM_WakeupSource-Port_Power</code></li> <li>▶ <code>BswMConditionValue: BswMBswMode</code></li> <li>▶ <code>BswMBswModeSourceType: BSWM_ECUM_WAKEUP_SOURCE</code></li> <li>▶ <code>BswMBswRequestedMode: ECUM_WKSTATUS_VALIDATED</code></li> </ul>	<p><code>BswM_Act_WakeUpTwo</code></p> <ul style="list-style-type: none"> <li>▶ <code>BswMAvailableActions: BswMUserCallout</code></li> <li>▶ <code>BswMUserCalloutFunction: BswM_OnWakeupsTwo()</code></li> </ul> <p><code>BswM_Act_NotifyRteRun</code></p> <ul style="list-style-type: none"> <li>▶ <code>BswMAvailableActions: BswMRteSwitch</code></li> <li>▶ <code>BswMRteSwitchInterfaceRef: reference to MODE-SWITCH-INTERFACE</code></li> <li>▶ <code>BswMSwitchedMode: reference to MODE-DECLARATION-GROUP</code></li> </ul>



BswMRule	BswMModeCondition	BswMAction
		<p>BswM_Act_RequestRunTwo</p> <ul style="list-style-type: none"> <li>▶ BswMAvailableActions: BswMUserCallout</li> <li>▶ BswMUserCalloutFunction: BswM_RequestMode(0, BSWM_RUN_TWO)</li> </ul>

## 4.3.4. Configuring EcuM and BswM for multi-core projects

### 4.3.4.1. Configuring startup on systems with multiple cores

To configure `startup` on a multi-core system, take the following steps:

On the master core:

1. Configure the `TASK(Init_Task_Master)` on the master to be triggered as a consequence of the `StartOs()` and inside it call `EcuM_StartupTwo()`.
2. Configure in `BswM` a rule and action list that shall trigger `STARTUP_TWO_A`.
3. `STARTUP_TWO_A` shall trigger the `NvM_ReadAll()` operation.
4. When `NvM_ReadAll()` is complete, the NvM Job end notification callout is called e.g. `EcuM_CB_NfyNvMJobEnd()`. Request `STARTUP_TWO_B` from this callout.
5. Configure in `BswM` a rule and action list that shall trigger `STARTUP_TWO_B`.
6. `STARTUP_TWO_B` shall request `BSWM_RUN_TWO`.
7. Configure in `BswM` a rule and action list that shall trigger `BSWM_RUN_TWO`.
8. `BswM_OnRunTwo()` is called.

On the slave core:

1. Configure the `TASK(Init_Task_Slave)` on the slave to be triggered as a consequence of the `StartOs()` and inside it call `EcuM_StartupTwo()`.
2. Configure in `BswM` a rule and action list that shall trigger `STARTUP_TWO_C`.
3. When `NvM_ReadAll()` is complete on the master core, the `BSWM_STARTUP_TWO_D` is requested by the master core on the slave core.
4. Configure in `BswM` a rule and action list that shall trigger `STARTUP_TWO_D`.
5. `STARTUP_TWO_D` shall request `BSWM_RUN_TWO`.
6. Configure in `BswM` a rule and action list that shall trigger `BSWM_RUN_TWO`.



## 7. BswM\_OnRunTwo() is called.

Below you have a diagram depicting an example on how the `startup` is working on an ECU with one master core and one slave core.

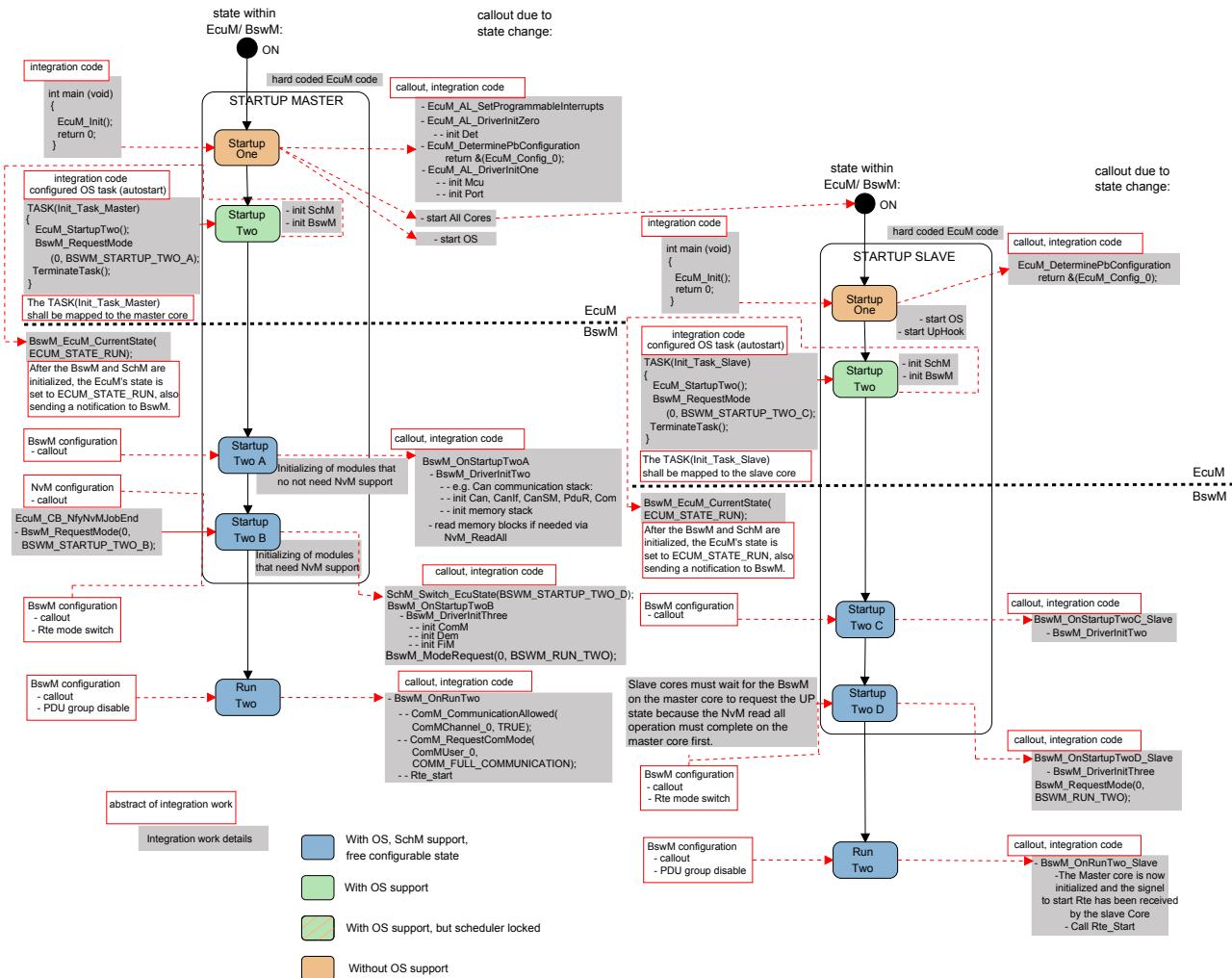


Figure 4.7. The startup state of the EcuM and of the BswM with multi-core

### 4.3.4.2. Configuring shutdown on systems with multiple cores

To shutdown a multi-core system, a synchronization mechanism between the slave core and the master core is needed.

To configure shutdown on a multi-core system, take the following steps:

On the master core:



1. Configure all the action to be done before going to state PrepShutdown.
2. Configure BSWM\_GO\_OFF\_ONE\_A to be called from state PrepShutdown.
3. Configure all the actions to be done in GO\_OFF\_ONE\_A state e.g. Rte\_Stop(), NvM\_WriteAll().
4. When NvM\_WriteAll() is complete, configure the NvM Job End callout e.g. EcuM\_CB\_NfyNvMJobEnd() to be called and BSWM\_GO\_OFF\_ONE\_B is requested from BswM.
5. Configure in BswM a rule and action list that shall trigger GO\_OFF\_ONE\_B.
6. GO\_OFF\_ONE\_B shall pass the notification that the NvM\_WriteAll() is finished to the slave core.
7. The master core shall wait until the slave core signals that it is ready for shutdown.
8. GO\_OFF\_ONE\_B shall call EcuM\_GoDown().

On the slave core:

1. Configure all the action to be done before going to state PrepShutdown.
2. Configure in BswM a rule and action list that shall trigger GO\_OFF\_ONE\_C.
3. Wait for GO\_OFF\_ONE\_D notification from the master core.
4. Configure in BswM a rule and action list that shall trigger GO\_OFF\_ONE\_D.
5. GO\_OFF\_ONE\_D shall call EcuM\_GoDown().

The callout EcuM\_AL\_SwitchOff() must be configured and the actual shutdown (POWER OFF/RESET) must be done via integration code since it is hardware specific. Also depending on project needs the shutdown might require additional operations.

Below you have a diagram depicting an example on how the shutdown is working on an ECU with one master core and one slave core.

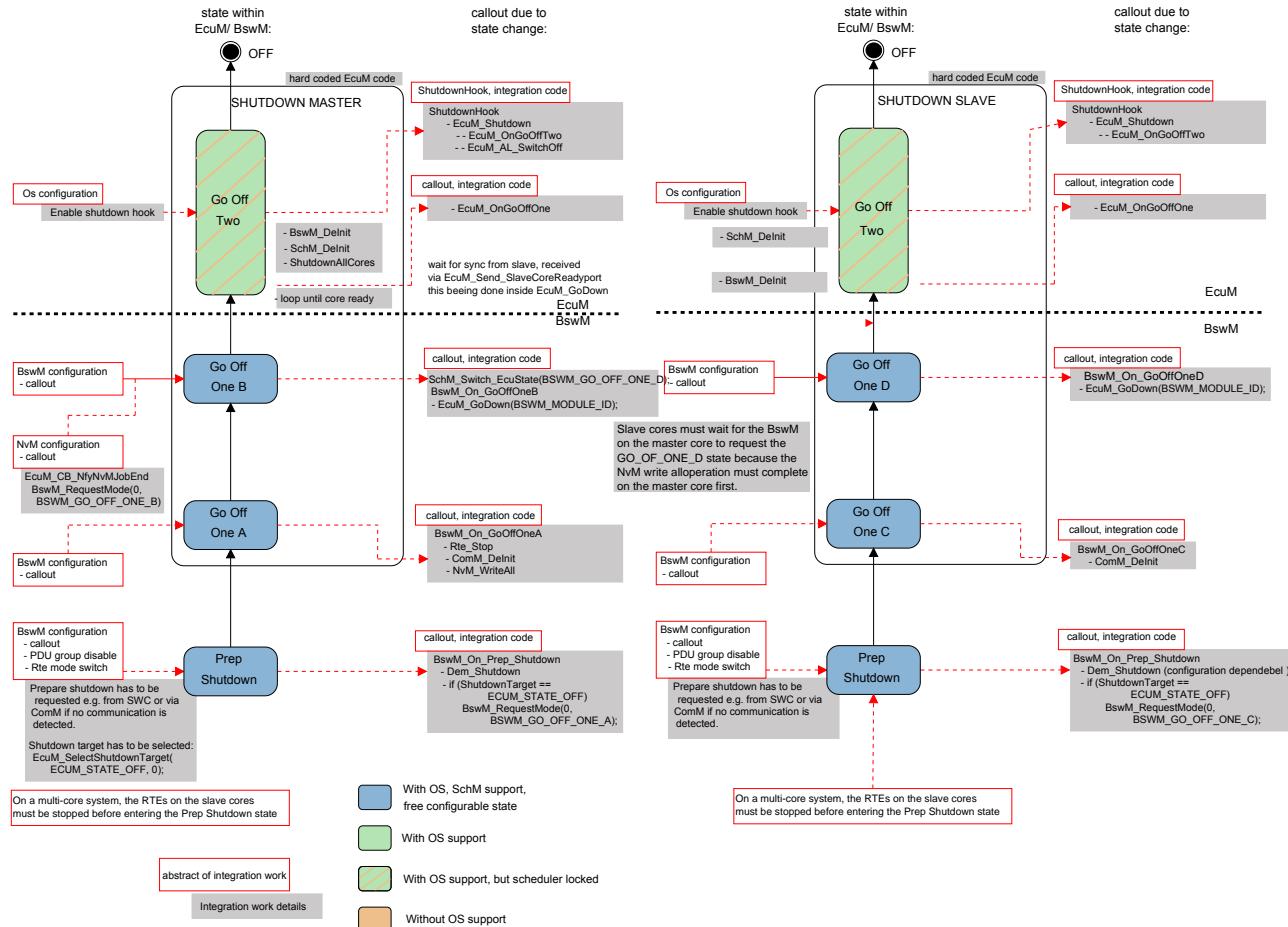


Figure 4.8. The shutdown states on a system with multi-core

#### 4.3.4.3. Configuring sleep on systems with multiple cores

To configure sleep on a multi-core system, take the following steps:

On the master core:

1. Configure the master core to notify the slave core to go to sleep mode, when the master is entering PrepShutdown state via mode switch interface (inter-core communication).
2. Configure BswM to call `EcuM_GoPoll()` as the last step before entering sleep state.

On the slave core:

1. When the notification to go to the sleep state comes from the master core, PrepShutdown must be executed on the slave core.
2. Configure BswM to trigger the PrepShutdown when the notification is received.
3. Configure BswM to call `EcuM_GoPoll()` as the last step before entering sleep state.



Below you have a diagram depicting an example on how the `sleep` mode is working on an ECU with one master core and one slave core.

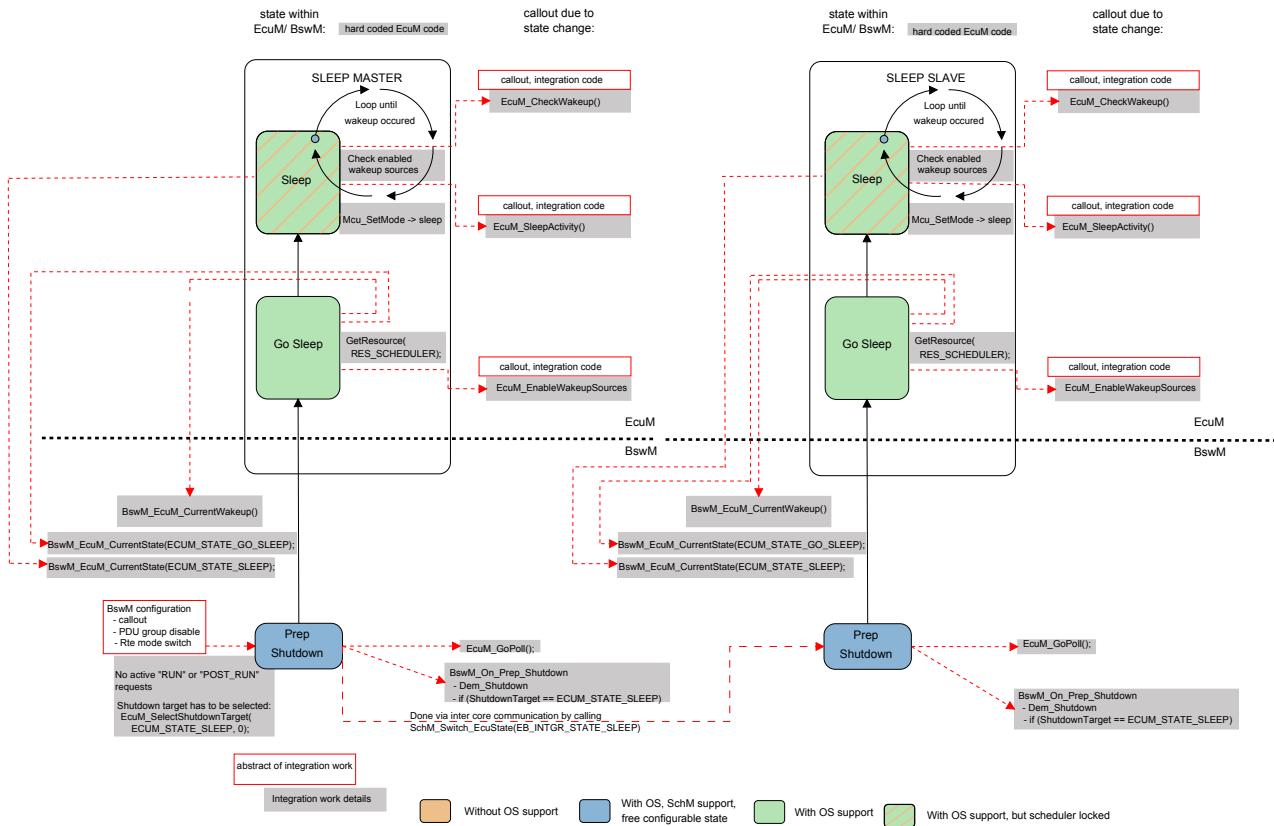


Figure 4.9. The `sleep` states of a system with multi-core

#### 4.3.4.4. Configuring the wakeup state on systems with multiple cores

The `wakeup` state is initiated after a pre-configured wakeup event occurs. To configure `wakeup` on a multi-core system, take the following steps:

On the master core:

1. The callout `EcuM_DisableWakeupsources()` should be configured as it is used for changing the hardware state so it will not report the same wakeup event again.
2. The callout `EcuM_AL_DriverRestart()` should be configured if any driver or CDD requires restart when returning from low power.
3. The callout `EcuM_StartWakeupsources()` should be configured as it is used for changing the hardware state so it can validate a pending wakeup event.



4. The callout `BswM_OnWakeupTwo()` shall be configured so that when a pending wakeup event on the master was validated, a set wakeup event is triggered on the slave core.
5. The callout `EcuM_CheckValidation()` is used to poll for a pending wakeup event in order to determine if it was validated or not.

On the slave core:

1. The callout `EcuM_DisableWakeupSources()` should be configured as it is used for changing the hardware state so it will not report the same wakeup event again.
2. The callout `EcuM_AL_DriverRestart()` should be configured if any driver or CDD requires restart when returning from low power.
3. The callout `EcuM_StartWakeupSources()` should be configured as it is used for changing the hardware state so it can validate a pending wakeup event.
4. The callout `EcuM_CheckValidation()` is used to poll for a pending wakeup event in order to determine if it was validated or not.
5. The callout `BswM_OnWakeupTwo()` shall be configured if integration code must be executed as part of wakeup two.

Below you have a diagram depicting an example on how the wake-up is working on an ECU with one master core and one slave core.

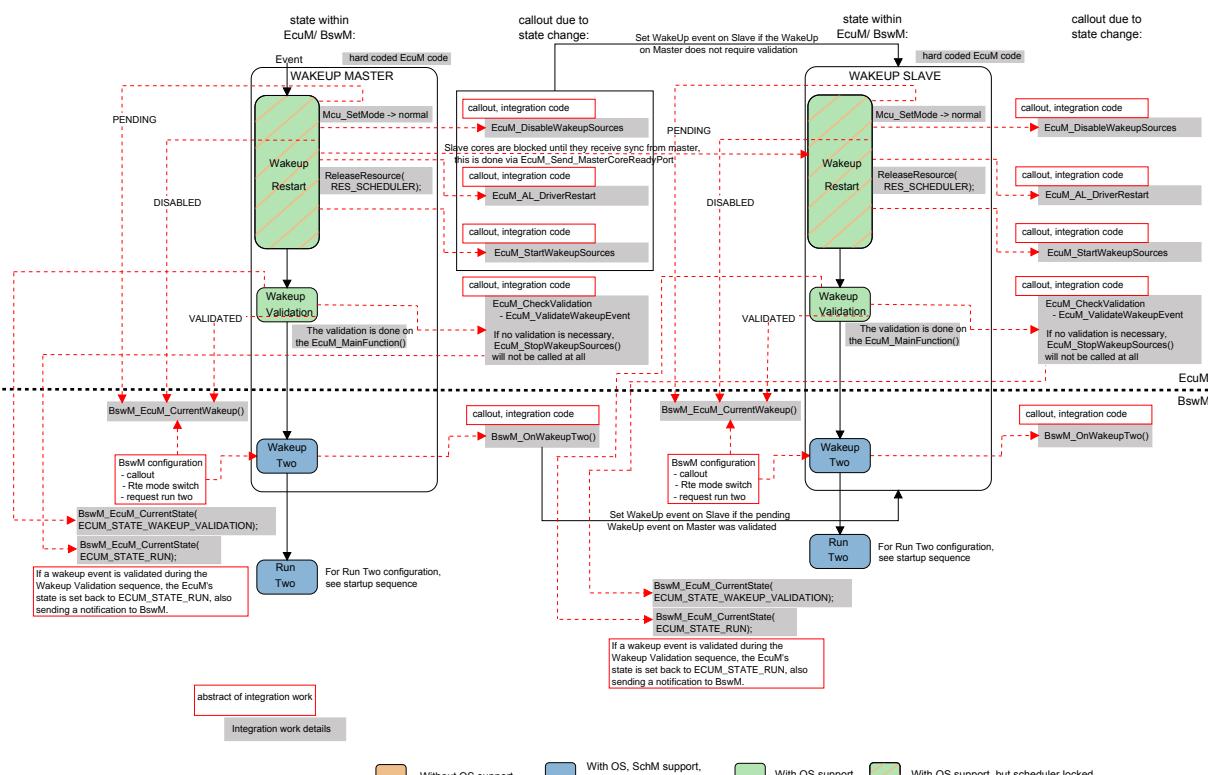


Figure 4.10. The wakeup states of a system with multi-core



---

**NOTE****Synchronizing wake-up between cores**

Depending on the core on which the wake-up event occurs, the other cores should be notified to wake up, too. One way to achieve this is to ensure that the core that receives the wake-up event sets wake-up events for the other cores. This can be done immediately after the wake-up event occurs if no validation is needed or after the validation has taken place if validation is needed.

---

## 4.4. BswM module user guide

### 4.4.1. Overview

This chapter provides you with `BswM`-specific information:

- ▶ [Section 4.4.2, “Background information”](#) explains the concepts of the `BswM` module.
- ▶ [Section 4.4.3, “Configuring the BswM module”](#) provides instructions on how to configure the `BswM` module with the `BswM` Editor. For additional configuration advice, see [Section 4.3, “Configuring the EcuM and the BswM modules”](#).

### 4.4.2. Background information

#### 4.4.2.1. Mode arbitration

The `BswM` module arbitrates mode requests from software components or from basic software modules, applies configurable rules to these requests, and performs configurable actions. In contrast to the `EcuM`, the `BswM` does not provide any state dependent handling. The arbitration, rules and actions that are configured in the `BswM` are executed independently of any state. If state dependent handling is needed, define a state variable externally and then check this state variable within the `BswM` rules.

[Figure 4.11, “Code example of a `BswM` rule”](#) gives an example of one rule configured in the `BswM` with the parameter `BswMRule`. In such a rule, a logical expression is configured via the parameter `BswMLogicalExpression`. This logical expression consists of one or more conditions. Each condition is configured by the parameter `BswMModeCondition`. Request ports are configured to use the rule that has been defined. When a request is received via the request port, the logical expression is evaluated and results either in `TRUE` or

in FALSE. Depending on this result, an action list that is configured with the parameter BswMActionList is executed. Each action list itself includes one or more actions. An action could be for example a user callout. This is configured using the parameter BswMAction.

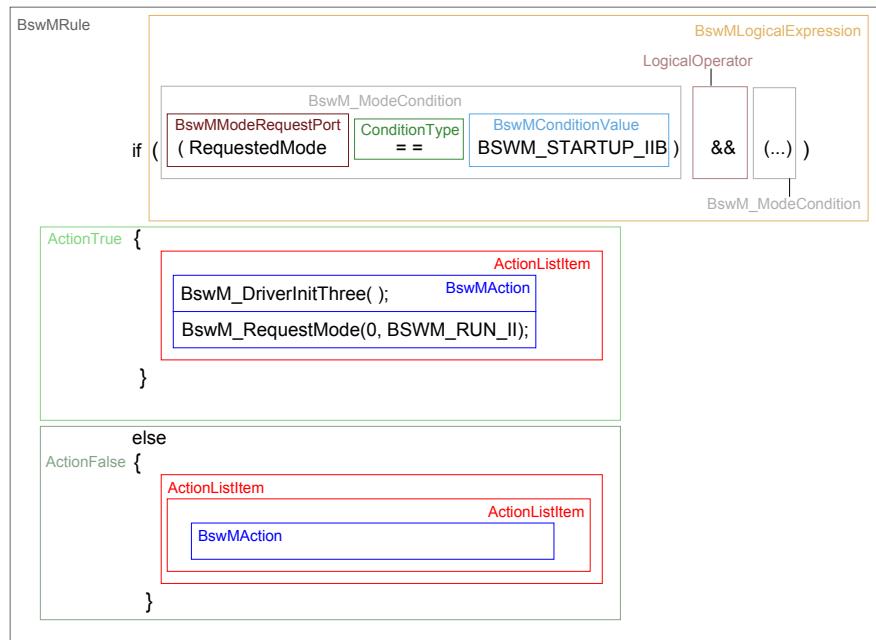


Figure 4.11. Code example of a BswM rule

[Figure 4.12, “Sample containers, references and multiplicity of a BswM rule”](#) shows the configuration containers, the references and their multiplicity in a BswM rule.

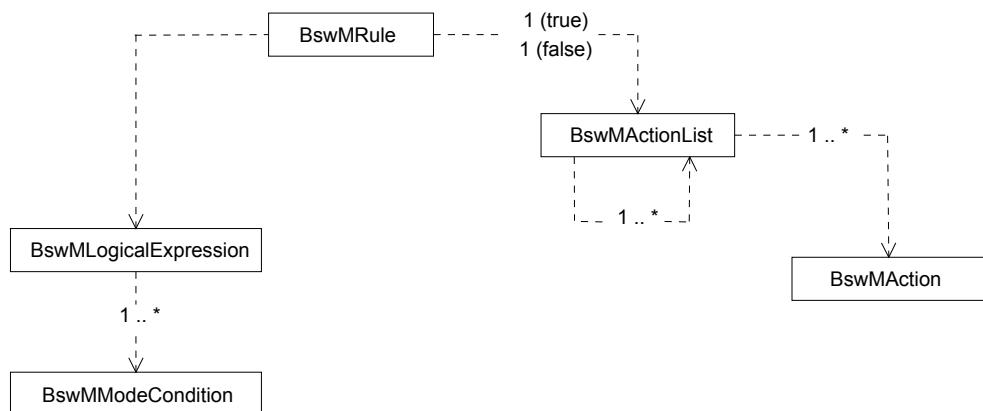


Figure 4.12. Sample containers, references and multiplicity of a BswM rule



#### 4.4.2.2. Queued mode arbitration

##### 4.4.2.2.1. Functional overview

Concurrent processing of mode arbitration requests is a challenging topic for an integrator because reentrancy issues or race conditions are common when the arbitration of a mode request port depends on the execution of an action list.

According to the AUTOSAR Specification of Basic Software Mode Manager, the system integrator is responsible for ensuring that the execution of an action list in the context of an immediate request does not jeopardize the system's consistency.

In order to minimize the risk that such issues affect a complex project, you can use the queued mode arbitration feature to better control the processing of arbitration requests. Instead of allowing concurrent execution, this feature ensures that the processing of an immediate request of any type or the processing of the `BswM_MainFunction()` cannot be interrupted by other requests.

This is achieved by queuing the requests that interrupt the processing of the existing immediate request or the processing of the `BswM_MainFunction()`. The queued requests are executed in the order of their arrival after the processing that they interrupted was completed.

Note that the queued arbitration requests are not executed from their original context anymore, but in the context of the request that they interrupted.

##### 4.4.2.2.2. Configuration options

The queued mode arbitration feature can be enabled via the optional container **BswMQQueuedArbitration**. The following configuration parameters can be used to fine-tune this feature.

The `BswMArbitrationQueueLength` parameter controls the size of the arbitration queue. By default, the value of this parameter is the maximum number of mode request ports in all `BswM` configurations. You can change this value to fit the project's requirements. If the arbitration queue is full and a new request needs to be enqueued, the new request is rejected. In this case, a Det error is reported if `BswMDevErrorDetect` is set to TRUE. Note that a larger queue increases the RAM usage of `BswM`.

The `BswMQueueDeferredRequests` parameter can be used to exclude or include deferred mode request ports from being queued. If set to FALSE, only immediate mode request ports are queued. If set to TRUE, both immediate and deferred mode request ports are queued.

The `BswMDelayMainFunction` parameter can be used to handle the situation when the `BswM_MainFunction()` interrupts the execution of an immediate mode request. If set to FALSE, the `BswM_MainFunction()` executes even if there is an immediate request being processed. Thus, the arbitration feature is bypassed. If set to TRUE, the `BswM_MainFunction()` is delayed until the processing of the immediate mode request is



completed. Note that any other request that interrupts the processing of the immediate request that the `BswM_MainFunction()` interrupted, is also processed before the `BswM_MainFunction()`.

#### 4.4.2.2.3. Configuration examples

##### 4.4.2.2.3.1. Example 1

An immediate generic request `I_GEN_1` is made. While `I_GEN_1` is processed, an immediate generic request `I_GEN_2` and a deferred `BswMCanSMIndication` request `D_CANSM` are made from an interrupt context. Depending on the queued mode arbitration configuration parameters, the following outcomes are possible:

- ▶ `BswMQueuedArbitration` is disabled

`I_GEN_2` and `D_CANSM` interrupt the execution of `I_GEN_1`.

- ▶ `BswMQueuedArbitration` is enabled, `BswMArbitrationQueueLength = 2`, `BswMQueueDeferredRequests = TRUE`

`I_GEN_2` and `D_CANSM` are enqueued, and their mode request handler functions return. After `I_GEN_1` finishes processing both arbitration and execution of action lists, `I_GEN_2` and `D_CANSM` are also processed, in this order. Note that both `I_GEN_2` and `D_CANSM` are processed from the context of the `BswM_RequestMode` corresponding to `I_GEN_1`.

- ▶ `BswMQueuedArbitration` is enabled, `BswMArbitrationQueueLength = 1`, `BswMQueueDeferredRequests = TRUE`

`I_GEN_2` is enqueued but `D_CANSM` is not because the queue is not large enough. Thus, a Det error is reported. The service ID corresponds to `BswM_CanSM_CurrentState`, the error ID is `BSWM_E_QUEUE_FULL`.

After `I_GEN_1` finishes processing both the arbitration and execution of action lists, `I_GEN_2` is processed from the context of the `BswM_RequestMode` corresponding to `I_GEN_1`.

- ▶ `BswMQueuedArbitration` is enabled, `BswMArbitrationQueueLength = 1`, `BswMQueueDeferredRequests = FALSE`

`I_GEN_2` is enqueued but `D_CANSM` is processed as soon as it arrives, from its original `BswM_CanSM_CurrentState` context. After `I_GEN_1` finishes processing, `I_GEN_2` is unqueued and processed from the context of the `BswM_RequestMode` corresponding to `I_GEN_1`.

##### 4.4.2.2.3.2. Example 2

An immediate generic request `I_GEN_1` is made. While `I_GEN_1` is processed, a generic deferred request `D_GEN` is made. The `SchM` triggers the `BswM_MainFunction()`. Afterwards, another generic immediate



request I\_GEN\_2 is performed. Depending on the mode arbitration configuration parameters, the following outcomes are possible:

- ▶ BswMQueuedArbitration is disabled

D\_GEN, BswM\_MainFunction() and I\_GEN\_2 interrupt I\_GEN\_1.

- ▶ BswMQueuedArbitration is enabled, BswMQueueDeferredRequests = TRUE, BswMDelayMainFunction = FALSE

D\_GEN is queued. BswM\_MainFunction() interrupts and executes while I\_GEN\_1 is processed. I\_GEN\_2 is queued. After I\_GEN\_1 finishes, D\_GEN and I\_GEN\_2 are processed in the order as they arrived. BswM\_MainFunction() does not arbitrate D\_GEN in this cycle because D\_GEN was queued before BswM\_MainFunction() executed and the port was not updated.

- ▶ BswMQueuedArbitration is enabled, BswMQueueDeferredRequests = FALSE, BswMDelayMainFunction = FALSE

D\_GEN is executed. The port is only updated because it is deferred. BswM\_MainFunction() interrupts and executes while I\_GEN\_1 is processed and I\_GEN\_2 is queued. BswM\_MainFunction() also arbitrates D\_GEN as D\_GEN updated the port. After I\_GEN\_1 finishes, I\_GEN\_2 is processed.

- ▶ BswMQueuedArbitration is enabled, BswMQueueDeferredRequests = TRUE, BswMDelayMainFunction = TRUE

First, D\_GEN is queued. Then, the BswM\_MainFunction() is delayed because I\_GEN\_1 has not yet finished processing. I\_GEN\_2 is also queued. After I\_GEN\_1 finishes processing, first D\_GEN is executed (port update). Then I\_GEN\_2 is processed. BswM\_MainFunction() is last to be triggered, even if it occurred before I\_GEN\_2. BswM\_MainFunction() arbitrates D\_GEN as D\_GEN updated the port before BswM\_MainFunction() executed.

Note that the BswM\_MainFunction() is first delayed (i.e. returns without performing any arbitration or control), and is later triggered by the BswM\_RequestMode corresponding to I\_GEN\_1.

#### 4.4.2.3.3. Example 3

If the BswM\_MainFunction() is interrupted by an immediate generic request I\_GEN\_1, then I\_GEN\_1 is queued and only processed after BswM\_MainFunction() finishes. Deferred requests are queued only if BswMQueueDeferredRequests = TRUE. This is the same behavior as in the above examples.

#### 4.4.2.3. Multi-core and BSW distribution support

In the BswM, multi-core support is implemented based on the "multi partition" support concept defined by AUTOSAR. In this "multi partition" support implementation, one instance of the BswM is assigned to each partition.



#### 4.4.2.3.1. Functional overview

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. Therefore, the `BswM` is distributed over multiple partitions that contain BSW modules. These independent `BswM` instances have partition specific configuration sets. The synchronization of the different instances is done via BSW modes.

All `BswM` instances are independent of each other and have their own configuration. This means that if a mode request is triggered in a context of a certain `BswM` instance, only that instance will perform the mode control and mode arbitration. Sometimes, however, it might be necessary to propagate a mode request to all `BswM` instances. This situation typically arises with single instance modules that can only trigger a mode change to one `BswM` instance only (on the partition they are located). This can, however, be achieved by using a `SchM` switch action on the partition that receives the mode change.

As an example, suppose there are 2 partitions, each on its own core. Both of these partitions have a `BswM` instance but only the first one has a `ComM` instance.

The `ComM` triggers a `BswM_ComM_CurrentMode` (`Network`, `RequestedMode`) mode request which needs to produce changes on both `BswM` partitions. Suppose there is a rule that triggers changes to the first `BswM` instance using an action list `ComM_ActionList`. In order to also notify the second `BswM` instance, a `SchM` switch action needs to be added to the `ComM_ActionList`. A `BswM` switch port is also required to configure this action and the user must use the proper mode declaration group provided by the `ComM` for this type of mode request. A custom mode declaration group can also be used.

The `SchM` switch action will switch the mode of the switch port to a value corresponding to the `RequestedMode` if the `ComM` request. On the second `BswM` instance, a BSW mode notification port needs to be added. This port needs to reference the same mode declaration group via the switch interface specified on the switch port from the first `BswM` instance. A rule needs to be added on the second `BswM` instance. This rule will verify that the BSW mode notification port is equal to the required mode declaration group value that was switched by the first instance via the switch port.

In order to finish the setup, the RTE Editor's BSW Mode Mapping tool needs to be used in order to connect the provided mode group corresponding to the switch port on the first instance to the required mode group corresponding to the BSW mode notification port on the second instance.

For a better understanding of how this works and what mode arbitration, rule, mode conditions, logical expressions, interfaces and ports do, read from the `AUTOSAR_SWS_BSWModeManager.pdf` chapter called "Functional specification".

#### 4.4.2.3.2. Configuration options

To use the **BSW Distribution** feature in the `BswM`, configure the parameter `BswMPartitionRef`. Refer to [Section 4.3, "Configuring the EcuM and the BswM modules"](#) for further integration and configuration instructions.



#### 4.4.2.4. Post-build selectable support

BswM is variant-aware and allows you to configure different values in different variants for several configuration parameters. These parameters are either existing post-build changeable containers that can have a different number of instances in different selectable variants. Or they are existing configuration parameters with the implementation config class set to `PostBuild` that can have different values in different selectable variants.

If there are multiple post-build selectable variants defined, the `BswM` produces different `BswM_{variant}.xgen` files consisting of the specific variant configuration. In this situation, `BswM_PBcfg.c` aggregates all the generated `BswM_{variant}_PBcfg.c` files.

If no different `BswM` configurations are required in a project that features multiple post-build selectable variants, you can use the `POST_BUILD_VARIANT_USED` configuration parameter in order for `BswM` to produce a single `BswM.xgen` file. If you set the `POST_BUILD_VARIANT_USED` to false, the behavior of `BswM` is the same as in a project without post-build selectable variants.

Note that if the selected `IMPLEMENTATION_CONFIG_VARIANT` is `VariantPreCompile`, you cannot set the `POST_BUILD_VARIANT_USED` parameter to true.

### 4.4.3. Configuring the BswM module

#### 4.4.3.1. BswM Editor user guide

##### 4.4.3.1.1. Overview

The BswM Editor is provided to simplify the configuration of the `BswM` module.

For this user guide, it is assumed that you are already familiar with the `BswM` module itself. For information on the `BswM` module, see the EB tresos AutoCore Generic documentation.

- ▶ [Section 4.4.3.1.2, “Background information”](#) provides an overview of the concepts of the BswM Editor.
- ▶ [Section 4.4.3.1.3, “Using the BswM Editor”](#) provides instructions on how to use the BswM Editor.

##### 4.4.3.1.2. Background information

The BswM Editor provides a text-based view of the `BswM` mode request ports, event request ports, mode switch ports, send ports, and rules. The text editor for the `BswM` configuration language provides:

- ▶ Syntax highlighting: keywords are highlighted showing the correct syntax.



- ▶ Smart code completion: use the key combination **Ctrl+Space** to see valid options when editing.
- ▶ Error and warning markers: warning or error symbols are shown at the start of lines that need to be corrected or completed.

Also, an outline view is provided that shows the dependencies between `BswM` ports and the rules that reference them. The outline view allows you to see the functionality of a complex configuration.

When you start the `BswM` Editor, the project's `BswM` ECU configuration, which is stored inside the `BswM.xdm` file, is converted into a special configuration language and loaded into the editor. When you use the `BswM` Editor, all changes are continuously saved internally. You can notice an asterisk next to the title of the **BswM Editor** tab.

If you close the **BswM Editor** tab, a confirmation window appears and you can save the changes to the `BswM` configuration. If you select **Yes**, the changes that are performed in the editor are converted and stored within `BswM.xdm`.

#### 4.4.3.1.2.1. Naming conventions

In order to simplify the understanding of the `BswM` configuration, the names of logical expressions, mode conditions, actions, and action lists can be skipped. The skipped names are generated when the editor is closed and the configuration is saved back to the project's `BswM.xdm`. After reopening the configuration, the generated names are displayed.

The replacement starts with the name of the container followed by a specific infix. For logical expressions, the container names are followed by an infix related to the rule that the logical expression references, i.e. `BswMLogicalExpression_{Rule}`.

For mode conditions, the container names are not followed by any sequence, i.e. `BswMModeCondition`.

For action lists, the container names are followed by an infix related to the rule that references them and also followed by an infix that specifies if it is a true or false action list (e.g. `BswMActionList_{Rule}_-TrueActionList`).

For actions, the container names are followed by the action type (i.e. `BswMAction_{ActionType}`).

All replacements end with the index of the respective item (e.g. `BswMLogicalExpression_MyRule_0`).

The names of rules, mode request ports, event request ports, send ports, and switch ports remain the same.

The `BswM` Editor also performs some optimizations in order to reduce the size of the configuration stored in the `BswM.xdm` file.

For example, if an action or mode condition is used with the same structure in two different rules in the `BswM` Editor, then only one container will be created for that action or mode condition.



This optimization does not apply to action lists and logical expressions. Even if two action lists have the same structure used in two different rules, two different containers will be created. This is done in order to keep the configuration intuitive. Also the optimization for elements with names is based on the name only, not on the structure.

In order to keep the configuration valid, the BswM Editor automatically adjusts the values of the `BswMMaxNumRules` and `BswMMaxNumActionLists` parameters. This adjustment might happen if new rules are added or the optimization process changes the number of action lists.

As the BswM Generic Editor is more flexible and it is easy to create configurations that are invalid, BswM Editor grammar contains some workarounds to open incomplete configurations and save as many cases as possible. The only case that the BswM Editor cannot save is when the configuration contains rules, mode request ports, event request ports, send ports, or switch ports that do not have a name. In that case the BswM Editor opens and shows the configuration, but it will not be able to save it until the names are provided. In order to indicate that your action is required, the place that should contain the name is marked with `!!no-name!!`. The following code shows a mode request port that does not have a name:

```
Port GenericRequest !!no-name!! evaluate deferred{
    ModeRequesterId=0;
    initialValue="0023";
}
```

You must replace this indicator with a name in order to save the configuration.

Be aware that if you remove an item's name in the Generic Editor and if the item is referenced by other items, the relations between the items will be lost. To repair the configuration, you need to manually update the names and also the proper references in the BswM Editor.

#### 4.4.3.1.3. Using the BswM Editor

- ▶ [Section 4.4.3.1.3.1, “Starting the BswM Editor”](#) describes basic steps on how to use the editor.
- ▶ [Section 4.4.3.1.3.2, “Creating a new mode request port”](#) provides an example on how to create a new mode request port.
- ▶ [Section 4.4.3.1.3.3, “Creating a new mode switch port”](#) provides an example on how to create a new mode switch port.
- ▶ [Section 4.4.3.1.3.4, “Creating a new event request port”](#) provides an example on how to create a new event request port.
- ▶ [Section 4.4.3.1.3.5, “Creating a new send port”](#) provides an example on how to create a new send port.
- ▶ [Section 4.4.3.1.3.6, “Creating a new rule”](#) provides an example on how to create a new rule.
- ▶ [Section 4.4.3.1.3.7, “Creating a new expression”](#) provides an example on how to create a new logical expression.



- ▶ [Section 4.4.3.1.3.8, “Creating a new condition”](#) provides an example on how to create a new condition.

#### 4.4.3.1.3.1. Starting the BswM Editor

To use the BswM Editor, add the `BswM` module to your project configuration and proceed as follows:

1. Open the **Generic Editor** of the BswM module. This loads the current configuration for the `BswM` module.
2. Switch to the **Configuration** tab and add a configuration. Multiple configurations are supported.
3. Address any errors that you have in the current configuration. You can only start the BswM Editor if the existing configuration is valid.
4. Save the `BswM` module configuration.
5. Open the BswM Editor of the `BswM` module.
6. Configure the options on the **General** tab according to your project requirements.
7. Switch to your **BswM Configuration** tab.
8. Use this context-sensitive editor to create mode requests, switch ports, and rules. [Figure 4.13, “BswM Editor”](#) shows the editor with an example mode request. Further examples are given in later sections.

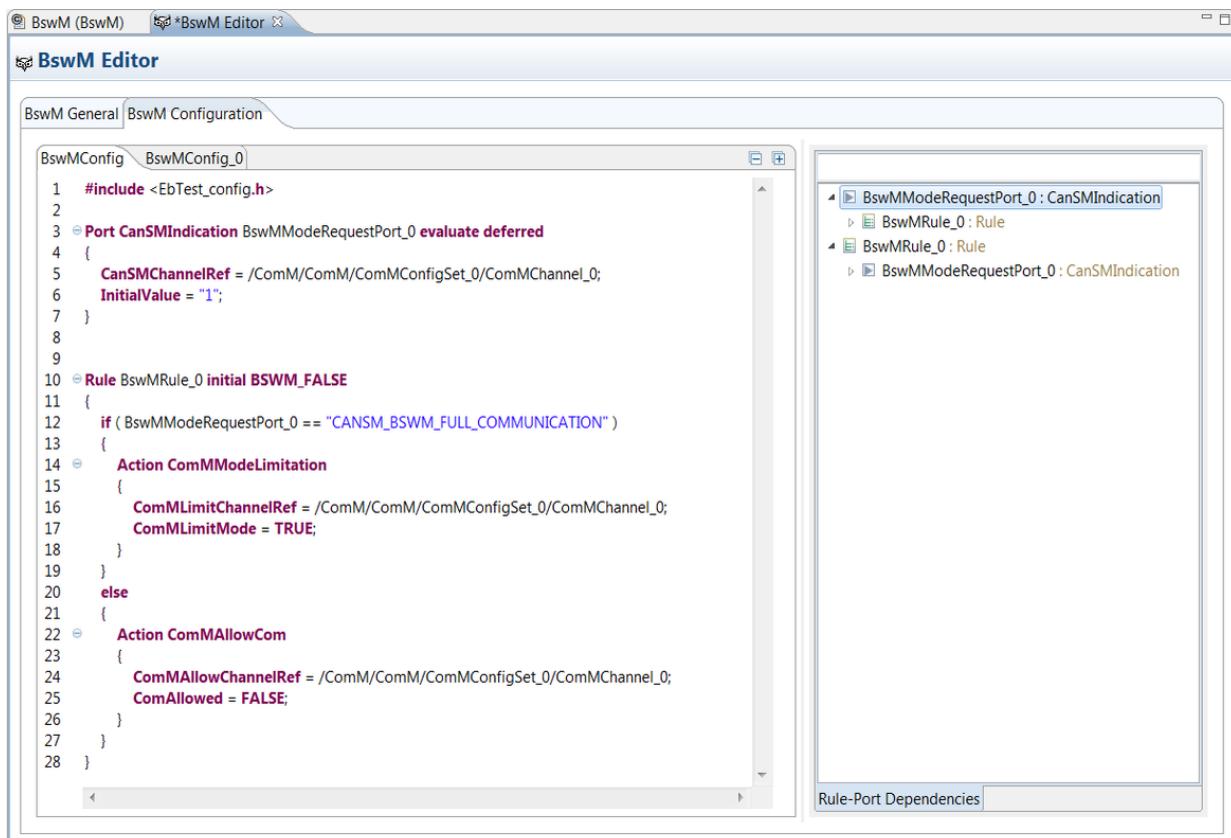


Figure 4.13. BswM Editor



9. Click the **Close** button on the **BswM Editor** tab to end the editing session.
10. Save your changes when prompted.
11. Save the `BswM` module configuration to permanently store your changes in the `BswM.xdm` file.

#### 4.4.3.1.3.2. Creating a new mode request port

You can define the following mode request ports at any point within the editor pane:

- ▶ `BswMBswModeNotification`
- ▶ `BswMCanSMIndication`
- ▶ `BswMComMIndication`
- ▶ `BswMComPncRequest`
- ▶ `BswMDcmComModeCurrentState`
- ▶ `BswMDcmComModeRequest`
- ▶ `BswMDcmResetModeRequest`
- ▶ `BswMDcmSessionModeRequest`
- ▶ `BswMEcuMIndication`
- ▶ `BswMEcuMWakeupSource`
- ▶ `BswMEthIfPortGroupLinkStateChg`
- ▶ `BswMEthSMIndication`
- ▶ `BswMFrSMIndication`
- ▶ `BswMGenericRequest`
- ▶ `BswMJ1939DcmBroadcastStatus`
- ▶ `BswMJ1939NmIndication`
- ▶ `BswMLinScheduleIndication`
- ▶ `BswMLinSMIndication`
- ▶ `BswMLinTpModeRequest`
- ▶ `BswMNvMJobModeIndication`
- ▶ `BswMNvMRequest`
- ▶ `BswMRteSwitchAckNotification`
- ▶ `BswMSdClientServiceCurrentState`
- ▶ `BswMSdConsumedEventGroupCurrentState`
- ▶ `BswMSdEventHandlerCurrentState`



- ▶ BswMSwcModeNotification
- ▶ BswMSwcModeRequest
- ▶ BswMTimer

The following code defines a new `BswMGenericRequest` mode request port:

```
Port GenericRequest BswM_ReqPort evaluate deferred
{
    ModeRequesterId = 0;
    RequestedModeMax = 5;
    initialValue = "BSWM_STARTUP_TWO_A";
}
```

The definition of this mode request port begins with the keyword `GenericRequest`. The user-defined name `BswM_ReqPort` follows. After the keyword `evaluate`, put either `deferred` or `immediate`. Use `deferred` if the value of the mode request port should be evaluated when the `BswM` main function is executed. Use `immediate` if the value of the mode request port should be immediately evaluated when the port is updated.

The following syntax diagram shows an example of the possible options for a generic request. Keywords and syntax are shown in clear boxes while objects that you provide are highlighted in gray.

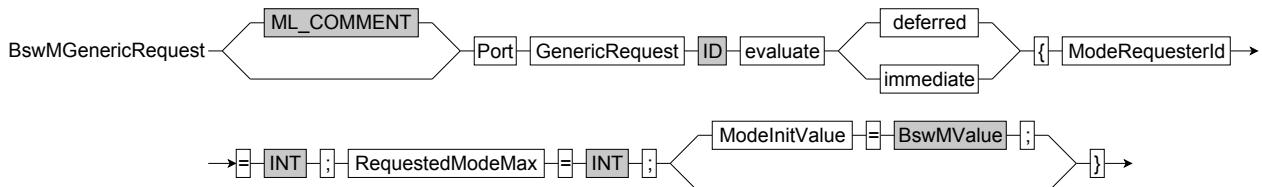


Figure 4.14. `BswMGenericRequest`

A generic mode request port has the following attributes:

- ▶ `ModeRequesterId`: This attribute corresponds to the configuration parameter `BswMModeRequesterId`.
- ▶ `RequestedModeMax`: This attribute corresponds to the configuration parameter `BswMRequestedModeMax`.
- ▶ `InitialValue`: This attribute corresponds to the configuration parameter `BswMModeInitValue`.

As you can see in the example above, the mode request port data structure keywords correspond almost completely to the ECU configuration parameter names. The prefix `BswM` has been removed though.

#### 4.4.3.1.3.3. Creating a new mode switch port

The following example defines a `BswMSwitchPort`:



```

SwitchPort BswMSwitchPort_Rte
{
    SwitchInterfaceRef = /BswMMode/PortInterfaces/BswMMode;
    DataMapping = /BswMMode/DataTypeMappingSets/BswMModeMapping;
    ModeSwitchQueueLength = 1;
}

```

The definition of a switch port begins with the keyword `SwitchPort` followed by the user-defined name. The following attributes must be defined for each switch port:

- ▶ `SwitchInterfaceRef`: This is the path to the mode switch interface to be used.
- ▶ `DataMapping`: This is the path to the data type mapping for the mode declaration group used within the referenced mode switch interface.
- ▶ `ModeSwitchQueueLength`: The number of mode switches that can be queued.

Switch ports are referenced by `RteSwitch` and `SchMSwitch` actions, which are defined within rule action lists.

The following syntax diagram shows an example of the possible options for a switch port. Keywords and syntax are shown in clear boxes while objects that you provide are highlighted in gray.

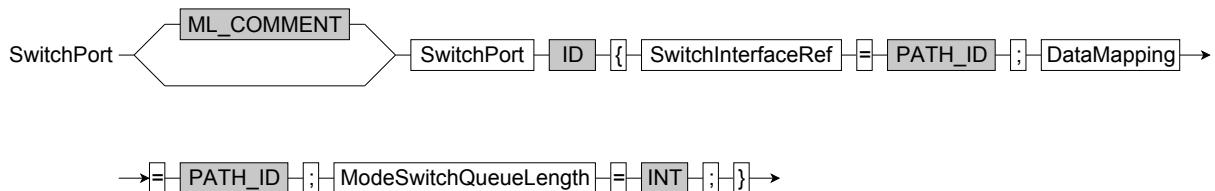


Figure 4.15. SwitchPort

#### 4.4.3.1.3.4. Creating a new event request port

There are two types of event request ports: actual `EventRequestPorts` and legacy event request ports implemented as `ModeRequestPorts`. The list of actual `EventRequestPorts` currently includes only one port:

- ▶ `BswMComMInitiateReset`

The following example defines an `EventRequestPort` of type `ComMInitiateReset`:

```

EventPort ComMInitiateReset eventPort evaluate immediate
{
}

```



The definition of an `EventRequestPort` begins with the keyword `EventPort` followed by the user-defined name. The keyword `ComMInitiateReset` follows. After the keyword `evaluate`, put either `deferred` or `immediate`:

- ▶ Use `deferred` if the value of the event request port should be evaluated when the `BswM` main function is executed.
- ▶ Use `immediate` if the value of the event request port should be immediately evaluated when the port is updated.

`EventRequestPorts` are referenced by `BswMClearEventRequests` actions, which are defined within rule action lists.

To create legacy event request ports, see [Section 4.4.3.1.3.2, “Creating a new mode request port”](#). The list of legacy event request ports includes the following three ports:

- ▶ `BswMDcmApplicationUpdatedIndication`
- ▶ `BswMWdgMRequestPartitionReset`
- ▶ `BswMNmIfCarWakeUpIndication`

#### 4.4.3.1.3.5. Creating a new send port

The following example defines a `BswMSendPort`:

```
SendPort BswMSendPort_A
{
    VariableDataPrototypeRef = /CounterDemo/IfCounter/CounterValue;
}
```

The definition of a send port begins with the keyword `SendPort` followed by the user-defined name of the send port. The following attribute must be defined for each send port:

- ▶ `VariableDataPrototypeRef`: This is the path to the variable data prototype to be used.

The following syntax diagram shows an example of the possible options for a send port. Keywords and syntax are shown in clear boxes while objects that you provide are highlighted in gray.

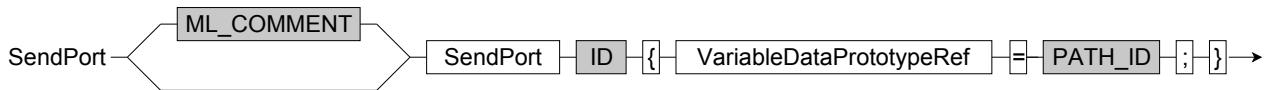


Figure 4.16. SendPort



#### 4.4.3.1.3.6. Creating a new rule

In the example below, a rule is defined that executes a triggered action list if the mode request port `BswM_ReqPort` is set to `BSWM_STARTUP_TWO_A`:

```

Rule BswMRule_0 initial BSWM_UNDEFINED
{
    if triggered (( BswM_ReqPort == "BSWM_STARTUP_TWO_A" ))
    {
        ActionList
        {
            Action UserCallout
            {
                UserCalloutFunction = "BswM_OnStartupTwoA()";
            }
            Action RteSwitch
            {
                RteSwitchPortRef = BswMSwitchPort_Rte;
                SwitchedMode =
                    /BswMMode/ModeDeclarationGroups/BswMModeGroup/BSWM_STARTUP_TWO_A;
            }
        }
    }
    else triggered
    {
        ActionList
        {
            Action UserCallout
            {
                UserCalloutFunction = "MyCalloutFunction()";
            }
        }
    }
}

```

A rule begins with the keyword `Rule`. The user-defined name follows. The keyword `initial` is followed by one of the following:

- ▶ `BSWM_TRUE`
- ▶ `BSWM_FALSE`
- ▶ `BSWM_UNDEFINED`

This defines how the `BswM` rule's triggered action list(s) shall be executed when the rule is first evaluated. If the rule does not have a triggered action list, the initial value does not have any influence on the rule's behavior.



A rule must have an if-statement. The if-statement begins with the keyword `if`. If the optional keyword `triggered` is added, the true action list is executed. This means the rule's result must change before the true action list can be executed again. The rule's logical expression follows. In this example, a primitive expression is defined, but complex expressions with logical `and`, `or`, `xor`, and `nand` can be used. For details on defining if-statements, see [Section 4.4.3.1.3.7, “Creating a new expression”](#). Within the body of the if-statement, the true action list is defined. An action list can execute multiple actions, (nested) action lists or (nested) rules. In this example, a user-callout action is executed, followed by an `Rte` switch action.

The `else`-statement is optional. This element defines the false action list. Like the true action list, the optional keyword `triggered` can be defined after the keyword `else` and before the statement body.

The following syntax diagram shows an example of the possible options for a rule. Keywords and syntax are shown in clear boxes while objects that you must provide are highlighted in gray.

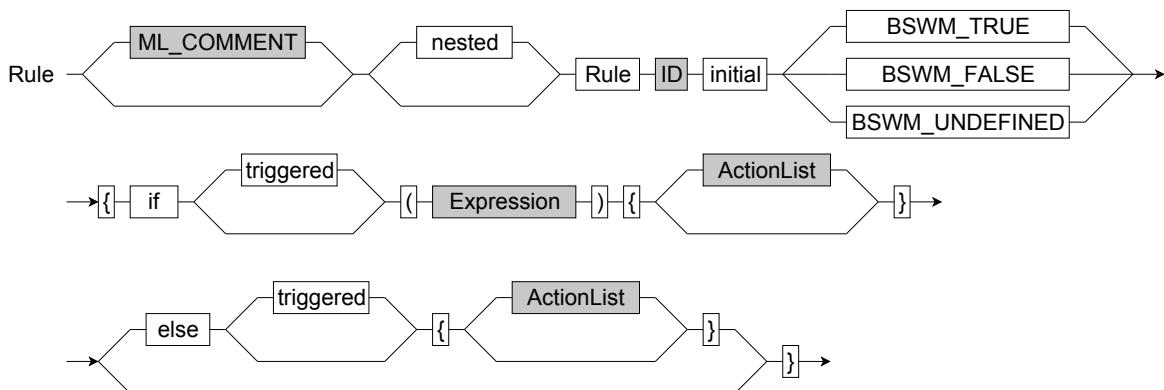


Figure 4.17. Rule

#### 4.4.3.1.3.7. Creating a new expression

Expressions are part of a rule and describe conditions that should be fulfilled to trigger a rule's action lists. The expression syntax is very flexible. Expressions are defined within an if-statement that contains the keyword `if` and a pair of parentheses:

```
if( <<EXPRESSION>> )
```

The syntax of an expression is as follows:

```
<<BSWM_LOGICAL_EXPRESSION_NAME>>: <<LOGICAL_OPERATOR>> ( <<EXPRESSION_ELEMENTS>> )
```



```
BswMLogicalExpression_{Rule})
```

An expression can have the following attributes:

- ▶ **BSWM\_LOGICAL\_EXPRESSION\_NAME**: The name of the expression that is used by `BswM` to identify the `BswMLogicalExpression` element. This attribute can be skipped together with the colon (:) that follows it.
- ▶ **LOGICAL\_OPERATOR**: An optional logical operator. This attribute can be one of the following values: OR, AND, XOR, NAND. It can be skipped if no logical operator is needed, i.e. if the expression contains one expression element only.
- ▶ **EXPRESSION\_ELEMENTS**: A set of expression elements that the expression contains. The elements can be conditions or other logical expressions. Elements should be separated with a comma.

Note that if you skip `BSWM_LOGICAL_EXPRESSION_NAME` and `LOGICAL_OPERATOR`, the expression should still contain the parentheses. This means that if you have a simple expression with a skipped name and no operator, the if-statement should have two pairs of parentheses:

```
if( ( port == "0" ) )
```

The inner pair of parentheses belongs to the expression. The outer pair of parentheses belongs to the if-statement. For the syntax of the conditions, see [Section 4.4.3.1.3.8, “Creating a new condition”](#).

#### 4.4.3.1.3.8. Creating a new condition

Conditions are defined within logical expressions.

You can define three types of conditions:

- ▶ [Mode conditions](#)
- ▶ [Event conditions](#)
- ▶ [Legacy event conditions](#)

##### 4.4.3.1.3.8.1. Mode conditions

Mode conditions are defined with the following syntax:

```
<<CONDITION_NAME>>: <<PORT_NAME>> <<OPERATOR>> <<VALUE>>
```

A mode condition can have the following attributes:

- ▶ **CONDITION\_NAME**: The condition name that is used by `BswM` to identify the `BswMModeCondition` element. This attribute can be skipped together with the colon (:) that follows it.



- ▶ **PORT\_NAME**: The name of the BswMModeRequestPort that this condition relates to.
- ▶ **OPERATOR**: The equality operator. Two values are possible: == and !=.
- ▶ **VALUE**: The value of the port. It can be a simple value, CompuScale or ModeDeclaration.

The following example defines a simple mode condition:

```
BswMModeCondition_1: port1 == "0"
```

Note that the `BswMModeCondition_1 : name` could be skipped. In this case the name is autogenerated when the configuration is saved.

#### 4.4.3.1.3.8.2. Event conditions

Event conditions are defined with the following syntax:

```
<<CONDITION_NAME>>: <<EVENT_PORT_NAME>> is <<STATE>>
```

An event condition can have the following attributes:

- ▶ **CONDITION\_NAME**: The condition name that is used by BswM to identify the BswMModeCondition element. This attribute can be skipped together with the colon (:) that follows it.
- ▶ **EVENT\_PORT\_NAME**: The name of the BswMEventRequestPort that this condition relates to.
- ▶ **STATE**: The state of the event port. Two values are possible: set and cleared.

The following example defines an event condition:

```
BswMModeCondition_1: evPort1 is set
```

Note that the `BswMModeCondition_1 : name` could be skipped. In this case the name is autogenerated when the configuration is saved.

#### 4.4.3.1.3.8.3. Legacy event conditions

Legacy event conditions are defined with the following syntax:

```
<<CONDITION_NAME>>: <<PORT_NAME>>
```



A legacy event condition can have the following attributes:

- ▶ `CONDITION_NAME`: The condition name that is used by `BswM` to identify the `BswMModeCondition` element. This attribute can be skipped together with the colon (:) that follows it.
- ▶ `PORT_NAME`: The name of the `BswMModeRequestPort` that this condition relates to. Note that only the legacy event ports listed in [Section 4.4.3.1.3.4, “Creating a new event request port”](#) are compatible.

The following example defines a legacy event condition:

```
BswMModeCondition_1: legacyPort
```

Note that the `BswMModeCondition_1: name` could be skipped. In this case the name is autogenerated when the configuration is saved.

## 4.5. ComM module user guide

### 4.5.1. Overview

This chapter provides you with `ComM`-specific information:

- ▶ [Section 4.5.2, “Background information”](#) explains the concepts of the `ComM` module.
- ▶ [Section 4.5.3, “Configuring the ComM module”](#) provides instructions on how to configure the `ComM` module.

### 4.5.2. Background information

This chapter describes specific features of the `ComM` module. For background information on the communication management, see chapter *Concepts* in the EB tresos AutoCore Generic documentation.

#### 4.5.2.1. BSW distribution

The `ComM` provides mechanisms for handling communication mode and network management when the communication stack is distributed over multiple cores and one bus type (e.g CAN, FR, IP, LIN) is mapped to one core. For the general concept of BSW distribution of the communication mode management, see chapter *Multi-core support* in the EB tresos AutoCore Generic documentation.



#### 4.5.2.1.1. Functional overview

In a multi-core context, the communication management functionality is processed on the master core. This means that both channel and partial network state machines are processed on this master core.

The `ComM` interacts with `Nm` and `Bussm` during the communication startup or communication shutdown. With a distributed communication stack, the `ComM` uses direct function calls when calling the `Nm` because `ComM` and `Nm` are mapped to the same core. The `ComM` uses cross-core `SchM_Call()` operations when calling a bus-specific state manager `Bussm`.

In case of partial networking, the `ComM` receives and transmits partial network `Com` signals. As the signals are processed locally on a core, the `ComM` provides implementations for satellites. The number of satellites depends on the number of cores that process the partial network signals. The satellites only provide the functionality for exchanging partial network information (PNI) between the master where the PNI is processed and the satellites where the PNI is received or transmitted by using core-local calls of `Com_SendSignal` and `Com_ReceiveSignal`.

Depending on the project use case, the diagnostic communication requests to `ComM` can be triggered from the same core or from a different core. In order to support both cases, the calls between `Dcm` and `ComM` are handled by using `SchM_Call()`.

You can enable or disable the multi-core support for `ComM`. When multi-core support is disabled, the `ComM` always uses direct function calls because all involved modules are mapped to the same core. When multi-core support is enabled, the `ComM` always uses `SchM_Call()`. You must ensure that the integrated modules are compatible and have multi-core support enabled.

The `ComM` multi-core BSW distribution support is enabled with the configuration parameter `ComMMultiCoreSupport`. When this parameter is enabled, the generated basic software module description includes:

- ▶ all required and provided client-server entries, based on the number of configured channels
- ▶ all implementations for the `ComM` satellites, based on the use of partial networking and the PNI signals
- ▶ all required and provided variable data prototypes, based on the configured PNI signals

The `ComM` generates multiple `BswImplementations` based on the following information:

- ▶ assignment of the `ComM` master core
- ▶ mapping of the partial network signals to PDUs
- ▶ mapping of the `Com` main processing functions

For information on how to configure the `ComM` BSW distribution support, see [Section 4.5.3, “Configuring the ComM module”](#)



## 4.5.3. Configuring the ComM module

### 4.5.3.1. Configuring ComM for BSW distribution

In this section, you configure the ComM for BSW distribution. For more information on the ComM BSW distribution, see [Section 4.5.2.1, “BSW distribution”](#).



#### Configuring ComM for BSW distribution

Prerequisite:

- All integrated modules are compatible and have BSW distribution enabled.
- In the Rte, the following is configured:
  - ▶ The module instances, i.e. the BswImplementationS, are properly mapped to the cores via OsAppli-cations.
  - ▶ Events are properly mapped to tasks.
  - ▶ Provided and required client-server connections are properly connected.
  - ▶ Provided and required sender-receiver connections are properly connected.

#### Step 1

To enable the ComM BSW distribution support, enable the configuration parameter ComMMultiCoreSupport.

#### Step 2

To define the ComM master core, reference the corresponding EcuCPartition in configuration parameter ComMMasterCoreEcuCPartitionRef.

#### Step 3

For each asynchronous interface between ComM and BusSm, connect the provided and required client-server entries for each channel. Based on this information, the Rte generates the SchM\_Call() operations that are used for inter-core communication.

#### Step 4

For each asynchronous interface between ComM and Dcm, connect the provided and required client-server entries for each channel. Based on this information, the Rte generates the SchM\_Call() operations that are used for inter-core communication.



## 4.6. EcuM module user guide

### 4.6.1. Overview

This chapter provides you with `EcuM`-specific information:

- ▶ [Section 4.6.2, “Background information”](#) explains the concepts of the `EcuM` module.
- ▶ For information on how to configure the `EcuM` module, see [Section 4.3, “Configuring the EcuM and the BswM modules”](#).

### 4.6.2. Background information

#### 4.6.2.1. EcuM Flex support

EB tresos AutoCore provides the flexible ECU management variant of the `EcuM` module. This variant of the `EcuM` module handles the early `startup` phases, the late `shutdown` phases and the `sleep` phases in which the whole OS scheduling is locked by the `EcuM` taking the OS resource `RES_SCHEDULER`. All other phases and states are completely configurable and handled by the `BswM` module as depicted in [Figure 4.18, “The basic software stack states”](#).

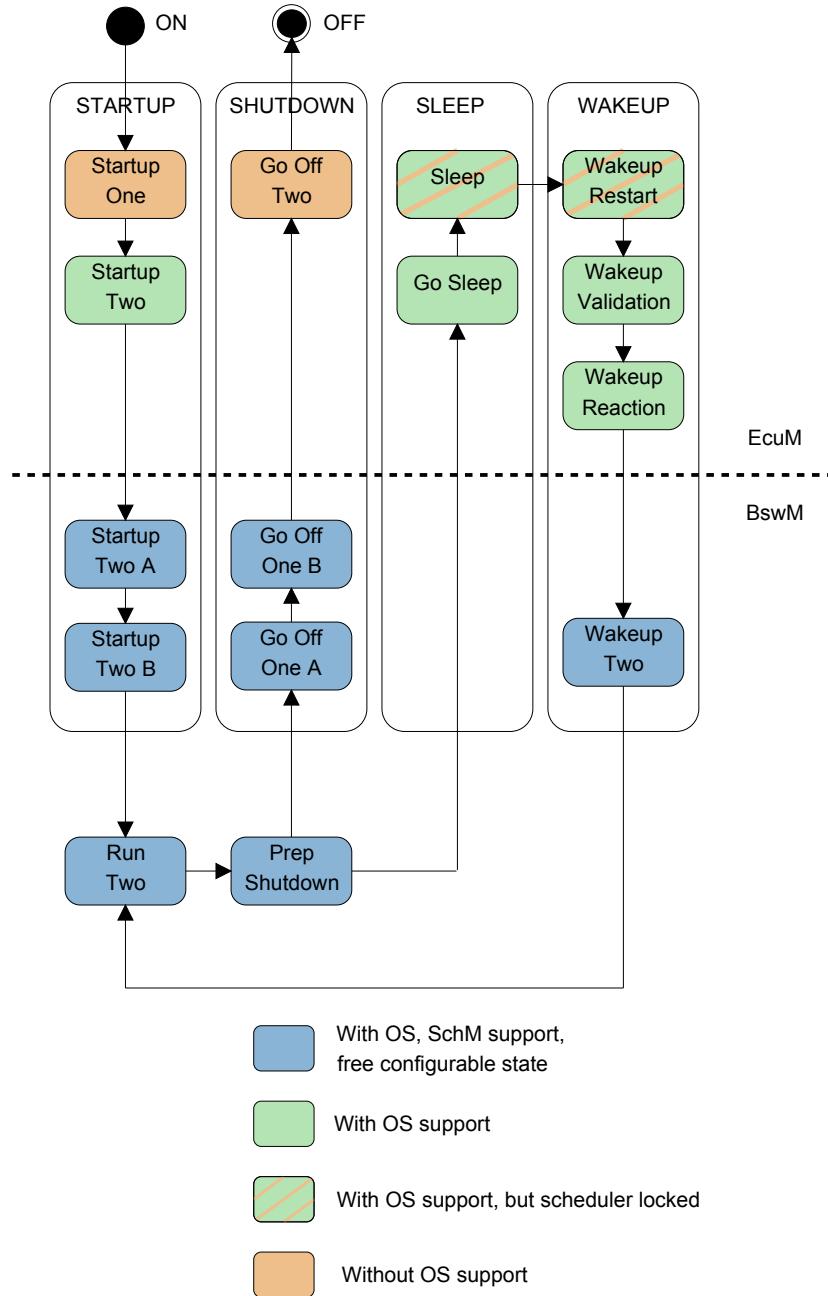


Figure 4.18. The basic software stack states

As the states of the **BswM** module are completely configurable, the states highlighted in blue in [Figure 4.18, "The basic software stack states"](#) show an example of a configuration. However, you may change and adapt this configuration to your system's needs.



#### 4.6.2.2. EcuM Fixed support

Some basic support of the ECU State Manager with a fixed state machine has been implemented in order to allow the integration of modules that are based on AUTOSAR 3.2 in projects that are based AUTOSAR 4.0 or later. Most notably the handling of `RUN` and `POST_RUN` requests has been implemented. The **EcuM Fixed support** feature is enabled by configuring the `EcuMFixedConfiguration` container.

##### 4.6.2.2.1. Functional overview

The main difference between the `EcuM` flexible and the `EcuM` fixed functionality is an extension of the `EcuM` state machine. The state machine is now controlled by five additional API functions that implement the request/release/kill behavior of `RUN` and `POST_RUN` states. A sixth API function can be called to get the status of wakeup sources. In addition, a new callback function for synchronization with the `BswM` module is provided.

After startup, the `EcuM` does not shut down until at least `EcuMRunMinimumDuration/EcuMPeriodMainFunction` calls to `EcuM_MainFunction()` have been performed. During this time, other modules have the possibility to issue `RUN` or `POST_RUN` requests. This then keeps the `EcuM` alive until all such requests are eventually released again.

Because some of the modules subject to the `EcuM` fixed state machine may have to be initialized by the `BswM`, some mechanism is needed to inform the `EcuM` about when these modules are ready. This is necessary, because the initialization may be performed asynchronously within `BswM`. The vendor specific callback function `Std_ReturnType EcuM_GoRun(void)` is provided by the `EcuM` and has to be called by `BswM` when all modules are initialized. Until this callback is called, `EcuM_MainFunction()` will exit without further action if **EcuM Fixed support** is enabled.

All other `EcuM` actions, especially sleep / wakeup / shutdown handling, are left to the `EcuM` flexible implementation, with the only exception of additional calls to `Rte` in order to signal the mode switches. The mode switch `RTE_MODE_EcuM_Mode_WAKE_SLEEP` has been added, at the end of the wakeup procedure ("Wakeup Sequence II" in the SWS). Restart of the system can then be handled by `Rte/BswM`.

The intended usage is as follows:

- ▶ System starts.
- ▶ `EcuM` starts.
- ▶ `EcuM` calls `BswM_Init()`.
- ▶ `BswM` does its initialization tasks, possibly asynchronously.
- ▶ `EcuM_MainFunction()` processes `EcuM` Flex tasks only.
- ▶ After some time, `BswM` initialization has finished -> `BswM` calls `EcuM_GoRun()`.
- ▶ `EcuM_MainFunction()` processes `EcuM` Fixed specific tasks, too.



- ▶ After some time: system goes to sleep.
- ▶ System wakes up again.
- ▶ `EcuM_MainFunction()` processes EcuM Flex tasks only, especially wakeup validation.
- ▶ After wakeup validation has been performed, EcuM signals `WAKE_SLEEP mode to Rte`.
- ▶ BswM does re-initialization tasks, possibly asynchronously.
- ▶ After some time, BswM re-initialization has finished -> BswM calls `EcuM_GoRun()` (e.g. in WAKEUP-II state).
- ▶ Again, `EcuM_MainFunction()` processes EcuM Fixed specific tasks, too.

#### 4.6.2.2. Configuration options

The following additional configuration options are available:

- ▶ `EcuMFixedConfiguration`: Enables **EcuM Fixed support**
- ▶ `EcuMFixedConfiguration/EcuMRunMinimumDuration`: Timeout for RUN requests
- ▶ `EcuMFixedConfiguration/EcuMFixedUserConfig`: List of users which can request a RUN or POST\_RUN state
- ▶ `EcuMDemEventParameterRefs/ECUM_E_ALL_RUN_REQUESTS_KILLED`: EventId reported to DEM when `EcuM_KillAllRUNRequests()` is called.

#### 4.6.2.3. Multi-core and BSW distribution support

In the EcuM, multi-core support is implemented based on the BSW distribution "Master-Satellite" concept defined by AUTOSAR. In this "Master-Satellite" implementation, the functionality of the EcuM module is distributed across multiple cores.

##### 4.6.2.3.1. Functional overview

The basic states for a multi-core system are depicted in [Figure 4.19, “The basic software stack states for a system with multi-core”](#).

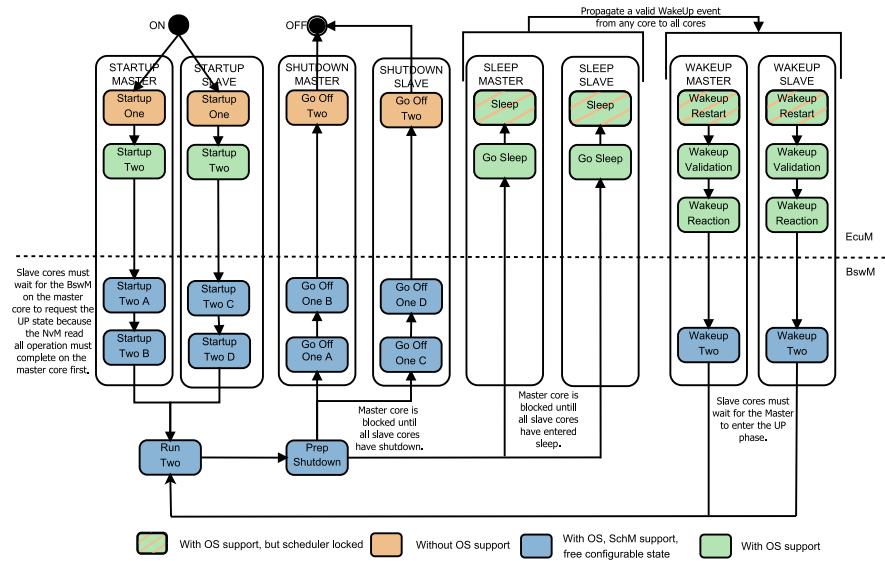


Figure 4.19. The basic software stack states for a system with multi-core

As in the single-core case, the states of the `BswM` module are completely configurable. The states highlighted in blue in [Figure 4.19, “The basic software stack states for a system with multi-core”](#) show an example of a configuration. However, you may change and adapt this configuration to your system's needs.

On a multi-core system, state transitions may encounter delays due to synchronization between the master and slave(s). This synchronization is required when transitioning between the phases mentioned below:

- ▶ STARTUP -> UP
- ▶ UP -> SLEEP
- ▶ UP -> SHUTDOWN
- ▶ SLEEP(WAKEUP) -> UP

Synchronization between master and slave cores is done via the `SchM` sender-receiver API for messaging between the cores. This is a deviation from the AUTOSAR standard as the standard specifies a synchronization method that will result in `Os` errors (RfC#61904).

To signal to the slave cores that the master core is ready, the master core shall send the value `TRUE` to the `EcuM_MasterCoreReadyPort`. To signal to the master core that a slave core is ready, a slave core shall send its `coreId` to the `EcuM_SlaveCoreReadyPort`. The `EcuM` instances need to be configured in the `Rte` configuration, otherwise the code that is generated for the `SchM` sender-receiver functions will be empty.

#### 4.6.2.3.2. Configuration options

To use the **BSW Distribution** feature in the `EcuM`, configure the parameters `EcuMEnableMulticore` and if needed, `EcuMMasterCoreId`. If you do not enable `EcuMMasterCoreId` an implicit value for the master



EcuM core, core-ID 0, is used. Refer to [Section 4.3, “Configuring the EcuM and the BswM modules”](#) for further integration and configuration instructions.

#### 4.6.2.4. Post-build selectable support

The EcuM is implemented with variant handling functionality, allowing multiple module configuration variants to be loaded at the same time.

##### 4.6.2.4.1. Functional overview

If the EcuM is configured with multiple variants, one `EcuM_variant-name_PBCfg.c` file is generated for each variant. This file contains the post-build configuration data structure. A specific variant can be selected by using the `EcuM_DeterminePbConfiguration()` callout function.

This callout function is invoked by `EcuM_Init()`, which is responsible for loading the post-build data structure.

It is not possible to retrieve the post-build data structure from the `PbCfgM` module, since EcuM initializes the `PbCfgM`.

##### 4.6.2.4.2. Configuration options

The following AUTOSAR configuration parameters support post-build selectable variants:

- ▶ `EcuMDefaultAppMode`
- ▶ `EcuMDefaultState`
- ▶ `EcuMDefaultResetModeRef`
- ▶ `EcuMDefaultSleepModeRef`

For the **EcuM Fixed** extension, the `EcuMRunMinimumDuration` parameter supports post-build selectable variants.

In addition to the AUTOSAR configuration parameters above, the driver initialization lists `EcuM-DriverInitListOne` and `EcuMDriverRestartList` support post-build selectable configurations.

For information on how to migrate the EcuM module from ACG-8.5.2 to ACG-8.6.0, see [Section 4.3.2.1, “Migrating the EcuM from ACG-8.5.2 to ACG-8.6.0”](#).



## 4.7. Nm module user guide

### 4.7.1. Overview

This chapter provides you with `Nm`-specific information:

- ▶ [Section 4.7.2, “Background information”](#) explains the concepts of the `Nm` module.
- ▶ [Section 4.7.3, “Configuring the Nm module”](#) provides instructions on how to configure the `Nm` module.

### 4.7.2. Background information

In this chapter, you find information describing specific features of the `Nm` module. Background information about the network management concept and the `Nm` module is provided in the Concepts chapter of the EB tresos AutoCore Generic documentation. You find instructions relating to the configuration of `Nm` features in the parameter descriptions in the `Nm` module references chapter. You can also view the parameter descriptions in the **Description** view in EB tresos Studio.

#### 4.7.2.1. Network management state information

Network management state information (NMS) is a feature that gives you the possibility to send user data information regarding the state of the bus as part of an `Nm` PDU.

##### 4.7.2.1.1. Functional overview

The basic principle of this feature is to handle the `Nm` user data as a signal and to use existing mechanisms from the communication stack in order to ensure consistency of the user data. For this purpose, a `Com` signal is configured in the `Nm` module.

When you enable this feature the lower `<bus>Nm` calls a callback function `Nm_StateChangeNotification()`. Within this function, the user data is updated by calling `Com_SendSignal()` with the identifier of the related `Com` signal configured in `Nm`.

The data sent within this signal is described in the table below:

Value	Name	Description
1	NM_RM_BSM	NM in state RepeatMessage(transition from BusSleepMode)
2	NM_RM_PBSM	NM in state RepeatMessage(transition from PrepareBusSleepMode)



Value	Name	Description
4	NM_NO_RM	NM in state NormalOperation(transition from RepeatMessage)
8	NM_NO_RS	NM in state NormalOperation(transition from ReadySleep)
16	NM_RM_RS	NM in state RepeatMessage(transition from ReadySleep)
32	NM_RM_NO	NM in state RepeatMessage(transition from NormalOperation)

Table 4.2. NMS signal data

#### 4.7.2.1.2. Configuration options

You can use the network management state information (NMS) feature by enabling the configuration parameters `NmStateChangeIndEnabled` and `NmStateReportEnabled`. The parameter `NmStateReportSignalRef` references the related `Com` signal. See the parameter descriptions for further information.

#### 4.7.2.2. BSW distribution

The `Nm` provides mechanisms to call the bus-specific network management modules mapped to different cores with one bus type processed on one core. For the general concept for BSW distribution of the mode management for communication, see chapter *Multi-core support* in the EB tresos AutoCore Generic documentation.

##### 4.7.2.2.1. Functional overview

The `Nm` module must be mapped to the same core as the `ComM` master where the channel and partial network state machines are processed. All function calls between `ComM` and `Nm` are done on the local core via direct function calls.

When multi-core support is enabled, the `Nm` interacts with bus-specific network management modules (`BusNms`) cross-core by using `SchM_Call()` operations.

The `Nm` module provides a set of synchronous APIs that are meant to be used for OEM-specific extensions. If you want to use these APIs in your project, you should always use the corresponding APIs provided by `BusNm` modules on the local core. The goal is to avoid cross-core synchronous interfaces because they could cause blocking.

The `Nm` BSW distribution support is enabled with configuration parameter `NmMultiCoreSupport`. If this parameter is enabled, the generated basic software module description includes all required and provided client-server entries. These entries are generated based on the number of configured channels. When the multi-core support is enabled in the `Nm`, you must also enable the multi-core support for the `BusNms` modules that are used in your project. This allows you to connect the required client-server connections in the `Rte`.



If the calling context of a `SchM_Call()` is unknown, a dummy context is used. You must ensure that the dummy context and relevant events are mapped to tasks with correctly assigned priorities. For example, in the case of `Nm`, the functions are normally called from the context of the `ComM` main processing function. In this case, you must make sure that the dummy context used has the same priority as the `ComM` context. For the cases where a function is executed from the context of an interrupt, you must make sure that the dummy context used has the highest priority in the system, similar to an interrupt.

## 4.7.3. Configuring the Nm module

### 4.7.3.1. Configuring Nm for BSW distribution

In this section, you configure the `Nm` for BSW distribution. For more information on the `Nm` BSW distribution, see [Section 4.7.2.2, “BSW distribution”](#).



#### Configuring Nm for BSW distribution

Prerequisite:

- All `BusNm`s modules that are used in the project have multi-core support enabled.
- In the `Rte`, the following is configured:
  - ▶ The module instances, i.e. the `BswImplementation`s, are properly mapped to the cores via `OsApplications`.
  - ▶ Events are properly mapped to tasks.
  - ▶ Provided and required client-server connections are properly connected.
  - ▶ Provided and required sender-receiver connections are properly connected.

#### Step 1

To enable the `Nm` BSW distribution support, enable the configuration parameter `NmMultiCoreSupport`.

#### Step 2

For each asynchronous interface between `Nm` and `BusNm`, connect the provided and required client-server entries for each channel. Based on this information, the `Rte` generates the `SchM_Call()` operations that are used for inter-core communication.



# 5. ACG8 Mode Management module references

## 5.1. Overview

This chapter provides module references for the ACG8 Mode Management product modules. These include a detailed description of all configuration parameters. Furthermore this chapter lists the application programming interface with all data types, constants and functions.

The content of the sections is sorted alphabetically according the EB tresos AutoCore Generic module names.

For further information on the functional behavior of these modules, refer to the chapter ACG8 Mode Management user's guide.

### 5.1.1. Notation in EB module references

EB notation may differ from the AUTOSAR standard notation in the software specification documents (SWS). This section describes the notation of *default value* and *range* fields in the EB module references.

#### 5.1.1.1. Default value of configuration parameters

If there is no default value specified for a parameter, the default value field is omitted to prevent ambiguity with parameters that have -- as default values.

Example: The parameter `BswMCompuConstText` of the `BswM` module of EB tresos AutoCore Generic 8 Mode Management has no default value field, therefore it is omitted.

#### 5.1.1.2. Range information of configuration parameters

The range of a configuration parameter contains an upper and a lower boundary. However, in special cases the range of allowed values can be computed by means of an XPath function that is evaluated at configuration time. An XPath function can either be a standard `xpath:<function>()` or a custom `cxpath:<function>()` function. The range of a configuration parameter may be computed based on other configuration parameters that are referenced from the XPath function. For more information on custom XPath functions, see section *Custom XPath Functions API* of the EB tresos Studio developer's guide.



Example: The parameter `BswMCompuConstText` of the `BswM` module of EB tresos AutoCore Generic 8 Mode Management has the custom XPath function `cxpath:getCompuMethodsVT()` in the range field which provides the allowed values.

## 5.2. BswM

### 5.2.1. Configuration parameters

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">BswMConfig</a>	1..n	This container contains the configuration parameters and sub containers of the AUTOSAR BswM module.
<a href="#">BswMGeneral</a>	1..1	General configuration parameters of the Basic SW Mode Manager.
<a href="#">ReportToDem</a>	1..1	<b>Label:</b> Production error handling Production error handling
<a href="#">BswMDefensiveProgramming</a>	1..1	<b>Label:</b> Defensive Programming Options  Parameters for defensive programming
<a href="#">CommonPublishedInformation</a>	1..1	<b>Label:</b> Common Published Information  Common container, aggregated by all modules. It contains published information about vendor and versions.
<a href="#">PublishedInformation</a>	1..1	<b>Label:</b> EB Published Information  Additional published parameters not covered by Common-PublishedInformation container.

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">IMPLEMENTATION_CONFIG_VARIANT</a>	1..1
<a href="#">POST_BUILD VARIANT USED</a>	1..1

<b>Parameter Name</b>	<b>IMPLEMENTATION_CONFIG_VARIANT</b>
<b>Label</b>	Config Variant



<b>Description</b>	Specifies the variant of the BswM used in the current project. If Ecum module from EB is used it is mandatory to have the PbcfgM module present in the project too.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	VariantPreCompile	
<b>Range</b>	VariantPreCompile VariantPostBuild	
<b>Configuration class</b>	<b>VariantPreCompile:</b> <b>VariantPostBuild:</b>	VariantPreCompile VariantPostBuild

<b>Parameter Name</b>	<b>POST_BUILD_VARIANT_USED</b>	
<b>Label</b>	Post Build Variant Used	
<b>Description</b>	This parameter can be used to bypass the postbuild selectable variant awareness of BswM. If set to FALSE, BswM will not be variant aware (default behaviour). If set to TRUE, BswM will be variant aware and will produce variant dependent artifacts (ie. different .xgen and PBcfg files for each variant). This parameter is useful for the situations where postbuild selectable variants exist in the project but the BswM doesn't have any variant specific configuration.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b> <b>VariantPostBuild:</b>	VariantPreCompile VariantPostBuild

### 5.2.1.1. BswMConfig

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">BswMArbitration</a>	1..1	<p>This container includes all configuration sub-containers and parameters related to the mode arbitration functionality of the BswM.</p> <p>The arbitration functionality detects changes in the request-and indication-states of available sources and computes if, and if yes which actions the BswM shall perform as reaction.</p>

**Containers included**

<a href="#">BswMDataTypeMappingSets</a>	0..1	Collection of references to DataTypeMappingSet.
<a href="#">BswMModeControl</a>	1..1	This container includes all configuration sub-containers and parameters related to the mode control functionality of the BswM.

**Parameters included**

Parameter name	Multiplicity
<a href="#">BswMPartitionRef</a>	0..1

Parameter Name	<b>BswMPartitionRef</b>	
<b>Description</b>	This references the partition the BswM shall run inside. On a single-core ECU it is not required to configure this parameter. In case of a multi-core ECU it is mandatory to configure this parameter even if there is only one BswM instance.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
	<b>Link:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

**5.2.1.2. BswMArbitration****Containers included**

Container name	Multiplicity	Description
<a href="#">BswMLogicalExpression</a>	0..n	<p>This container describes the logical expressions that can be used for the mode arbitration.</p> <p>The logical expressions are built of a set of arguments and a logical operator.</p> <p>Each argument can either be a mode condition or a sub-expression to allow definition of more complex logical expressions.</p> <p>The AND and OR operator accept an unlimited number of arguments; NAND and XOR are not associative, so only two arguments are allowed.</p>



Containers included		
		Note that the order of evaluation of the expressions is not defined.
<a href="#">BswMModeCondition</a>	0..n	<p>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.</p> <p>A mode condition compares a mode request port state with some predefined value and results in a boolean value.</p>
<a href="#">BswMEventRequestPort</a>	0..n	<p>Each instance of this container defines an event which can be sent to the BswM. Basic Software Modules may send these events to the BswM by calling the corresponding BswM C-API (for example: BswM_ComM_InitiateReset()).</p>
<a href="#">BswMModeRequestPort</a>	0..n	<p>Each instance of this container defines a mode request interface that is used to requests or indicate modes from/to the BswM.</p> <p>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.</p> <p><i>There are different types of mode requests:</i></p> <ul style="list-style-type: none"> <li>▶ Mode requests from the SW-C:s</li> <li>▶ Mode Requests from other BSW modules such as the DCM.</li> <li>▶ State/mode indications from the RTE or other BSW modules such as the bus specific State Managers</li> </ul> <p><i>Note : BswM treats all request and indications in the exact same way.</i></p>
<a href="#">BswMRule</a>	0..n	<p>Each instance of this container describes a BswM arbitration rule.</p> <p>The rule either consists of a simple mode condition or a more complex logical expression which in turn calculates a boolean value from multiple mode conditions.</p> <p>This container also references the action lists that shall be invoked when the rule is evaluated to True or False.</p>



### 5.2.1.3. BswMLogicalExpression

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMLogicalOperator</a>	0..1
<a href="#">BswMArgumentRef</a>	1..n

<b>Parameter Name</b>	<b>BswMLogicalOperator</b>	
<b>Description</b>	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.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	ENUMERATION	
<b>Range</b>	BSWM_AND BSWM_NAND BSWM_OR BSWM_XOR	
<b>Configuration class</b>	<b>PreCompile:</b> <b>Link:</b>	VariantPreCompile VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMArgumentRef</b>	
<b>Description</b>	This is a choice reference either to a mode condition or a sub-expression.	
<b>Multiplicity</b>	1..n	
<b>Type</b>	CHOICE-REFERENCE	
<b>Range</b>	node:paths(as:modconf('BswM')[1]/BswMConfig/*/BswMArbitration/BswMModeCondition/*[asc_bswm:getBswMConfig(.) = asc_bswm:getBswMConfig(node:current())]) node:paths(as:modconf('BswM')[1]/BswMConfig/*/BswMArbitration/BswMLogicalExpression/*[(asc_bswm:getBswMConfig(.) = asc_bswm:getBswMConfig(node:current())) and (node:name(.) != node:name(node:current('/..../)))]	
<b>Configuration class</b>	<b>PreCompile:</b> <b>Link:</b>	VariantPreCompile VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	



### 5.2.1.4. BswMModeCondition

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">BswMConditionValue</a>	1..1	This container holds the parameters and references necessary to identify the mode type and the value that the mode request is compared to.

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMConditionType</a>	1..1
<a href="#">BswMConditionMode</a>	1..1

<b>Parameter Name</b>	<b>BswMConditionType</b>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Range</b>	asc_bswm:getConditionTypeRange(../BswMConditionMode)	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMConditionMode</b>	
<b>Description</b>	This parameter references the mode request port that is used for the condition.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	CHOICE-REFERENCE	
<b>Range</b>	node:paths(as:modconf('BswM')[1]/BswMConfig/*/BswMArbitration/*[node:name(.) = 'BswMEventRequestPort' or node:name(.) = 'BswMModeRequestPort']/*[asc_bswm:getBswMConfig(.) = asc_bswm:getBswMConfig(node:current())])	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	



### 5.2.1.5. BswMConditionValue

Containers included		
Container name	Multiplicity	Description
<a href="#">BswMBswMode</a>	1..1	This container defines the value and type of a mode in the BSW.
<a href="#">BswMCompuScaleModeValue</a>	1..1	This container contains parameters used to define a mode value.
<a href="#">BswMModeDeclaration</a>	1..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.

### 5.2.1.6. BswMBswMode

Parameters included	
Parameter name	Multiplicity
<a href="#">BswMBswRequestedMode</a>	1..1

Parameter Name	<b>BswMBswRequestedMode</b>	
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..1	
Type	STRING	
Configuration class	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.7. BswMCompuScaleModeValue

Parameters included	
Parameter name	Multiplicity
<a href="#">BswMCompuConstText</a>	1..1

**Parameters included**

<a href="#">BswMCompuMethodRef</a>	1..1
------------------------------------	------

Parameter Name	<b>BswMCompuConstText</b>	
<b>Description</b>	The value of this parameter shall match the VT member of a CompuConst defined within the referenced CompuMethod (BswMCompuMethodRef). The interval value of the corresponding CompuScale shall be used as the mode request value.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	STRING	
<b>Range</b>	asc_bswm:getCompuMethodsVT(..//BswMCompuMethodRef)	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH	

Parameter Name	<b>BswMCompuMethodRef</b>	
<b>Description</b>	This is a foreign reference to the CompuMethod used for mode requests. The CompuMethod category shall be TEXTTABLE.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FOREIGN-REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH	

**5.2.1.8. BswMModeDeclaration****Parameters included**

Parameter name	<b>Multiplicity</b>
<a href="#">BswMModeValueRef</a>	1..1

Parameter Name	<b>BswMModeValueRef</b>
<b>Description</b>	This is a foreign reference to the Mode Declaration used for the mode requests corresponding to this condition.
<b>Multiplicity</b>	1..1



Type	FOREIGN-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.9. BswMEventRequestPort

Containers included		
Container name	Multiplicity	Description
<a href="#">BswMEventRequestSource</a>	1..1	This choice container specifies the source of the event request. The sender of the event can be another BSW Module, such as ComM.

Parameters included	
Parameter name	Multiplicity
<a href="#">BswMEventRequestProcessing</a>	1..1

Parameter Name	<b>BswMEventRequestProcessing</b>	
Description	This parameter defines if the processing of the mode arbitration shall be done immediately when an event request is received or if it shall be deferred to the processing of the main function of BswM.	
Multiplicity	1..1	
Type	ENUMERATION	
Range	BSWM_DEFERRED BSWM_IMMEDIATE	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.10. BswMEventRequestSource

Containers included		
Container name	Multiplicity	Description


**Containers included**

<a href="#">BswMComMInitiateReset</a>	1..1	This is an indication from the ComM to signal a shutdown.
---------------------------------------	------	---

### 5.2.1.11. BswMComMInitiateReset

### 5.2.1.12. BswMModeRequestPort

**Containers included**

Container name	Multiplicity	Description
<a href="#">BswMModeInitValue</a>	0..1	This container defines the initial mode value that is used by BswM for the corresponding mode request after initialization. This container is optional and shall only be used for Mode Requests that do not already have an initial value.
<a href="#">BswMModeRequestSource</a>	1..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.

**Parameters included**

Parameter name	Multiplicity
<a href="#">BswMRequestProcessing</a>	1..1

Parameter Name	<b>BswMRequestProcessing</b>	
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..1	
Type	ENUMERATION	
Range	BSWM_DEFERRED BSWM_IMMEDIATE	
Configuration class	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECU	



### 5.2.1.13. BswMModelInitValue

Containers included		
Container name	Multiplicity	Description
<a href="#">BswMBswModelInitValue</a>	1..1	This is the choice for a initial mode value used for the initialization of mode requests in case the request is made by a BSW module.
<a href="#">BswMCompuScaleModeValue</a>	1..1	This container contains parameters used to define a mode value.
<a href="#">BswMSwcModelInitValue</a>	1..1	This is the choice for a foreign reference to the Mode Declaration used for the initialization of mode requests in case the request is made by a SW-C.

### 5.2.1.14. BswMBswModelInitValue

Parameters included	
Parameter name	Multiplicity
<a href="#">BswMBswModelInitValueMode</a>	1..1

Parameter Name	<b>BswMBswModelInitValueMode</b>	
Description	This parameter defines the initial mode value that is used by BswM for the corresponding mode request after initialization.	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.15. BswMCompuScaleModeValue

Parameters included	
Parameter name	Multiplicity
<a href="#">BswMCompuConstText</a>	1..1
<a href="#">BswMCompuMethodRef</a>	1..1



<b>Parameter Name</b>	<b>BswMCompuConstText</b>	
<b>Description</b>	The value of this parameter shall match the VT member of a CompuConst defined within the referenced CompuMethod (BswMCompuMethodRef). The interval value of the corresponding CompuScale shall be used as the mode request value.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	STRING	
<b>Range</b>	asc_bswm:getCompuMethodsVT(..//BswMCompuMethodRef)	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>BswMCompuMethodRef</b>	
<b>Description</b>	This is a foreign reference to the CompuMethod used for mode requests. The CompuMethod category shall be TEXTTABLE.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FOREIGN-REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.2.1.16. BswMSwcModelInitValue

<b>Parameters included</b>		
<b>Parameter name</b>	<b>Multiplicity</b>	
<a href="#">BswMSwcModelInitValueRef</a>	1..1	

<b>Parameter Name</b>	<b>BswMSwcModelInitValueRef</b>	
<b>Description</b>	This is a foreign reference to the Mode Declaration used for the initialization of mode requests.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FOREIGN-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild



<b>Origin</b>	AUTOSAR_ECUC
---------------	--------------

### 5.2.1.17. BswMMModeRequestSource

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">BswMBswModeNotification</a>	1..1	The source of the mode request is a Bsw Module.
<a href="#">BswMCanSMIndication</a>	1..1	This is an indication of the current state of the CAN State Manager.
<a href="#">BswMComMIndication</a>	1..1	This is an indication of the current communication mode of a channel in the Communication Manager.
<a href="#">BswMComMPncRequest</a>	1..1	This is a request of the current communication mode of a Partial Network Cluster in the Communication Manager.
<a href="#">BswMDcmApplicationUpdatedIndication</a>	1..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.
<a href="#">BswMDcmComModeCurrentState</a>	1..1	The source of the mode request is the Diagnostic Communication Manager.
<a href="#">BswMDcmComModeRequest</a>	1..1	The source of the mode request is the Diagnostic Communication Manager.
<a href="#">BswMDcmResetModeRequest</a>	1..1	This is a reset mode request from the DCM.  This container does not contain any parameters since there are no further configuration needed for this type of request.
<a href="#">BswMDcmSessionModeRequest</a>	1..1	This is a session mode request from the DCM.  This container does not contain any parameters since there are no further configuration needed for this type of request.
<a href="#">BswMEcuMIndication</a>	1..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.
<a href="#">BswMEcuMWakeupSource</a>	1..1	This is a notification of the current state of an ECU State Manager wakeup source.
<a href="#">BswMEthIfPortGroupLinkStateChg</a>	1..1	This is an indication from the EthIf if the link state of a Ethernet interface switch port group has changed.



<b>Containers included</b>		
<a href="#">BswMEthSMIndication</a>	1..1	This is an indication of the current state of the Ethernet State Manager.
<a href="#">BswMFrSMIndication</a>	1..1	This is an indication of the current state of the FlexRay State Manager.
<a href="#">BswMGenericRequest</a>	1..1	This mode request originates from a requester that is not among the list of standardized mode requesters (i.e. the different resource managers).
<a href="#">BswMJ1939DcmBroadcastStatus</a>	1..1	This is a notification of the desired broadcast status per network, triggered via DM13. Note that these ports can only have values 0 or 1 due to how the BswM_J1939DcmBroadcastStatus transmits information (via a single 16 bit parameter where each bit represent a ComM channel). This imposes further restrictions regarding mode conditions or initial values.
<a href="#">BswMJ1939NmIndication</a>	1..1	This is an indication of the current state of the J1939 network management module.
<a href="#">BswMLinScheduleIndication</a>	1..1	This is an indication of the currently active LIN Schedule Table for a specific LIN Interface.
<a href="#">BswMLinSMIndication</a>	1..1	This is an indication of the current state of the LIN State Manager.
<a href="#">BswMLinTpModeRequest</a>	1..1	This is a LinTp mode request from the LinIf. This port corresponds to a call of the BswM_LinTp_RequestMode API.
<a href="#">BswMNmlfCarWakeUpIndication</a>	1..1	This is an indication of a CarWakeUp from the Nmlf.
<a href="#">BswMNvMJobModeIndication</a>	1..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:
<a href="#">BswMNvMRequest</a>	1..1	This is a mode request for a NvM Block.
<a href="#">BswMRteSwitchAckNotification</a>	1..1	This is a notification from the RTE that a mode transition has been completed.
<a href="#">BswMSdClientServiceCurrentState</a>	1..1	Used by Service Discovery module to indicate current state of the Client Service (available/down).
<a href="#">BswMSdConsumedEventGroupCurrentState</a>	1..1	Used by Service Discovery to indicate current status of the EventHandler (requested/released).
<a href="#">BswMSdEventHandlerCurrentState</a>	1..1	Used by Service Discovery to indicate current status of the EventHandler (requested/released).

**Containers included**

<a href="#">BswMSwcModeNotification</a>	1..1	This is a mode switch notification associated with a RTE switch interface.
<a href="#">BswMSwcModeRequest</a>	1..1	The source of the mode request is a SW Component.
<a href="#">BswMTimer</a>	1..1	This is a timer which can be used for time dependent rules. In order to start/stop this timer, a BswMTimerControl action with BswMTimerAction = BSWM_TIMER_START/BSWM_TIMER_STOP must reference it. This mode request port can be used normally in mode conditions. Note that the initial value of this mode request port is BSWM_TIMER_STOPPED and is not configurable. Mode conditions referencing this mode request port shall only use BSWM_TIMER_STOPPED, BSWM_TIMER_STARTED or BSWM_TIMER_EXPIRED as mode values.
<a href="#">BswMWdgMRequestPartitionReset</a>	1..1	This is a Partition Reset request from the WdgM.  This port corresponds to a call of the <code>BswM_WdgM_RequestPartitionReset</code> API.

**5.2.1.18. BswMBswModeNotification****Parameters included**

Parameter name	Multiplicity
<a href="#">BswMBswModeDeclarationGroupPrototypeRef</a>	1..1
<b>Parameter Name</b>	<b>BswMBswModeDeclarationGroupPrototypeRef</b>
<b>Description</b>	This is a foreign reference to the Mode Declaration Group Prototype.
<b>Multiplicity</b>	1..1
<b>Type</b>	FOREIGN-REFERENCE
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile <b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

**5.2.1.19. BswMCanSMIndication****Parameters included**

Parameter name	Multiplicity

**Parameters included**

<a href="#">BswMCanSMChannelRef</a>	1..1
-------------------------------------	------

<b>Parameter Name</b>	<b>BswMCanSMChannelRef</b>	
<b>Description</b>	This is a reference to the CanSM channel handle that the mode request corresponds to.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

**5.2.1.20. BswMComMIndication**

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMComMChannelRef</a>	1..1

<b>Parameter Name</b>	<b>BswMComMChannelRef</b>	
<b>Description</b>	This is a reference to the Communication Manager channel handle that the indication corresponds to.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

**5.2.1.21. BswMComMPncRequest**

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMComMPncRef</a>	1..1
<b>Parameter Name</b>	<b>BswMComMPncRef</b>



<b>Description</b>	This is a reference to the Communication Manager PNC handle of the Partial Network Cluster that the request corresponds to.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.22. BswMDcmApplicationUpdatedIndication

### 5.2.1.23. BswMDcmComModeCurrentState

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMDcmComMChannelRef</a>	1..1

<b>Parameter Name</b>	<b>BswMDcmComMChannelRef</b>	
<b>Description</b>	This is a reference to the Communication Manager channel handle that the indication corresponds to.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.24. BswMDcmComModeRequest

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMDcmComMChannelRef</a>	1..1
<a href="#">BswMDcmComMNetwork</a>	1..1



<b>Parameter Name</b>	<b>BswMDcmComMChannelRef</b>	
<b>Description</b>	This is a reference to the Communication Manager channel handle that the indication corresponds to.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMDcmComMNetwork</b>	
<b>Description</b>	This configuration parameter is not used. The reference to the ComM channel shall be defined within the parameter BswMDcmComMChannelRef.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	STRING	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.25. BswMDcmResetModeRequest

### 5.2.1.26. BswMDcmSessionModeRequest

### 5.2.1.27. BswMEcuMIndication

### 5.2.1.28. BswMEcuMWakeupSource

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>

**Parameters included**

<a href="#">BswMEcuMWakeupSrcRef</a>	1..1
--------------------------------------	------

<b>Parameter Name</b>	<b>BswMEcuMWakeupSrcRef</b>	
<b>Description</b>	This is a reference to the ECU State Manager Wakeup Source that the indication corresponds to.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

**5.2.1.29. BswMEthIfPortGroupLinkStateChg****Parameters included**

Parameter name	Multiplicity
<a href="#">BswMEthIfSwitchPortGroupRef</a>	1..1

<b>Parameter Name</b>	<b>BswMEthIfSwitchPortGroupRef</b>	
<b>Description</b>	This is a reference to the Ethernet Interface Switch Port Group that the indication corresponds to.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

**5.2.1.30. BswMEthSMIndication****Parameters included**

Parameter name	Multiplicity
<a href="#">BswMEthSMChannelRef</a>	1..1



<b>Parameter Name</b>	<b>BswMEthSMChannelRef</b>	
<b>Description</b>	This is a reference to the ComM channel respectively the EthSM channel that the mode request corresponds to.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.31. BswMFrSMIndication

<b>Parameters included</b>		
<b>Parameter name</b>	<b>Multiplicity</b>	
<a href="#">BswMFrSMChannelRef</a>	1..1	

<b>Parameter Name</b>	<b>BswMFrSMChannelRef</b>	
<b>Description</b>	This is a reference to the FlexRay Cluster handle that the mode request corresponds to.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.32. BswMGenericRequest

<b>Parameters included</b>		
<b>Parameter name</b>	<b>Multiplicity</b>	
<a href="#">BswMModeRequesterId</a>	1..1	
<a href="#">BswMRequestedModeMax</a>	1..1	

<b>Parameter Name</b>	<b>BswMModeRequesterId</b>
-----------------------	----------------------------



<b>Description</b>	This parameters identifies the different users of the generic mode request interface.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMRequestedModeMax</b>	
<b>Description</b>	This parameter is not used in the current BswM implementation. The upper limit for the requested modes is 65535. The BswM_ModeType is statically defined as uint16.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.33. BswMJ1939DcmBroadcastStatus

<b>Parameters included</b>		
<b>Parameter name</b>	<b>Multiplicity</b>	
<a href="#">BswMJ1939DcmChannelRef</a>	1..1	

<b>Parameter Name</b>	<b>BswMJ1939DcmChannelRef</b>	
<b>Description</b>	Reference to the communication channel which is affected by this mode request.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	



### 5.2.1.34. BswMJ1939NmIndication

Parameters included	
Parameter name	Multiplicity
<a href="#">BswMJ1939NmChannelRef</a>	1..1
<a href="#">BswMJ1939NmNodeRef</a>	1..1

Parameter Name	<b>BswMJ1939NmChannelRef</b>	
Description	This is a reference to the J1939Nm channel handle that the mode request corresponds to.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	<b>VariantPreCompile:</b> <b>VariantPostBuild:</b>	VariantPreCompile VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>BswMJ1939NmNodeRef</b>	
Description	This is a reference to the node that the mode request corresponds to.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	<b>VariantPreCompile:</b> <b>VariantPostBuild:</b>	VariantPreCompile VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.35. BswMLinScheduleIndication

Parameters included	
Parameter name	Multiplicity
<a href="#">BswMLinScheduleRef</a>	1..1

Parameter Name	<b>BswMLinScheduleRef</b>
Description	This is a reference to the LIN Schedule Table handle that the mode request corresponds to.
Multiplicity	1..1



Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.36. BswMLinSMIndication

Parameters included	
Parameter name	Multiplicity
<a href="#">BswMLinSMChannelRef</a>	1..1

Parameter Name	<b>BswMLinSMChannelRef</b>	
Description	This is a reference to the LIN channel handle that the mode request corresponds to.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.37. BswMLinTpModeRequest

Parameters included	
Parameter name	Multiplicity
<a href="#">BswMLinTpChannelRef</a>	1..1

Parameter Name	<b>BswMLinTpChannelRef</b>	
Description	This is a reference to the LIN Interface Channel that the mode request corresponds to.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile



	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.38. BswMNmIfCarWakeUpIndication

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMNmChannelRef</a>	1..1

<b>Parameter Name</b>	<b>BswMNmChannelRef</b>	
<b>Description</b>	This is a reference to the channel handle that the indication corresponds to.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.39. BswMNvMJobModelIndication

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMNvmService</a>	1..1

<b>Parameter Name</b>	<b>BswMNvmService</b>	
<b>Description</b>	Identifies the Nvm job which is related to the mode request.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Range</b>	NvmReadAll	
	NvmWriteAll	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild



<b>Origin</b>	AUTOSAR_ECUC
---------------	--------------

### 5.2.1.40. BswMNvMRequest

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMNvMBlockRef</a>	1..1

<b>Parameter Name</b>	<b>BswMNvMBlockRef</b>	
<b>Description</b>	This is a reference to the NvM Block Descriptor that the request corresponds to.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.41. BswMRteSwitchAckNotification

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMSwitchPortRef</a>	1..1

<b>Parameter Name</b>	<b>BswMSwitchPortRef</b>	
<b>Description</b>	References the switch port which will receive the notification.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Range</b>	node:paths(as:modconf('BswM')[1]/BswMConfig/*/BswMMODE- Control/BswMSwitchPort/*[asc_bswm:getBswMConfig(.) = asc_- bswm:getBswMConfig(node:current())])	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	



#### 5.2.1.42. BswMSdClientServiceCurrentState

Parameters included	
Parameter name	Multiplicity
<a href="#">BswMSdClientMethodsRef</a>	1..1

Parameter Name	<b>BswMSdClientMethodsRef</b>	
Description	This is a reference to a client service in the Sd module.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

#### 5.2.1.43. BswMSdConsumedEventGroupCurrentState

Parameters included	
Parameter name	Multiplicity
<a href="#">BswMSdConsumedEventGroupRef</a>	1..1

Parameter Name	<b>BswMSdConsumedEventGroupRef</b>	
Description	This is a reference to an eventGroup that is defined within a client service in the Sd module.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

#### 5.2.1.44. BswMSdEventHandlerCurrentState

Parameters included	
Parameter name	Multiplicity

**Parameters included**

<a href="#">BswMSdEventHandlerRef</a>	1..1
---------------------------------------	------

<b>Parameter Name</b>	<b>BswMSdEventHandlerRef</b>	
<b>Description</b>	This is a reference to an event handler that is defined within a server service in the Sd module.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

**5.2.1.45. BswMSwcModeNotification****Parameters included**

<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMSwcModeNotificationModeDeclarationGroupPrototypeRef</a>	1..1

<b>Parameter Name</b>	<b>BswMSwcModeNotificationModeDeclarationGroupPrototypeRef</b>	
<b>Description</b>	This is a foreign reference to the ModeDeclarationGroupPrototype.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FOREIGN-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

**5.2.1.46. BswMSwcModeRequest****Parameters included**

<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMSwcModeRequestModeDeclarationGroupPrototypeRef</a>	0..1
<a href="#">BswMSwcModeRequestVariableDataPrototypeRef</a>	0..1

<b>Parameter Name</b>	<b>BswMSwcModeRequestModeDeclarationGroupPrototypeRef</b>
-----------------------	---



<b>Description</b>	This is a foreign reference to the ModeDeclarationGroupPrototype. As the mode request is SR-Communication the BswM shall provide a SR-Interface which corresponds to the ModeDeclarationGroupPrototype.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FOREIGN-REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
	<b>Link:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMSwcModeRequestVariableDataPrototypeRef</b>	
<b>Description</b>	This is a foreign reference to the VariableDataPrototype. It's the SWC's responsibility to define a SR-Interface.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FOREIGN-REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
	<b>Link:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.47. BswMTimer

### 5.2.1.48. BswMWdgMRequestPartitionReset

<b>Parameters included</b>		
<b>Parameter name</b>	<b>Multiplicity</b>	
<a href="#">BswMWdgMOsApplicationRef</a>	0..1	
<a href="#">BswMWdgMRequestPartitionResetRef</a>	1..1	

<b>Parameter Name</b>	<b>BswMWdgMOsApplicationRef</b>
<b>Description</b>	This is a reference to the Os application that requests the reset. If this parameter is not enabled (eg. the default application is used), then the BswM will react to a call of BswM_WdgM_RequestPartitionReset only if the application id is 0 (default application).
<b>Multiplicity</b>	0..1



Type	REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
	<b>Link:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>BswMWdgMRequestPartitionResetRef</b>	
<b>Description</b>	This is a reference to the partition that shall be reset. This parameter is not used. Instead, the BswMWdgMOsApplicationRef parameter can be used to specify the application for which the reset shall be done.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.49. BswMRule

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMNestedExecutionOnly</a>	1..1
<a href="#">BswMRuleInitState</a>	1..1
<a href="#">BswMRuleExpressionRef</a>	1..1
<a href="#">BswMRuleFalseActionList</a>	0..1
<a href="#">BswMRuleTrueActionList</a>	0..1

<b>Parameter Name</b>	<b>BswMNestedExecutionOnly</b>
<b>Description</b>	<p>This parameter determines if the current rule is independent or subordinate.</p> <p>If BswMNestedExecutionOnly = FALSE (default), the current rule is considered independent and it will be evaluated either at the BswM_MainFunction (if its logical expressions reference deferred mode request ports) or as soon as the state of one of the referenced mode request ports changes (in case they are immediate).</p> <p>If BswMNestedExecutionOnly = TRUE, the current rule is considered subordinate and it will only evaluated as part of an action list execution.</p>



<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

Parameter Name	<b>BswMRuleInitState</b>	
<b>Description</b>	<p>This parameter is a part of the reset/initialization behavior of BswM.</p> <p>Action lists are executed when the result of a rule evaluation have changed since the last evaluation.</p> <p>This parameter defines the previous evaluation result of a rule to be used after initialization of the BswM.</p> <p>If this parameter is set to BSWM_TRUE, the evaluation result is treated as changed if the rule is evaluated to false.</p> <p>If this parameter is set to BSWM_FALSE, the evaluation result is treated as changed if the rule is evaluated to true.</p> <p>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.</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Range</b>	BSWM_FALSE BSWM_TRUE BSWM_UNDEFINED	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

Parameter Name	<b>BswMRuleExpressionRef</b>	
<b>Description</b>	This is a reference to the logical expression that is evaluated for each rule.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	



<b>Range</b>	node:paths(as:modconf('BswM')[1]/BswMConfig/*/BswMArbitration/BswMLogicalExpression/*[asc_bswm:getBswMConfig(.) = asc_bswm:getBswMConfig(node:current())])	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	
<b>Parameter Name</b>	<b>BswMRuleFalseActionList</b>	
<b>Description</b>	This is a reference to the action list that shall be executed when the rule is evaluated to <code>False</code> .	
<b>Multiplicity</b>	0..1	
<b>Type</b>	REFERENCE	
<b>Range</b>	node:paths(as:modconf('BswM')[1]/BswMConfig/*/BswMMODEControl/BswMActionList/*[asc_bswm:getBswMConfig(.) = asc_bswm:getBswMConfig(node:current())])	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	
<b>Parameter Name</b>	<b>BswMRuleTrueActionList</b>	
<b>Description</b>	This is a reference to the action list that shall be executed when the rule is evaluated to <code>True</code> .	
<b>Multiplicity</b>	0..1	
<b>Type</b>	REFERENCE	
<b>Range</b>	node:paths(as:modconf('BswM')[1]/BswMConfig/*/BswMMODEControl/BswMActionList/*[asc_bswm:getBswMConfig(.) = asc_bswm:getBswMConfig(node:current())])	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.50. BswMDataTypeMappingSets

Parameters included	
Parameter name	Multiplicity

**Parameters included**

<a href="#">BswMDataTypeMappingSetRef</a>	1..n
---	------

<b>Parameter Name</b>	<b>BswMDataTypeMappingSetRef</b>	
<b>Description</b>	Reference to DataTypeMappingSet.	
<b>Multiplicity</b>	1..n	
<b>Type</b>	FOREIGN-REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
	<b>Link:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

**5.2.1.51. BswMModeControl**

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">BswMAction</a>	0..n	Each container of this type defines an action.  These actions can be part of one or several action lists.
<a href="#">BswMActionList</a>	0..n	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.
<a href="#">BswMRteModeRequestPort</a>	0..n	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.
<a href="#">BswMSwitchPort</a>	0..n	This container specifies PPorts and/or providedModeDeclarationGroups , which the BswM has to create in its SWCD resp. BSWMD.

**5.2.1.52. BswMAction**

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>



#### Containers included

<a href="#">BswMAvailableActions</a>	1..1	Choice container including the available actions to be used in the action lists.
--------------------------------------	------	--

#### 5.2.1.53. BswMAvailableActions

##### Containers included

Container name	Multiplicity	Description
<a href="#">BswMClearEventRequest</a>	1..1	This container contains a reference to a BswMEventRequestPort which will be cleared (i.e. set to CLEAR state) when this action is executed.
<a href="#">BswMComMAllowCom</a>	1..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.
<a href="#">BswMComMModeLimitation</a>	1..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.
<a href="#">BswMComMModeSwitch</a>	1..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.
<a href="#">BswMDeadlineMonitoring-Control</a>	1..1	This container includes all parameters related to enabling and disabling of deadline monitoring for one or several PDUs in COM.
<a href="#">BswMEcuMGoDown</a>	1..1	This container defines the UserId which shall be forwarded to the GoDown request.
<a href="#">BswMEcuMSelectShutdown-Target</a>	1..1	This container defines the shutdown target.
<a href="#">BswMEthIfSwitchPort-GroupRequestMethod</a>	1..1	This container includes all parameters related to requesting a mode for the EthIfSwtPortGroup. The EthIf_SwitchPort-GroupRequestMethod API is called when this action is executed.
<a href="#">BswMJ1939DcmStateSwitch</a>	1..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.



<b>Containers included</b>		
<a href="#">BswMJ1939RmStateSwitch</a>	1..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.
<a href="#">BswMLinScheduleSwitch</a>	1..1	This container includes all parameters related to a switch of LIN schedule table. LinSM_ScheduleRequest is called when this action is configured.
<a href="#">BswMNMControl</a>	1..1	This container includes all parameters related to enabling and disabling of Network Management communication.  Disabling of NM communication can be requested by DCM.
<a href="#">BswMPduGroupSwitch</a>	1..1	This container includes references to the PDU groups that shall be enabled and disabled. Com_IpduGroupControl is called when this action is configured.
<a href="#">BswMPduRouterControl</a>	1..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.
<a href="#">BswMRteModeRequest</a>	1..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.
<a href="#">BswMRteSwitch</a>	1..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.
<a href="#">BswMSchMSwitch</a>	1..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.
<a href="#">BswMSdClientServiceModeRequest</a>	1..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.
<a href="#">BswMSdConsumedEventGroupModeRequest</a>	1..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.
<a href="#">BswMSdServerServiceModeRequest</a>	1..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.

**Containers included**

<a href="#">BswMSwitchIPduMode</a>	1..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.
<a href="#">BswMTimerControl</a>	1..1	This container includes all parameters for the action to start or to stop a timer.
<a href="#">BswMTriggerIPduSend</a>	1..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.
<a href="#">BswMUserCallout</a>	1..1	This container includes all details needed for a user defined function call.

**5.2.1.54. BswMClearEventRequest****Parameters included**

Parameter name	Multiplicity
<a href="#">BswMClearEventRequestPortRef</a>	1..1

<b>Parameter Name</b>	<b>BswMClearEventRequestPortRef</b>	
<b>Description</b>	This parameter references the BswMEventRequestPort which will be cleared.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Range</b>	node:paths(as:modconf('BswM')[1]/BswMConfig/*/BswMArbitration/BswMEventRequestPort/*[asc_bswm:getBswMConfig(.) = asc_bswm:getBswMConfig(node:current())])	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

**5.2.1.55. BswMComMAllowCom****Parameters included**

Parameter name	Multiplicity
<a href="#">BswMComAllowed</a>	1..1

**Parameters included**[BswMComMAllowChannelRef](#)

1..1

<b>Parameter Name</b>	<b>BswMComAllowed</b>	
<b>Description</b>	The parameter BswMComMAllowChannelRef refers to a channel which will allow or block communication using the function ComM_CommunicationAllowed()	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

**Parameter Name****BswMComMAllowChannelRef**

This is a reference to the ComM Channel for which communication shall be allowed or blocked.

**Multiplicity**

1..1

**Type**

SYMBOLIC-NAME-REFERENCE

**Configuration class****VariantPreCompile:** VariantPreCompile**VariantPostBuild:** VariantPostBuild**Origin**

AUTOSAR\_ECUC

**5.2.1.56. BswMComMModeLimitation****Parameters included****Parameter name**[BswMComMLimitMode](#)**Multiplicity**

1..1

[BswMComMLimitChannelRef](#)

1..1

**Parameter Name****BswMComMLimitMode****Description**

This parameter specifies if the channel referenced by the BswMComMLimitChannelRef shall be limited to NoCom (true) or if the limitation shall be lifted (false).

**Multiplicity**

1..1

**Type**

BOOLEAN



<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMComMLimitChannelRef</b>	
<b>Description</b>	This is a reference to the ComM channel for which the communication mode should be limited.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.57. BswMComMMModeSwitch

<b>Parameters included</b>		
<b>Parameter name</b>	<b>Multiplicity</b>	
<a href="#">BswMComMRequestedMode</a>	1..1	
<a href="#">BswMComMUserRef</a>	1..1	

<b>Parameter Name</b>	<b>BswMComMRequestedMode</b>	
<b>Description</b>	This parameter specifies if the requested communication mode.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Range</b>	BSWM_FULL_COM BSWM_NO_COM	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMComMUserRef</b>	
<b>Description</b>	This is a reference to the ComM User that is associated to the Communication channel for which the communication mode should be requested.	
<b>Multiplicity</b>	1..1	



Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.58. BswMDeadlineMonitoringControl

Parameters included		
Parameter name	Multiplicity	
<a href="#">BswMDisabledDMPduGroupRef</a>	0..n	
<a href="#">BswMEnabledDMPduGroupRef</a>	0..n	

Parameter Name	<b>BswMDisabledDMPduGroupRef</b>	
Description	This is a reference to a PDU Group for which the Deadline Monitoring should be disabled.	
Multiplicity	0..n	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>BswMEnabledDMPduGroupRef</b>	
Description	This is a reference to a PDU Group for which the Deadline Monitoring should be enabled.	
Multiplicity	0..n	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.59. BswMEcuMGoDown

Parameters included		
Parameter name	Multiplicity	

**Parameters included**

<a href="#">BswMEcuMUserIdRef</a>	1..1
-----------------------------------	------

<b>Parameter Name</b>	<b>BswMEcuMUserIdRef</b>	
<b>Description</b>	This is a reference to a EcUM UserId.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

**5.2.1.60. BswMEcuMSelectShutdownTarget**

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMEcuMShutdownTarget</a>	1..1
<a href="#">BswMEcuMResetModeRef</a>	0..1
<a href="#">BswMEcuMShutdownTargetRef</a>	1..1
<a href="#">BswMEcuMSleepModeRef</a>	0..1

<b>Parameter Name</b>	<b>BswMEcuMShutdownTarget</b>	
<b>Description</b>	Selects the EcUM shutdown target type.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Range</b>	ECUM_STATE_SLEEP	
	ECUM_STATE_RESET	
	ECUM_STATE_OFF	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMEcuMResetModeRef</b>	
<b>Description</b>	This is a reference to a reset mode.	
<b>Multiplicity</b>	0..1	



Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
	Link:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>BswMEcuMShutdownTargetRef</b>	
Description	This configuration parameter is not used. The configuration parameter BswMEcuMShutdownTargetRef has been renamed to BswMEcuMSleepModeRef and the parameter BswMEcuMResetModeRef has been introduced. Rationale: A BswMEcuMShutdownTarget may reference a SleepMode or a ResetMode.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>BswMEcuMSleepModeRef</b>	
Description	This is a reference to a sleep mode.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
	Link:	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.61. BswMEthIfSwitchPortGroupRequestMode

Parameters included		
Parameter name		Multiplicity
<a href="#">BswMEthTrcvMode</a>		1..1
<a href="#">BswMEthIfSwitchPortGroupRef</a>		1..1

Parameter Name	<b>BswMEthTrcvMode</b>
Description	This parameter contains the mode which will be requested.
Multiplicity	1..1



Type	ENUMERATION	
Range	BSWM_ETHTRCV_MODE_ACTIVE BSWM_ETHTRCV_MODE_DOWN	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>BswMEthIfSwitchPortGroupRef</b>	
Description	This is a reference to the Ethernet Interface Switch Port Group which will receive the request.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.62. BswMJ1939DcmStateSwitch

Parameters included		
Parameter name	Multiplicity	
<a href="#">BswMJ1939DcmRequestedState</a>	1..1	
<a href="#">BswMJ1939DcmChannelRef</a>	1..1	
<a href="#">BswMJ1939DcmNodeRef</a>	1..1	

Parameter Name	<b>BswMJ1939DcmRequestedState</b>	
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..1	
Type	ENUMERATION	
Range	BSWM_J1939DCM_STATE_OFFLINE BSWM_J1939DCM_STATE_ONLINE	
Configuration class	VariantPreCompile:	VariantPreCompile



	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMJ1939DcmChannelRef</b>	
<b>Description</b>	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.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMJ1939DcmNodeRef</b>	
<b>Description</b>	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.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.63. BswMJ1939RmStateSwitch

<b>Parameters included</b>		
<b>Parameter name</b>	<b>Multiplicity</b>	
<a href="#">BswMJ1939RmRequestedState</a>	1..1	
<a href="#">BswMJ1939RmChannelRef</a>	1..1	
<a href="#">BswMJ1939RmNodeRef</a>	1..1	
<b>Parameter Name</b>	<b>BswMJ1939RmRequestedState</b>	
<b>Description</b>	This parameter describes the communication state of the J1939 Request Manager and corresponds to the parameter "new state" of the function J1939Rm_SetState.	



<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Range</b>	BSWM_J1939RM_STATE_OFFLINE BSWM_J1939RM_STATE_ONLINE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMJ1939RmChannelRef</b>	
<b>Description</b>	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.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMJ1939RmNodeRef</b>	
<b>Description</b>	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.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.64. BswMLinScheduleSwitch

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMLinScheduleRef</a>	1..1



<b>Parameter Name</b>	<b>BswMLinScheduleRef</b>	
<b>Description</b>	This is a reference to the LIN schedule table that the LIN SM shall change to.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.65. BswMNMControl

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMNMAction</a>	1..1
<a href="#">BswMComMNetworkHandleRef</a>	1..1

<b>Parameter Name</b>	<b>BswMNMAction</b>	
<b>Description</b>	This parameter specifies if the communication of the corresponding NM channel should be enabled or disabled.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Range</b>	BSWM_NM_DISABLE	
	BSWM_NM_ENABLE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMComMNetworkHandleRef</b>	
<b>Description</b>	This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild



<b>Origin</b>	AUTOSAR_ECUC
---------------	--------------

### 5.2.1.66. BswMPduGroupSwitch

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMPduGroupSwitchReinit</a>	1..1
<a href="#">BswMDisabledPduGroupRef</a>	0..n
<a href="#">BswMEnabledPduGroupRef</a>	0..n

<b>Parameter Name</b>	<b>BswMPduGroupSwitchReinit</b>	
<b>Description</b>	This parameter defines if the values of timers, memorized last values etc. are retained or reinitialized during a PDU Group Switch.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMDisabledPduGroupRef</b>	
<b>Description</b>	This is a reference to a PDU Group that should be disabled.	
<b>Multiplicity</b>	0..n	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMEnabledPduGroupRef</b>	
<b>Description</b>	This is a reference to a PDU Group that should be enabled.	
<b>Multiplicity</b>	0..n	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
	<b>PostBuild:</b>	VariantPostBuild



<b>Origin</b>	AUTOSAR_ECUC
---------------	--------------

### 5.2.1.67. BswMPduRouterControl

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswPduRouterAction</a>	1..1
<a href="#">BswMPduRoutingPathGroupRef</a>	1..n

<b>Parameter Name</b>	<b>BswPduRouterAction</b>	
<b>Description</b>	This parameter specifies if the routing of the corresponding PDU should be enabled or disabled.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Range</b>	BSWM_PDUR_DISABLE BSWM_PDUR_ENABLE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMPduRoutingPathGroupRef</b>	
<b>Description</b>	This is a reference to the PDU Routing Path Group for which the routing in the PDU Router should be enabled or disabled.	
<b>Multiplicity</b>	1..n	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.68. BswMRteModeRequest

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>

**Parameters included**

<a href="#">BswMRequestedModeRef</a>	1..1
<a href="#">BswMRteModeRequestPortRef</a>	1..1

<b>Parameter Name</b>	<b>BswMRequestedModeRef</b>	
<b>Description</b>	This is a foreign reference to the Mode Declaration used for the mode request	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FOREIGN-REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMRteModeRequestPortRef</b>	
<b>Description</b>	This is a reference to a BswMRteModeRequestPort.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Range</b>	node:paths(as:modconf('BswM')[1]/BswMConfig/*/BswMMModeControl/BswMRteModeRequestPort/*[asc_bswm:getBswMConfig(.) = asc_bswm:getBswMConfig(node:current())])	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

**5.2.1.69. BswMRteSwitch****Parameters included**

<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMSwitchedMode</a>	1..1
<a href="#">BswMRteSwitchPortRef</a>	1..1

**Parameter Name**

<b>Parameter Name</b>	<b>BswMSwitchedMode</b>
<b>Description</b>	This parameter references a Mode Declaration of a Mode Declaration Group.
<b>Multiplicity</b>	1..1



<b>Type</b>	FOREIGN-REFERENCE	
<b>Range</b>	asc_bswm:getCompatibleModeDeclarations(..../BswMRteSwitchPortRef)	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMRteSwitchPortRef</b>	
<b>Description</b>	This is a reference to the BswMSwitchPort.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Range</b>	node:paths(as:modconf('BswM')[1]/BswMConfig/*/BswMMODE- Control/BswMSwitchPort/*[asc_bswm:getBswMConfig(.) = asc_- bswm:getBswMConfig(node:current())])	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.70. BswMSchMSwitch

Parameters included		
Parameter name		Multiplicity
<a href="#">BswMSchMSwitchedMode</a>		1..1
<a href="#">BswMSchMSwitchPortRef</a>		1..1

<b>Parameter Name</b>	<b>BswMSchMSwitchedMode</b>	
<b>Description</b>	This parameter contains the integer value that corresponds to a certain mode in a Mode Declaration Group.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FOREIGN-REFERENCE	
<b>Range</b>	asc_bswm:getCompatibleModeDeclarations(..../BswMSchMSwitchPortRef)	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild



<b>Origin</b>	AUTOSAR_ECUC
---------------	--------------

<b>Parameter Name</b>	<b>BswMSchMSwitchPortRef</b>	
<b>Description</b>	This is a reference to the BswMSwitchPort.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Range</b>	node:paths(as:modconf('BswM')[1]/BswMConfig/*/BswMMODE- Control/BswMSwitchPort/*[asc_bswm:getBswMConfig(.) = asc_- bswm:getBswMConfig(node:current())])	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.71. BswMSdClientServiceModeRequest

<b>Parameters included</b>		
<b>Parameter name</b>		<b>Multiplicity</b>
<a href="#">BswMSdClientMethodsRef</a>		1..1
<a href="#">BswMSdClientServiceState</a>		1..1

<b>Parameter Name</b>	<b>BswMSdClientMethodsRef</b>	
<b>Description</b>	This is a reference to a client service in the Sd module.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMSdClientServiceState</b>	
<b>Description</b>	This parameter specifies if the corresponding client service shall be released or requested.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	



<b>Range</b>	BSWM_SD_CLIENT_SERVICE_RELEASED BSWM_SD_CLIENT_SERVICE_REQUESTED	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.2.1.72. BswMSdConsumedEventGroupModeRequest

Parameters included	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMSdConsumedEventGroupRef</a>	1..1
<a href="#">BswMSdConsumedEventGroupState</a>	1..1

<b>Parameter Name</b>	<b>BswMSdConsumedEventGroupRef</b>	
<b>Description</b>	This is a reference to an eventGroup that is defined within a client service in the Sd module.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMSdConsumedEventGroupState</b>	
<b>Description</b>	This parameter specifies if the corresponding consumed event group shall be released or requested.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Range</b>	BSWM_SD_CONSUMED_EVENTGROUP_RELEASED	
	BSWM_SD_CONSUMED_EVENTGROUP_REQUESTED	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	



### 5.2.1.73. BswMSdServerServiceModeRequest

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMSdServerMethodsRef</a>	1..1
<a href="#">BswMSdServerServiceState</a>	1..1

<b>Parameter Name</b>	<b>BswMSdServerMethodsRef</b>	
<b>Description</b>	This is a reference to a server service in the Sd module.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMSdServerServiceState</b>	
<b>Description</b>	This parameter specifies if the corresponding server service shall be down or available.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Range</b>	BSWM_SD_SERVER_SERVICE_DOWN BSWM_SD_SERVER_SERVICE_AVAILABLE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.2.1.74. BswMSwitchIPduMode

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMSwitchIPduModeValue</a>	1..1
<a href="#">BswMSwitchIPduModeRef</a>	1..1
<b>Parameter Name</b>	<b>BswMSwitchIPduModeValue</b>



<b>Description</b>	This parameter defines which transmission mode shall be selected during this call.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMSwitchIPduModeRef</b>	
<b>Description</b>	This is a reference to an I-PDU for which the transmission mode shall be set.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.75. BswMTimerControl

<b>Parameters included</b>		
<b>Parameter name</b>		<b>Multiplicity</b>
<a href="#">BswMTimerAction</a>		1..1
<a href="#">BswMTimerValue</a>		1..1
<a href="#">BswMTimerRef</a>		1..1

<b>Parameter Name</b>	<b>BswMTimerAction</b>	
<b>Description</b>	Specify the action for the timer. The timer can be started or stopped.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	BSWM_TIMER_START	
<b>Range</b>	BSWM_TIMER_START	
	BSWM_TIMER_STOP	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild



<b>Origin</b>	AUTOSAR_ECUC
---------------	--------------

<b>Parameter Name</b>	<b>BswMTimerValue</b>	
<b>Description</b>	Specify the timer value (in seconds) that is used when the timer is started. Note that this value must be a multiple of BswMMainFunctionPeriod.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
	<b>Link:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMTimerRef</b>	
<b>Description</b>	Specify the Timer for which the timer action shall be executed.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Range</b>	node:paths(as:modconf('BswM')[1]/BswMConfig/*[BswMArbitration/BswMModeRequestPort/*[asc_bswm:getBswMConfig(.) = asc_bswm:getBswMConfig(node:current()) and BswMModeRequestSource = 'BswMTimer'])	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.76. BswMTriggeredIPduSend

Parameters included		
Parameter name		Multiplicity
<a href="#">BswMTriggeredIPduRef</a>		1..n

<b>Parameter Name</b>	<b>BswMTriggeredIPduRef</b>	
<b>Description</b>	This is a reference to an I-PDU that should be triggered for transmission.	
<b>Multiplicity</b>	1..n	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile



	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

### 5.2.1.77. BswMUserCallout

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMUserCalloutFunction</a>	1..1

<b>Parameter Name</b>	<b>BswMUserCalloutFunction</b>	
<b>Description</b>	Specifies a complete function call (including parameters) the BswM will make when this action is executed. A semicolon for ending the statement is not needed.  Note that the return value of this function is completely ignored.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	STRING	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

### 5.2.1.78. BswMActionList

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">BswMActionListItem</a>	1..n	This container defines an item in an action list.

<b>Parameters included</b>		
<b>Parameter name</b>	<b>Multiplicity</b>	
<a href="#">BswMActionListExecution</a>		1..1

<b>Parameter Name</b>	<b>BswMActionListExecution</b>
<b>Description</b>	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



	<p>parameter does not have an effect when this action list is executed within another action list. Possible values:</p> <ul style="list-style-type: none"> <li>▶ BSWM_CONDITION: action list shall be executed every time the rule is evaluated</li> <li>▶ BSWM_TRIGGER: action list shall be executed every time the result of the evaluation changes</li> </ul>				
<b>Multiplicity</b>	1..1				
<b>Type</b>	ENUMERATION				
<b>Range</b>	BSWM_CONDITION BSWM_TRIGGER				
<b>Configuration class</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>VariantPreCompile:</b></td><td style="padding: 2px;">VariantPreCompile</td></tr> <tr> <td style="padding: 2px;"><b>VariantPostBuild:</b></td><td style="padding: 2px;">VariantPostBuild</td></tr> </table>	<b>VariantPreCompile:</b>	VariantPreCompile	<b>VariantPostBuild:</b>	VariantPostBuild
<b>VariantPreCompile:</b>	VariantPreCompile				
<b>VariantPostBuild:</b>	VariantPostBuild				
<b>Origin</b>	AUTOSAR_ECU				

### 5.2.1.79. BswMActionListItem

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMAbortOnFail</a>	1..1
<a href="#">BswMActionListIndex</a>	1..1
<a href="#">BswMActionListItemRef</a>	1..1
<a href="#">BswMReportFailToDemRef</a>	0..1

<b>Parameter Name</b>	<b>BswMAbortOnFail</b>					
<b>Description</b>	<p>This parameter defines if the execution of the action list shall be aborted if this specific action returns <code>E_NOT_OK</code>.</p> <p><i>Note that this is only applicable for actions that have <code>E_NOT_OK</code> as a possible return value.</i></p>					
<b>Multiplicity</b>	1..1					
<b>Type</b>	BOOLEAN					
<b>Default value</b>	false					
<b>Configuration class</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>VariantPreCompile:</b></td><td style="padding: 2px;">VariantPreCompile</td></tr> <tr> <td style="padding: 2px;"><b>VariantPostBuild:</b></td><td style="padding: 2px;">VariantPostBuild</td></tr> </table>	<b>VariantPreCompile:</b>	VariantPreCompile	<b>VariantPostBuild:</b>	VariantPostBuild	
<b>VariantPreCompile:</b>	VariantPreCompile					
<b>VariantPostBuild:</b>	VariantPostBuild					



<b>Origin</b>	AUTOSAR_ECUC
---------------	--------------

<b>Parameter Name</b>	<b>BswMActionListIndex</b>	
<b>Description</b>	This parameter defines the index of the action in the action list.  It is used define in which order the actions shall be performed.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMActionListItemRef</b>	
<b>Description</b>	The action item can either be an atomic action or a reference to another action list or rule.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	CHOICE-REFERENCE	
<b>Range</b>	node:paths(as:modconf('BswM')[1]/BswMConfig/*/BswMMODEControl/BswMAction/*[asc_bswm:getBswMConfig(.) = asc_bswm:getBswMConfig(node:current())]) node:paths(as:modconf('BswM')[1]/BswMConfig/*/BswMMODEControl/BswMActionList/*[(asc_bswm:getBswMConfig(.) = asc_bswm:getBswMConfig(node:current())) and (node:name(.) != node:name(node:current('/.../...)))]) node:paths(as:modconf('BswM')[1]/BswMConfig/*/BswMARbitration/BswMRule/*[asc_bswm:getBswMConfig(.) = asc_bswm:getBswMConfig(node:current())])	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMReportFailToDemRef</b>	
<b>Description</b>	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.	
<b>Multiplicity</b>	0..1	



Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.80. BswMRteModeRequestPort

Containers included		
Container name	Multiplicity	Description
<a href="#">BswMRteModeRequest-PortInterfaceRef</a>	1..1	This is an instance reference to the variable data prototype used for the mode request.

Parameters included	
Parameter name	Multiplicity
<a href="#">BswMRteModeRequestPortInterfaceMappingRef</a>	0..1

Parameter Name	<b>BswMRteModeRequestPortInterfaceMappingRef</b>	
Description	This is a foreign reference to the variable and parameter interface mapping used for the mode request.	
Multiplicity	0..1	
Type	FOREIGN-REFERENCE	
Configuration class	Link:	VariantPostBuild
	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

### 5.2.1.81. BswMRteModeRequestPortInterfaceRef

Parameters included		
Parameter name	Multiplicity	
<a href="#">TARGET</a>	1..1	
<a href="#">CONTEXT</a>	0..n	
Parameter Name	<b>TARGET</b>	



<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	CONTEXT
<b>Multiplicity</b>	0..n
<b>Type</b>	REFERENCE
<b>Range</b>	SW-COMPONENT-PROTOTYPE*
	PORT-PROTOTYPE
<b>Origin</b>	AUTOSAR_ECUC

### 5.2.1.82. BswMSwitchPort

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMModeSwitchInterfaceRef</a>	1..1
<a href="#">BswMModeSwitchQueueLength</a>	1..1

<b>Parameter Name</b>	<b>BswMModeSwitchInterfaceRef</b>	
<b>Description</b>	Reference to the ModeSwitchInterface of this BswMModeSwitchPort.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FOREIGN-REFERENCE	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMModeSwitchQueueLength</b>	
<b>Description</b>	This parameters defines the queue length of the mode switch port.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	1	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild



<b>Origin</b>	AUTOSAR_ECUC
---------------	--------------

### 5.2.1.83. BswMGeneral

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">BswMQueuedArbitration</a>	0..1	This optional container controls the queued mode arbitration feature which ensures that the processing of an immediate request of any type or the processing of the BswM_MainFunction cannot be interrupted by other requests. This is achieved by queueing the requests which interrupt the processing of the existing immediate request or the processing of the BswM_MainFunction. The queued requests are executed in the order of their arrival after the processing they interrupted has been completed. If the container is disabled (by default), BswM's behaviour remains the same as before. If the container is enabled, the queued mode arbitration feature is enabled and can be finely tuned via the provided parameters.
<a href="#">BswMUserIncludeFiles</a>	0..1	Collection of header file names which shall be included by the BswM.

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMBinaryGenerationVerified</a>	1..1
<a href="#">BswMCanSMEEnabled</a>	1..1
<a href="#">BswMComMEnabled</a>	1..1
<a href="#">BswMDcmEnabled</a>	1..1
<a href="#">BswMLegacyDcmEnabled</a>	1..1
<a href="#">BswMDevErrorDetect</a>	1..1
<a href="#">BswMEcuMEnabled</a>	1..1
<a href="#">BswMEthIfEnabled</a>	1..1
<a href="#">BswMEthSMEEnabled</a>	1..1
<a href="#">BswMFrSMEEnabled</a>	1..1
<a href="#">BswMGenericRequestEnabled</a>	1..1
<a href="#">BswMJ1939DcmEnabled</a>	1..1

**Parameters included**

<a href="#">BswMJ1939NmEnabled</a>	1..1
<a href="#">BswMLinSMEEnabled</a>	1..1
<a href="#">BswMLinTPEnabled</a>	1..1
<a href="#">BswMLogicalExpressionsOptimization</a>	1..1
<a href="#">BswMActionsOptimization</a>	1..1
<a href="#">BswMMainFunctionPeriod</a>	0..1
<a href="#">BswMMaxNumActionLists</a>	1..1
<a href="#">BswMMaxNumRules</a>	1..1
<a href="#">BswMNmEnabled</a>	1..1
<a href="#">BswMNvMEnabled</a>	1..1
<a href="#">BswMRelocatablePbcfgEnable</a>	1..1
<a href="#">BswMRteUsage</a>	1..1
<a href="#">BswMSchMEnabled</a>	1..1
<a href="#">BswMSdEnabled</a>	1..1
<a href="#">BswMVersionInfoApi</a>	1..1
<a href="#">BswMWdgMEnabled</a>	1..1

**Parameter Name****BswMBinaryGenerationVerified**

<b>Description</b>	If this parameter is TRUE and the selected variant is VariantPostBuild then symbolic names are not allowed as valid values for the BswMBswModelInitValue-Mode parameter.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

**Parameter Name****BswMCanSMEEnabled**

<b>Description</b>	Enable/disable CanSM module related BswM API <ul style="list-style-type: none"> <li>▶ true : Enabled</li> <li>▶ false: Disabled</li> </ul>
<b>Multiplicity</b>	1..1



Type	BOOLEAN	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>BswMComMEnabled</b>	
Description	Enable/disable ComM module related BswM API <ul style="list-style-type: none"> <li>▶ true : Enabled</li> <li>▶ false: Disabled</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>BswMDcmEnabled</b>	
Description	Enable/disable Dcm module related BswM API: <ul style="list-style-type: none"> <li>▶ BswM_Dcm_CommunicationMode_CurrentState</li> <li>▶ BswM_Dcm_ApplicationUpdated (stub only)</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>BswMLegacyDcmEnabled</b>	
Description	Enable/disable the 4.0.2 legacy Dcm related API: <ul style="list-style-type: none"> <li>▶ BswM_Dcm_RequestCommunicationMode</li> <li>▶ BswM_Dcm_RequestSessionMode</li> <li>▶ BswM_Dcm_RequestResetMode</li> </ul> <p>Note that a 4.0.2 compatible Dcm is needed in order for the BswM to produce compilable code.</p>	



<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>BswMDevErrorDetect</b>	
<b>Description</b>	Switches the Development Error Detection and Notification ON or OFF.  ► true : Enabled / ON ► false: Disabled / OFF	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMEcuMEnabled</b>	
<b>Description</b>	Enables/Disables EcuM-related API  ► true : Enabled ► false: Disabled	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMEthIfEnabled</b>	
<b>Description</b>	Enables/Disables EthIf-related API  ► true : Enabled ► false: Disabled	
<b>Multiplicity</b>	1..1	



Type	BOOLEAN	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>BswMEthSMEEnabled</b>	
Description	Enables/Disables EthSM-related API <ul style="list-style-type: none"> <li>▶ true : Enabled</li> <li>▶ false: Disabled</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>BswMFrSMEEnabled</b>	
Description	Enables/Disables FrSM-related API <ul style="list-style-type: none"> <li>▶ true : Enabled</li> <li>▶ false: Disabled</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>BswMGenericRequestEnabled</b>	
Description	Enables/Disables GenericRequest-related API <ul style="list-style-type: none"> <li>▶ true : Enabled</li> <li>▶ false: Disabled</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Configuration class	VariantPreCompile:	



	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMJ1939DcmEnabled</b>	
<b>Description</b>	Enable/disable J1939Dcm module related BswM API: true: Enabled false: Disabled	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMJ1939NmEnabled</b>	
<b>Description</b>	Enable/disable J1939Nm module related BswM API. true: Enabled false: Disabled	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMLinSMEEnabled</b>	
<b>Description</b>	Enables/Disables LinSM-related API <ul style="list-style-type: none"> <li>▶ true : Enabled</li> <li>▶ false: Disabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMLinTPEnabled</b>	
<b>Description</b>	Enables/Disables LinTP-related API <ul style="list-style-type: none"> <li>▶ true : Enabled</li> </ul>	



	▶ false: Disabled	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

Parameter Name	<b>BswMLogicalExpressionsOptimization</b>	
<b>Description</b>	This parameter controls the optimization of logical expressions w.r.t ROM consumption. If set to TRUE, BswM will only generate linktime artifacts for the expressions which are directly referenced by rules, ignoring unused or nested expressions. If set to FALSE, BswM will generate linktime artifacts for all expressions. This parameter is ignored and treated as TRUE if IMPLEMENTATION_CONFIG_VARIANT = VariantPreCompile. IMPORTANTE NOTE: If BswMLogicalExpressionsOptimization = TRUE, rules cannot be assigned a previously unused expression or an expression which is part of another logical expression at postbuild time.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

Parameter Name	<b>BswMActionsOptimization</b>	
<b>Description</b>	This parameter controls the optimization of actions w.r.t ROM consumption. If set to TRUE, BswM will only generate linktime artifacts for the actions which are actually used and ignoring all others. If set to FALSE, BswM will generate linktime artifacts for all actions. This parameter is ignored and treated as TRUE if IMPLEMENTATION_CONFIG_VARIANT = VariantPreCompile. IMPORTANTE NOTE: If BswMActionsOptimization = TRUE, action list items cannot be assigned previously unused actions at postbuild time.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild



<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

<b>Parameter Name</b>	<b>BswMMainFunctionPeriod</b>	
<b>Description</b>	The cycle time of the periodic main function of BswM defined in seconds .	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
	<b>Link:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMMaxNumActionLists</b>	
<b>Description</b>	The maximum number of BswM action lists supported by this configuration.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>BswMMaxNumRules</b>	
<b>Description</b>	The maximum number of BswM rules supported by this configuration.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>BswMNmEnabled</b>	
<b>Description</b>	Enables/Disables NM-related API ► true : Enabled ► false: Disabled	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile



	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>BswMNvMEnabled</b>	
<b>Description</b>	Enables/Disables NvM-related API <ul style="list-style-type: none"> <li>▶ true : Enabled</li> <li>▶ false: Disabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>BswMRelocatablePbcfgEnable</b>	
<b>Description</b>	Enables/disable support for relocatable post-build configuration. <ul style="list-style-type: none"> <li>▶ True: Postbuild configuration relocatable in memory.</li> <li>▶ False: Postbuild configuration not relocatable in memory.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	true	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>BswMRteUsage</b>	
<b>Label</b>	Enable Rte Usage	
<b>Description</b>	This parameter enables the usage of the RTE for this module.  For an easy integration it is recommended to disable the usage of the RTE at the beginning of the integration work.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild



<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

<b>Parameter Name</b>	<b>BswMSchMEnabled</b>	
<b>Description</b>	This parameter is disabled and its default value is always 'true' as the SchM interface is always needed by the BswM.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	true	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMSdEnabled</b>	
<b>Description</b>	enable/disable Sd module related BswM_Sd_CurrentState API:	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMVersionInfoApi</b>	
<b>Description</b>	Switches the possibility to read the version information with the service <code>BswM_GetVersionInfo()</code> .	
	<ul style="list-style-type: none"> <li>▶ true: Enabled</li> <li>▶ false: Disabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	true	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>BswMWdgMEnabled</b>
-----------------------	------------------------



<b>Description</b>	Enables/Disables WdgM-related API <ul style="list-style-type: none"> <li>▶ true : Enabled</li> <li>▶ false: Disabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.84. BswMQueuedArbitration

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMArbitrationQueueLength</a>	1..1
<a href="#">BswMQueueDeferredRequests</a>	1..1
<a href="#">BswMDelayMainFunction</a>	1..1

<b>Parameter Name</b>	<b>BswMArbitrationQueueLength</b>	
<b>Description</b>	This parameter controls the size of the arbitration queue. By default, the value of this parameter is the maximum number of mode request ports in all BswM configurations but this can be manually changed in order to fit the project's requirements. In case the arbitration queue is full and a new request needs to be enqueued, it will be rejected and a DET error will be reported if BswMDevErrorDetect is set to TRUE. Note that a larger queue will also increase the RAM usage of BswM.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Range</b>	<=255 >=1	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>BswMQueueDeferredRequests</b>
-----------------------	----------------------------------



<b>Description</b>	This parameter can be used to exclude or include deferred mode request ports from being queued. If set to FALSE, only immediate mode request ports will be queued. If set to TRUE, both immediate and deferred mode request ports will be queued.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

Parameter Name	<b>BswMDelayMainFunction</b>	
<b>Description</b>	This parameter can be used to handle the situation when the BswM_MainFunction interrupts the execution of an immediate mode request. If set to FALSE, the BswM_MainFunction will execute even if there is an immediate request being processed (thus bypassing the arbitration feature). If set to TRUE, the BswM_MainFunction will be delayed until the processing of the immediate mode request is completed. Note that any other request interrupting the processing of the immediate request the BswM_MainFunction interrupted will also be processed before the BswM_MainFunction.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.2.1.85. BswMUserIncludeFiles

<b>Parameters included</b>		
<b>Parameter name</b>	<b>Multiplicity</b>	
<a href="#">BswMUserIncludeFile</a>	1..n	

Parameter Name	<b>BswMUserIncludeFile</b>
Multiplicity	1..n



Type	STRING	
Configuration class	PreCompile:	VariantPreCompile
	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.86. ReportToDem

Parameters included	
Parameter name	Multiplicity
<a href="#">BswMActionFailedReportToDem</a>	1..1
<a href="#">BswMActionFailedReportToDemDetErrorId</a>	1..1

Parameter Name	<b>BswMActionFailedReportToDem</b>	
Label	BswM Action Failure	
Description	<p>Selects the handling of the production error: <i>BSWM_E_ACTION_FAILED</i></p> <ul style="list-style-type: none"> <li>▶ DEM: All errors are reported to the Diagnostics Event Manager (Dem).</li> <li>▶ DET: All errors are reported to the Development Error Tracer (Det) if enabled.</li> <li>▶ DISABLE: Production errors are not reported at all.</li> </ul>	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	DISABLE	
Range	DEM DET DISABLE	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>BswMActionFailedReportToDemDetErrorId</b>
Label	BswM Action Failed DemToDet ErrorId
Description	This parameter is used to report Dem to Det. It shall be checked that the variable <code>BswMActionFailedReportToDemDetErrorId</code> is set to a value between



	15 and 255; the default value shall be 15. A preprocessor define <code>BSWM_E_ACTION_FAILED</code> shall be generated holding the value of <code>BswMActionFailedReportToDemDetErrorId</code> .				
Dependencies:					
	<ul style="list-style-type: none"> <li>▶ The identifier to report Dem to Det must not be zero-based.</li> </ul>				
<b>Multiplicity</b>	1..1				
<b>Type</b>	INTEGER				
<b>Default value</b>	15				
<b>Range</b>	<p>&lt;=255</p> <hr/> <p>&gt;=15</p>				
<b>Configuration class</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>PreCompile:</b></td><td style="padding: 2px;">VariantPreCompile</td></tr> <tr> <td style="padding: 2px;"><b>PostBuild:</b></td><td style="padding: 2px;">VariantPostBuild</td></tr> </table>	<b>PreCompile:</b>	VariantPreCompile	<b>PostBuild:</b>	VariantPostBuild
<b>PreCompile:</b>	VariantPreCompile				
<b>PostBuild:</b>	VariantPostBuild				
<b>Origin</b>	Elektrobit Automotive GmbH				

### 5.2.1.87. BswMDefensiveProgramming

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">BswMDefProgEnabled</a>	1..1
<a href="#">BswMPrecondAssertEnabled</a>	1..1
<a href="#">BswMPostcondAssertEnabled</a>	1..1
<a href="#">BswMStaticAssertEnabled</a>	1..1
<a href="#">BswMUnreachAssertEnabled</a>	1..1
<a href="#">BswMInvariantAssertEnabled</a>	1..1

<b>Parameter Name</b>	<b>BswMDefProgEnabled</b>
<b>Label</b>	Enable Defensive Programming
<b>Description</b>	<p>Enables or disables the defensive programming feature for the module BswM.</p> <p>Note: This feature is dependent on the use of the development error detection module. To use the defensive programming feature, proceed as follows:</p> <ol style="list-style-type: none"> <li>1. Enable development error detection</li> <li>2. Enable defensive programming</li> </ol>



	3. Enable assertions as required	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>BswMPrecondAssertEnabled</b>	
<b>Label</b>	Enable Precondition Assertions	
<b>Description</b>	<p>Enables handling of precondition assertion checks reported from the module BswM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>BswMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>BswMDefProgEnabled</code>): must be enabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>BswMPostcondAssertEnabled</b>	
<b>Label</b>	Enable Postcondition Assertions	
<b>Description</b>	<p>Enables handling of postcondition assertion checks reported from the module BswM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>BswMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>BswMDefProgEnabled</code>): must be enabled</li> </ul>	
<b>Multiplicity</b>	1..1	



Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>BswMStaticAssertEnabled</b>	
Label	Enable Static Assertions	
Description	<p>Enables handling of static assertion checks reported from the module BswM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>BswMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>BswMDefProgEnabled</code>): must be enabled</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>BswMUreachAssertEnabled</b>	
Label	Enable Unreachable Code Assertions	
Description	<p>Enables handling of unreachable code assertion checks reported from the module BswM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>BswMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>BswMDefProgEnabled</code>): must be enabled</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	



<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>BswMInvariantAssertEnabled</b>	
<b>Label</b>	Enable Invariant Assertions	
<b>Description</b>	<p>Enables handling of invariant assertion checks reported from functions of the module BswM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>BswMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>BswMDefProgEnabled</code>): must be enabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.2.1.88. CommonPublishedInformation

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">ArMajorVersion</a>	1..1
<a href="#">ArMinorVersion</a>	1..1
<a href="#">ArPatchVersion</a>	1..1
<a href="#">SwMajorVersion</a>	1..1
<a href="#">SwMinorVersion</a>	1..1
<a href="#">SwPatchVersion</a>	1..1
<a href="#">ModuleId</a>	1..1
<a href="#">VendorId</a>	1..1
<a href="#">Release</a>	1..1



<b>Parameter Name</b>	<b>ArMajorVersion</b>
<b>Label</b>	AUTOSAR Major Version
<b>Description</b>	Major version number of AUTOSAR specification on which the appropriate implementation is based on.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	1
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>ArMinorVersion</b>
<b>Label</b>	AUTOSAR Minor Version
<b>Description</b>	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	2
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>ArPatchVersion</b>
<b>Label</b>	AUTOSAR Patch Version
<b>Description</b>	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	0
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SwMajorVersion</b>
<b>Label</b>	Software Major Version
<b>Description</b>	Major version number of the vendor specific implementation of the module.
<b>Multiplicity</b>	1..1



<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	1
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SwMinorVersion</b>
<b>Label</b>	Software Minor Version
<b>Description</b>	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	15
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SwPatchVersion</b>
<b>Label</b>	Software Patch Version
<b>Description</b>	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	7
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>ModuleId</b>
<b>Label</b>	Numeric Module ID
<b>Description</b>	Module ID of this module from Module List
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	42
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH



<b>Parameter Name</b>	<b>VendorId</b>
<b>Label</b>	Vendor ID
<b>Description</b>	Vendor ID of the dedicated implementation of this module according to the AU-TOSAR vendor list
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	1
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>Release</b>
<b>Label</b>	Release Information
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING_LABEL
<b>Default value</b>	
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.2.1.89. PublishedInformation

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">PbcfgMSupport</a>	1..1

<b>Parameter Name</b>	<b>PbcfgMSupport</b>
<b>Label</b>	PbcfgM support
<b>Description</b>	Specifies whether or not the BswM can use the PbcfgM module for post-build support.
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	true
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH



## 5.2.2. Application programming interface (API)

### 5.2.2.1. Macro constants

#### 5.2.2.1.1. BSWM\_AR\_RELEASE\_MAJOR\_VERSION

Purpose	AUTOSAR release major version.
Value	4U

#### 5.2.2.1.2. BSWM\_AR\_RELEASE\_MINOR\_VERSION

Purpose	AUTOSAR release minor version.
Value	0U

#### 5.2.2.1.3. BSWM\_AR\_RELEASE\_REVISION\_VERSION

Purpose	AUTOSAR release revision version.
Value	3U

#### 5.2.2.1.4. BSWM\_E\_INVALID\_PAR

Purpose	Error Code.
Value	0x03U
Description	A parameter was invalid.

#### 5.2.2.1.5. BSWM\_E\_NO\_INIT

Purpose	Error Code.
Value	0x01U



<b>Description</b>	A service was called prior to initialization.
--------------------	---

#### 5.2.2.1.6. BSWM\_E\_NULL\_POINTER

<b>Purpose</b>	Error Code.
<b>Value</b>	0x02U
<b>Description</b>	A null pointer was passed as an argument.

#### 5.2.2.1.7. BSWM\_E\_PARAM\_CONFIG

<b>Purpose</b>	Error Code.
<b>Value</b>	0x06U
<b>Description</b>	The provided configuration is inconsistent.

#### 5.2.2.1.8. BSWM\_E\_PARAM\_POINTER

<b>Purpose</b>	Error Code.
<b>Value</b>	0x07U
<b>Description</b>	A parameter pointer was invalid.

#### 5.2.2.1.9. BSWM\_E\_QUEUE\_FULL

<b>Purpose</b>	Error Code.
<b>Value</b>	0x09U
<b>Description</b>	A new mode request needs to be queued but the mode arbitration queue has reached its maximum capacity.

#### 5.2.2.1.10. BSWM\_E\_REQ\_MODE\_OUT\_OF\_RANGE

<b>Purpose</b>	Error Code.
<b>Value</b>	0x05U
<b>Description</b>	A requested mode was out of range.



#### 5.2.2.1.11. BSWM\_E\_REQ\_USER\_OUT\_OF\_RANGE

<b>Purpose</b>	Error Code.
<b>Value</b>	0x04U
<b>Description</b>	A requesting user was out of range.

#### 5.2.2.1.12. BSWM\_E\_WRONG\_CONTEXT

<b>Purpose</b>	Error code.
<b>Value</b>	0x08U
<b>Description</b>	The API has been called from an invalid partition context.

#### 5.2.2.1.13. BSWM\_MODULE\_ID

<b>Purpose</b>	AUTOSAR module identification.
<b>Value</b>	42U

#### 5.2.2.1.14. BSWM\_NO\_CFGCLASSMIX\_REQUIRED

<b>Purpose</b>	
<b>Value</b>	

#### 5.2.2.1.15. BSWM\_SID\_CANSM\_CURRENT\_STATE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x05U
<b>Description</b>	Definition of service ID for BswM_CanSM_CurrentState().

#### 5.2.2.1.16. BSWM\_SID\_COMM\_CURRENT\_MODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x0EU
<b>Description</b>	Definition of service ID for BswM_ComM_CurrentMode().



#### 5.2.2.1.17. BSWM\_SID\_COMM\_CURRENT\_PNC\_MODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x15U
<b>Description</b>	Definition of service ID for BswM_ComM_CurrentPNCMode().

#### 5.2.2.1.18. BSWM\_SID\_COMM\_INITIATE\_RESET

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x22U
<b>Description</b>	Definition of service ID for BswM_ComM_InitiateReset().

#### 5.2.2.1.19. BSWM\_SID\_DCM\_APPLICATION\_UPDATED

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x14U
<b>Description</b>	Definition of service ID for BswM_Dcm_ApplicationUpdated().

#### 5.2.2.1.20. BSWM\_SID\_DCM\_COMMUNICATIONMODE\_CURRENT\_STATE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x06U
<b>Description</b>	Definition of service ID for BswM_Dcm_CommunicationMode_CurrentState().

#### 5.2.2.1.21. BSWM\_SID\_DCM\_REQUEST\_COMMUNICATION\_MODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0xFEU
<b>Description</b>	Definition of service ID for BswM_Dcm_RequestCommunicationMode().

#### 5.2.2.1.22. BSWM\_SID\_DCM\_REQUEST\_RESET\_MODE

<b>Purpose</b>	AUTOSAR API service ID.
----------------	-------------------------



<b>Value</b>	0x07U
<b>Description</b>	Definition of service ID for <code>BswM_Dcm_RequestResetMode()</code> .

#### 5.2.2.1.23. BSWM\_SID\_DCM\_REQUEST\_SESSION\_MODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x08U
<b>Description</b>	Definition of service ID for <code>BswM_Dcm_RequestSessionMode()</code> .

#### 5.2.2.1.24. BSWM\_SID\_DEINIT

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x04U
<b>Description</b>	Definition of service ID for <code>BswM_Deinit()</code> .

#### 5.2.2.1.25. BSWM\_SID\_ECUM\_CURRENT\_STATE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x0FU
<b>Description</b>	Definition of service ID for <code>BswM_EcuM_CurrentState()</code> .

#### 5.2.2.1.26. BSWM\_SID\_ECUM\_CURRENT\_WAKEUP

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x10U
<b>Description</b>	Definition of service ID for <code>BswM_EcuM_CurrentWakeups()</code> .

#### 5.2.2.1.27. BSWM\_SID\_ETHIF\_PORTGROUPLINKSTATECHG

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x26U
<b>Description</b>	Definition of service ID for <code>BswM_EthIf_PortGroupLinkStateChg()</code> .



#### 5.2.2.1.28. BSWM\_SID\_ETHSM\_CURRENT\_STATE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x0DU
<b>Description</b>	Definition of service ID for BswM_EthSM_CurrentState().

#### 5.2.2.1.29. BSWM\_SID\_FRSM\_CURRENT\_STATE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x0CU
<b>Description</b>	Definition of service ID for BswM_FrSM_CurrentState().

#### 5.2.2.1.30. BSWM\_SID\_GET\_VERSION\_INFO

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x01U
<b>Description</b>	Definition of service ID for <a href="#">BswM_GetVersionInfo()</a> .

#### 5.2.2.1.31. BSWM\_SID\_INIT

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x00U
<b>Description</b>	Definition of service ID for <a href="#">BswM_Init()</a> .

#### 5.2.2.1.32. BSWM\_SID\_INVALID

<b>Purpose</b>	
<b>Value</b>	0xFFU

#### 5.2.2.1.33. BSWM\_SID\_J1939DCMBROADCASTSTATUS

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x1BU
<b>Description</b>	Definition of service ID for BswM_J1939DcmBroadcastStatus().



#### 5.2.2.1.34. BSWM\_SID\_J1939NM\_STATECHANGENOTIFICATION

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x18U
<b>Description</b>	Definition of service ID for BswM_J1939Nm_StateChangeNotification().

#### 5.2.2.1.35. BSWM\_SID\_LINSM\_CURRENT\_SCHEDULE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x0AU
<b>Description</b>	Definition of service ID for BswM_LinSM_CurrentSchedule().

#### 5.2.2.1.36. BSWM\_SID\_LINSM\_CURRENT\_STATE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x09U
<b>Description</b>	Definition of service ID for BswM_LinSM_CurrentState().

#### 5.2.2.1.37. BSWM\_SID\_LINTP\_REQUESTMODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x0BU
<b>Description</b>	Definition of service ID for BswM_LinTp_RequestMode().

#### 5.2.2.1.38. BSWM\_SID\_MAINFUNCTION

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x03U
<b>Description</b>	Definition of service ID for BswM_MainFunction().

#### 5.2.2.1.39. BSWM\_SID\_NM\_CAR\_WAKEUP\_INDICATION

<b>Purpose</b>	AUTOSAR API service ID.
----------------	-------------------------



<b>Value</b>	0x24U
<b>Description</b>	Definition of service ID for BswM_Nmlf_CarWakeUpIndication().

#### 5.2.2.1.40. BSWM\_SID\_NVM\_CURRENT\_BLOCK\_MODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x16U
<b>Description</b>	Definition of service ID for BswM_NvM_CurrentBlockMode().

#### 5.2.2.1.41. BSWM\_SID\_NVM\_CURRENT\_JOB\_MODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x17U
<b>Description</b>	Definition of service ID for BswM_NvM_CurrentJobMode().

#### 5.2.2.1.42. BSWM\_SID\_REQUESTMODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x02U
<b>Description</b>	Definition of service ID for bswm_requestmode().

#### 5.2.2.1.43. BSWM\_SID\_SD\_CLIENTSERVICE\_CURRENTSTATE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x1FU
<b>Description</b>	Definition of service ID for BswM_Sd_ClientServiceCurrentState().

#### 5.2.2.1.44. BSWM\_SID\_SD\_CONSUMEDEVENTGROUP\_CURRENTSTATE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x21U
<b>Description</b>	Definition of service ID for BswM_Sd_ConsumedEventGroupCurrentState().



#### 5.2.2.1.45. BSWM\_SID\_SD\_EVENTHANDLER\_CURRENTSTATE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x20U
<b>Description</b>	Definition of service ID for BswM_Sd_EventHandlerCurrentState().

#### 5.2.2.1.46. BSWM\_SID\_TRIGGER\_SLAVE\_RTE\_STOP

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x13U
<b>Description</b>	Definition of service ID for BswM_TriggerSlaveRTEStop().

#### 5.2.2.1.47. BSWM\_SID\_TRIGGER\_STARTUP\_PHASE2

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x12U
<b>Description</b>	Definition of service ID for BswM_TriggerStartUpPhase2().

#### 5.2.2.1.48. BSWM\_SID\_WDGM\_REQUEST\_PARTITION\_RESET

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x11U
<b>Description</b>	Definition of service ID for BswM_WdgM_RequestPartitionReset().

#### 5.2.2.1.49. BSWM\_SW\_MAJOR\_VERSION

<b>Purpose</b>	AUTOSAR module major version.
<b>Value</b>	1U

#### 5.2.2.1.50. BSWM\_SW\_MINOR\_VERSION

<b>Purpose</b>	AUTOSAR module minor version.
----------------	-------------------------------



<b>Value</b>	15U
--------------	-----

#### 5.2.2.1.51. BSWM\_SW\_PATCH\_VERSION

<b>Purpose</b>	AUTOSAR module patch version.
<b>Value</b>	7U

#### 5.2.2.1.52. BSWM\_VENDOR\_ID

<b>Purpose</b>	AUTOSAR vendor identification: Elektrobit Automotive GmbH.
<b>Value</b>	1U

### 5.2.2.2. Functions

#### 5.2.2.2.1. BswM\_Deinit

<b>Purpose</b>	Deinitializes the BSW Mode Manager.
<b>Synopsis</b>	<code>void BswM_Deinit ( void );</code>
<b>Service ID</b>	4
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	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  Precondition: None

#### 5.2.2.2.2. BswM\_GetVersionInfo

<b>Purpose</b>	Return the modules version information.	
<b>Synopsis</b>	<code>void BswM_GetVersionInfo ( Std_VersionInfoType * VersionInfoPtr );</code>	
<b>Parameters (out)</b>	VersionInfoPtr	- Pointer to struct to be filled with the version information.



<b>Description</b>	This function provides the information to module vendor ID, module ID and software version major.minor.patch  Precondition: BSWM_VERSION_INFO_API = STD_ON
--------------------	--

### 5.2.2.2.3. BswM\_Init

<b>Purpose</b>	Initializes the BSW Mode Manager.	
<b>Synopsis</b>	<code>void BswM_Init ( const BswM_ConfigType * ConfigPtr );</code>	
<b>Service ID</b>	0	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	ConfigPtr	- Pointer to a selected configuration structure.
<b>Description</b>	<p>This routine initializes the BSW Mode Manager. After execution of this routine the BSW Mode Manager is ready to arbitrate incoming mode requests</p> <p>Precondition: None</p>	

### 5.2.2.2.4. BswM\_IsValidConfig

<b>Purpose</b>	Validates the post-build configuration data structure passed to the BswM_Init function.	
<b>Synopsis</b>	<code>Std_ReturnType BswM_IsValidConfig ( const void * voidPtr );</code>	
<b>Parameters (in)</b>	voidPtr	The pointer to a BswM post-build data structure. If a NULL_PTR is passed, the BswM will attempt to retrieve the BswM post-build configuration from the PbcfgM module.
<b>Parameters (in,out)</b>	voidPtr	The pointer to a BswM post-build data structure. If a NULL_PTR is passed, the BswM will attempt to retrieve the BswM post-build configuration from the PbcfgM module.
<b>Return Value</b>		
E_OK	When the pre-compile, link-time and platform hash values stored within the post-	



build structure correspond to the hash values of the compiled source files. Otherwise, E\_NOT\_OK will be returned.

### 5.2.2.2.5. BswM\_RequestMode

<b>Purpose</b>	Generic function call to request modes.	
<b>Synopsis</b>	<pre>void <b>BswM_RequestMode</b> ( BswM_UserType requesting_user , BswM_- ModeType requested_mode );</pre>	
<b>Service ID</b>	2	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	requesting_user	The user that requests the mode
	requested_mode	The requested mode.
<b>Description</b>	<p>This function is used to request modes and shall only be used by other BSW modules that does not have a specific mode request interface.</p> <p>Precondition: None</p>	

## 5.2.3. Integration notes

### 5.2.3.1. Exclusive areas

This section describes the exclusive areas used by the BswM module.

#### 5.2.3.1.1. SCHM\_BSWM\_EXCLUSIVE\_AREA

<b>Protected data structures</b>	This memory section is used to protect all shared data within the same partition from mutual access. Note that each BswM partition will have its own exclusive area.
<b>Recommended locking mechanism</b>	These exclusive areas must always be protected by a locking mechanism. The options for locking are described in the EB tresos AutoCore Generic documentation. Refer to the section Mapping exclusive areas in the



basic software modules in the Integration notes section for details.

### 5.2.3.2. Production errors

The BswM supports reporting of production errors for action list items. If you configure the `BswMReportFail-ToDemRef` parameter of an action list item to a valid `DemEventParameter` reference, the status is reported as follows:

- ▶ If the configured action of the action list item returns `E_OK`, the Dem error `BSWM_E_ACTION_FAILED` with a `DEM_EVENT_STATUS_PASSED` status is reported.
- ▶ If the configured action of the action list item returns `E_NOT_OK`, the Dem error `BSWM_E_ACTION_FAILED` with a `DEM_EVENT_STATUS_FAILED` is reported.

You can use the same reference to a `DemEventParameter` for multiple action list items.

---

**NOTE**
**Availability of error report**


The error report is only available if the `BswMActionListItemRef` parameter references a `BswMAction`. If the `BswMActionListItemRef` parameter references a `BswMActionList` or a `BswMRule`, no Dem errors are reported.

---

### 5.2.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section `Memory mapping and compiler abstraction` in the Integration notes section for details.

The following table provides the list of sections that may be mapped for this module:

<b>Memory section</b>
CONFIG_DATA_UNSPECIFIED
VAR_INIT_UNSPECIFIED
VAR_NO_INIT_UNSPECIFIED
CONST_UNSPECIFIED
INTERNAL_VAR_POWER_ON_INIT_UNSPECIFIED
CODE
VAR_NO_INIT_8
VAR_POWER_ON_INIT_32

#### 5.2.3.4. Integration requirements

**WARNING      Integration requirements list is not exhaustive**



The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user's guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

##### 5.2.3.4.1. intgr.BswM.EB\_INTREQ\_BswM\_0001

<b>Description</b>	The PbcfgM module support shall always be used in case the selected IMPLEMENTATION_CONFIG_VARIANT is VariantPostBuild.
<b>Rationale</b>	The EcuM calls the BswM_Init API with NULL_PTR as a parameter. Consequently, the BswM must obtain the the post-build configuration via the PbcfgM.

##### 5.2.3.4.2. intgr.BswM.EB\_INTREQ\_BswM\_0002

<b>Description</b>	The BswM shall reject a configuration in which the selected IMPLEMENTATION_CONFIG_VARIANT is VariantPostBuild and a symbolic name value is configured as an initial value of a mode request port via BswMBswModelInitValueMode.
<b>Rationale</b>	In order to support the binary generation of the BswM's post-build configuration, the BswM now requires that the initial value defined within the parameter BswMBswModelInitValueMode be a numerical constant if the selected variant is VariantPostBuild. This constraint is due to the fact the numerical value of the symbolic value can only be known at runtime but the XGEN generation of the binary representation of the post-build configuration needs to be done before.

##### 5.2.3.4.3. intgr.BswM.EB\_INTREQ\_BswM\_0003

<b>Description</b>	The BswMSchMEnabled parameter shall be read only and its default value set to 'TRUE'.
<b>Rationale</b>	The SchM interface is always needed by the BswM after Bsw distribution support was added.

##### 5.2.3.4.4. intgr.BswM.EB\_INTREQ\_BswM\_0004

<b>Description</b>	It falls within the user's responsibilities to be aware and resolve any reentrancy issues that might occur when immediate mode request ports are configured.
--------------------	--



<b>Rationale</b>	The execution of both user callout functions and predefined BswM actions as a result of the mode arbitration of immediate mode request ports is potentially dangerous and can lead to unexpected behaviour. In case of the predefined BswM actions, the user must consult the API specification of the functions which are called by the BswM and verify their reentrancy status. As an example, the BswMPduGroupSwitch action triggers the execution of the Com_IpduGroupControl() API which is not reentrant. Configuring such an action within a rule whose logical expression uses an immediate mode request port might lead to unexpected behaviour.
------------------	---

#### 5.2.3.4.5. intgr.BswM.EB\_INTREQ\_BswM\_0005

<b>Description</b>	BswM API's shall not be called from CAT1 interrupts if multiple BswM instances are configured.
<b>Rationale</b>	In CAT1 interrupts the BswM cannot use the GetApplicationID API to determine the exact instance that will arbitrate the request.

#### 5.2.3.4.6. intgr.BswM.EB\_INTREQ\_BswM\_BswMPduGroupSwitch\_BswMDeadlineMonitoringControl

<b>Description</b>	BswMPduGroupSwitch and BswMDeadlineMonitoringControl actions can only be configured in one BswM instance
<b>Rationale</b>	The Com module only has one instance and the API's corresponding to the BswMPduGroupSwitch and BswMDeadlineMonitoringControl actions are not designed to be called concurrently. As the BswM allows multiple instances, only one of them can safely call the two Com API's.

#### 5.2.3.4.7. intgr.BswM.EB\_INTREQ\_BswMLogicalExpressionsOptimization

<b>Description</b>	If configuration parameter BswMLogicalExpressionsOptimization is set to TRUE, the parameter BswMRuleExpressionRef cannot reference previously unused logical expressions or logical expressions which are part of other logical expressions at PostBuild time.
<b>Rationale</b>	Linktime artifacts for each logical expression are needed in order for the BswMRuleExpressionRef parameter to be changed at PostBuild time. However, this increases ROM consumption. If changing BswMRuleExpressionRef in the above described manner is not required at PostBuild time, the linktime ROM consumption can be improved by setting BswMLogicalExpressionsOptimization to TRUE.



#### 5.2.3.4.8. intgr.BswM.EB\_INTREQ\_BswMActionsOptimization

<b>Description</b>	If configuration parameter BswMActionsOptimization is set to TRUE, the parameter BswMActionListRef cannot reference a previously unused action at PostBuild time.
<b>Rationale</b>	Linktime artifacts for each action are needed in order for the BswMActionListRef parameter to be changed at PostBuild time. However, this increases ROM consumption. If changing BswMActionListRef in the above described manner is not required at PostBuild time, the linktime ROM consumption can be improved by setting BswMActionsOptimization to TRUE.

#### 5.2.3.4.9. intgr.BswM.EB\_INTREQ\_BswMSchMSwitchPostbuild

<b>Description</b>	At PostBuild time a BswMSchMSwitch action cannot be configured for a rule referencing different mode request ports than the original rule it was mapped to.
<b>Rationale</b>	If at PostBuild time a BswMSchMSwitch is mapped to a rule referencing different mode request ports than the one it was mapped initially the basic software module description requires an update. Updating the basic software module description at PostBuild time is not supported in the ACG.

## 5.3. ComM

### 5.3.1. Configuration parameters

Containers included		
Container name	Multiplicity	Description
<a href="#">ComMDefensiveProgramming</a>	1..1	<b>Label:</b> Defensive Programming Options  Parameters for defensive programming
<a href="#">ComMConfigSet</a>	1..1	This container is the base for a multiple configuration set.
<a href="#">ComMGeneral</a>	1..1	<b>Label:</b> General Configuration  General configuration parameters of the Communication Manager.

**Containers included**

<a href="#">CommonPublishedInformation</a>	1..1	<b>Label:</b> Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
<a href="#">PublishedInformation</a>	1..1	<b>Label:</b> EB Published Information Additional published parameters not covered by CommonPublishedInformation container.

**Parameters included**

Parameter name	Multiplicity
<a href="#">IMPLEMENTATION_CONFIG_VARIANT</a>	1..1

<b>Parameter Name</b>	<b>IMPLEMENTATION_CONFIG_VARIANT</b>	
<b>Label</b>	Configuration Variant	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	VariantPostBuild	
<b>Range</b>	VariantPostBuild	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild

**5.3.1.1. ComMDefensiveProgramming****Parameters included**

Parameter name	Multiplicity
<a href="#">ComMDefProgEnabled</a>	1..1
<a href="#">ComMPrecondAssertEnabled</a>	1..1
<a href="#">ComMPostcondAssertEnabled</a>	1..1
<a href="#">ComMStaticAssertEnabled</a>	1..1
<a href="#">ComMUreachAssertEnabled</a>	1..1
<a href="#">ComMInvariantAssertEnabled</a>	1..1

<b>Parameter Name</b>	<b>ComMDefProgEnabled</b>
<b>Label</b>	Enable Defensive Programming
<b>Description</b>	Enables or disables the defensive programming feature for the module ComM.



	Note: This feature is dependent on the use of the development error detection module. To use the defensive programming feature, proceed as follows:  1. Enable development error detection 2. Enable defensive programming 3. Enable assertions as required
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>ComMPrecondAssertEnabled</b>	
<b>Label</b>	Enable Precondition Assertions	
<b>Description</b>	<p>Enables handling of precondition assertion checks reported from the module ComM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>ComMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>ComMDefProgEnabled</code>): must be enabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>ComMPostcondAssertEnabled</b>	
<b>Label</b>	Enable Postcondition Assertions	
<b>Description</b>	<p>Enables handling of postcondition assertion checks reported from the module ComM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>ComMDevErrorDetect</code>): must be enabled</li> </ul>	



	<ul style="list-style-type: none"> <li>▶ Enable Defensive Programming (<code>ComMDefProgEnabled</code>): must be enabled</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>ComMStaticAssertEnabled</b>
<b>Label</b>	Enable Static Assertions
<b>Description</b>	<p>Enables handling of static assertion checks reported from the module ComM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>ComMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>ComMDefProgEnabled</code>): must be enabled</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>ComMUnreachAssertEnabled</b>
<b>Label</b>	Enable Unreachable Code Assertions
<b>Description</b>	<p>Enables handling of unreachable code assertion checks reported from the module ComM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>ComMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>ComMDefProgEnabled</code>): must be enabled</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false



<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>ComMInvariantAssertEnabled</b>	
<b>Label</b>	Enable Invariant Assertions	
<b>Description</b>	<p>Enables handling of invariant assertion checks reported from functions of the module ComM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>ComMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>ComMDefProgEnabled</code>): must be enabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.3.1.2. ComMConfigSet

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#"><u>ComMChannel</u></a>	1..n	This container contains the configuration (parameters) of the bus channel(s). The channel parameters shall be harmonized within the whole communication stack.
<a href="#"><u>ComMPnc</u></a>	0..n	This container contains the configuration of the partial network cluster (PNC).
<a href="#"><u>ComMUser</u></a>	0..n	<p>This container contains a list of user.</p> <p>For each user the Rte creates ports for requesting communication modes and for notification of mode changes.</p>

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>

**Parameters included**

<a href="#">ComMPncEnabled</a>	0..1
--------------------------------	------

Parameter Name	<b>ComMPncEnabled</b>	
<b>Description</b>	Defines whether in this configuration set the partial networking is enabled.	
	<ul style="list-style-type: none"> <li>▶ TRUE: Enabled</li> <li>▶ FALSE: Disabled</li> </ul>	
<b>Multiplicity</b>	0..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	true	
<b>Configuration class</b>	<a href="#">PostBuild:</a>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

**5.3.1.3. ComMChannel****Containers included**

Container name	Multiplicity	Description
<a href="#">ComMNetworkManagement</a>	1..1	<b>Label:</b> Network Management Configuration This container contains the configuration parameters of the networkmanagement.
<a href="#">ComMUserPerChannel</a>	0..n	This container contains a list of identifiers that are needed to refer to a user in the system which is linked to a channel.

**Parameters included**

Parameter name	Multiplicity
<a href="#">ComMBusType</a>	1..1
<a href="#">ComMChannelId</a>	1..1
<a href="#">ComMFullCommRequestNotificationEnabled</a>	1..1
<a href="#">ComMGlobalNvmBlockDescriptor</a>	1..1
<a href="#">ComMMainFunctionPeriod</a>	1..1
<a href="#">ComMNoCom</a>	1..1
<a href="#">ComMNoWakeup</a>	1..1
<a href="#">ComMPncGatewayType</a>	0..1
<a href="#">ComMManageReference</a>	0..1



<b>Parameter Name</b>	<b>ComMBusType</b>	
<b>Label</b>	Bus Type	
<b>Description</b>	<p>This parameter identifies the bus type of the network channel.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ If the bus type of the network channel is set to <i>COMM_BUS_TYPE_INTERNAL</i> the Network Management Variant has to be set to <i>NONE</i> and if the bus type of the network channel is set to <i>COMM_BUS_TYPE_LIN</i> the Network Management Variant has to be set to <i>LIGHT</i></li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	<i>COMM_BUS_TYPE_CAN</i>	
<b>Range</b>	<p><i>COMM_BUS_TYPE_CAN</i></p> <p><i>COMM_BUS_TYPE_ETH</i></p> <p><i>COMM_BUS_TYPE_FR</i></p> <p><i>COMM_BUS_TYPE_INTERNAL</i></p> <p><i>COMM_BUS_TYPE_LIN</i></p>	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ComMChannelId</b>	
<b>Label</b>	Network Channel ID	
<b>Description</b>	<p>Identification number of the network channel.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ The identifier of the network channel must be zero-based, consecutive and unique.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ComMFullCommRequestNotificationEnabled</b>	
<b>Description</b>	Defines if the optional SenderReceiver Port of Interface <i>ComM_CurrentChannelRequest</i> will be provided for the network channel.	



	<ul style="list-style-type: none"> <li>▶ TRUE: Enabled</li> <li>▶ FALSE: Disabled</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>ComMGlobalNvmBlockDescriptor</b>	
<b>Label</b>	Enable Storage for NoWakeUp Inhibition	
<b>Description</b>	<p>If this parameter is set to "true", the NoWakeUp inhibition state of the channel shall be stored (in some implementation specific way) in the block pointed to by ComMGlobalNvmBlockDescriptor.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ If the parameter is set to true, a valid Nvm block reference must be given in the (existing, i.e. multiplicity 1) ComMGlobalNvmBlockDescriptor pointing to a sufficiently big Nvm block.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild	
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ComMMainFunctionPeriod</b>	
<b>Label</b>	Main Function Period	
<b>Description</b>	<p>Specifies the period the MainFunction has to be triggered in seconds.</p> <p>ComM scheduling shall be at least as fast as the communication stack and a schedule longer than 100ms makes no sense for communication.</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.020	
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild	
<b>Origin</b>	AUTOSAR_ECUC	



<b>Parameter Name</b>	<b>ComMNoCom</b>	
<b>Label</b>	No Change To Communication	
<b>Description</b>	<p>If this parameter is enabled it is not possible to switch the mode of the network channel to <i>Full Communication</i> or <i>Silent Communication</i>.</p> <p>This parameter just defines the initial value. The limitation can be modified by calling <code>ComM_LimitChannelToNoComMode()</code>.</p> <ul style="list-style-type: none"> <li>▶ TRUE: Enabled - Not allowed to switch to Communication Modes above.</li> <li>▶ FALSE: Disabled - Allowed to switch Communication Modes above.</li> </ul> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ Mode Limitation must be enabled.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ComMNoWakeUp</b>	
<b>Label</b>	Prevent Channel Wake Up	
<b>Description</b>	<p>Defines if an ECU is not allowed to wake-up the network channel.</p> <ul style="list-style-type: none"> <li>▶ TRUE: Enabled (not allowed to wake-up)</li> <li>▶ FALSE: Disabled.</li> </ul> <p>This is the default/init value of the runtime variable that can be changed during runtime using <code>ComM_PreventWakeUp()</code>.</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ComMPncGatewayType</b>	
<b>Label</b>	Partial Network Gateway Type	
<b>Description</b>	Identifies the Partial Network Gateway behaviour of a ComMChannel.	



<b>Multiplicity</b>	0..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	COMM_GATEWAY_TYPE_ACTIVE	
<b>Range</b>	COMM_GATEWAY_TYPE_ACTIVE COMM_GATEWAY_TYPE_PASSIVE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ComMManageReference</b>	
<b>Description</b>	Reference to the managing channel.  ► ImplementationType: COMM_ChannelType	
<b>Multiplicity</b>	0..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.3.1.4. ComMNetworkManagement

<b>Parameters included</b>		
<b>Parameter name</b>		<b>Multiplicity</b>
<a href="#">ComMNmLightTimeout</a>		0..1
<a href="#">ComMNmVariant</a>		1..1
<a href="#">ComMPncNmRequest</a>		1..1

<b>Parameter Name</b>	<b>ComMNmLightTimeout</b>
<b>Label</b>	Network Management Light Timeout
<b>Description</b>	<p>Defines the timeout (in seconds) after state <i>ready sleep</i> is left.</p> <p>Range: greater than 0 and less than or equal to 255</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>► Even though the AUTOSAR Specification allows a minimum value of 0, the possible minimum value has been restricted to 0.001 which is also the minimum value of ComMMainFunctionPeriod. Rationale: A timeout value config-</li> </ul>



	ured to be less than the value of the Main Function Period cannot really be ensured.
	<ul style="list-style-type: none"> <li>▶ The value of this parameter must be greater or equal the value of ComM-MainFunctionPeriod.</li> </ul>
<b>Multiplicity</b>	0..1
<b>Type</b>	FLOAT
<b>Default value</b>	10.000
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU

<b>Parameter Name</b>	<b>ComMNmVariant</b>	
<b>Label</b>	Network Management Variant	
<b>Description</b>	<p>Defines the functionality of the networkmanagement. Shall be harmonized with NM configuration.</p> <ul style="list-style-type: none"> <li>▶ NONE : No NM available</li> <li>▶ LIGHT : No AUTOSAR NM available but functionality to shut down a network channel (not synchronized).</li> <li>▶ PASSIVE : AUTOSAR NM available but ECU is not allowed to keep the bus awake.</li> <li>▶ FULL : AUTOSAR NM available (default)</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	FULL	
<b>Range</b>	FULL LIGHT NONE PASSIVE SLAVE_ACTIVE SLAVE_PASSIVE	
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild	
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>ComMPncNmRequest</b>	
<b>Label</b>	Pnc Network Request	



<b>Description</b>	If this parameter equals true then every time a FULL Communication is requested due to a change in the PNC state machine to PNC_REQUESTED Nm shall be called using the API Nm_NetworkRequest.  Dependencies: <ul style="list-style-type: none"><li>▶ Channel is not limited to COMM_NO_COM. Refer &lt;function&gt;ComM_LimitChannelToNoComMode()&lt;/function&gt;.</li><li>▶ Wake up is not inhibited. Refer parameter ComMWakeupInhibitionEnabled</li></ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.3.1.5. ComMUserPerChannel

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">ComMUserChannel</a>	1..1
<b>Parameter Name</b>	
<b>Parameter Name</b>	<b>ComMUserChannel</b>
<b>Label</b>	User Reference
<b>Description</b>	Reference to the ComMUser that corresponds to this channel user.  ▶ ImplementationType: COMM_UserHandleType
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.3.1.6. ComMPnc

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>

**Containers included**

<a href="#">ComMPncComSignal</a>	0..n	Represents the PncComSignals which are used to communicate the EIRA and ERA status of this PNC.
----------------------------------	------	---

**Parameters included**

Parameter name	Multiplicity
<a href="#">ComMPnclId</a>	1..1
<a href="#">ComMChannelPerPnc</a>	1..n
<a href="#">ComMUserPerPnc</a>	0..n
<a href="#">ComMPncEthIfSwitchPortGroupRef</a>	0..255

<b>Parameter Name</b>	<b>ComMPnclId</b>	
<b>Label</b>	PNC ID	
<b>Description</b>	Partial network cluster identification number.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ComMChannelPerPnc</b>	
<b>Description</b>	Reference to the ComMChannel that is required for this PNC. ► ImplementationType: COMM_ChannelType	
<b>Multiplicity</b>	1..n	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ComMUserPerPnc</b>	
<b>Label</b>	User Per PNC	
<b>Description</b>	Reference to the ComMUsers that correspond to this PNC. ► ImplementationType: COMM_UserHandleType	
<b>Multiplicity</b>	0..n	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild



<b>Origin</b>	AUTOSAR_ECUC
<b>Parameter Name</b>	<b>ComMPncEthIfSwitchPortGroupRef</b>
<b>Label</b>	EthIf Switch port group references
<b>Description</b>	Reference to the PortGroups that correspond to this PNC
<b>Multiplicity</b>	0..255
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.3.1.7. ComMPncComSignal

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">ComMPncComSignalDirection</a>	1..1
<a href="#">ComMPncComSignalKind</a>	1..1
<a href="#">ComMPncComSignalChannelRef</a>	1..1
<a href="#">ComMPncComSignalRef</a>	1..1

<b>Parameter Name</b>	<b>ComMPncComSignalDirection</b>
<b>Label</b>	PNC Com Signal Direction
<b>Description</b>	Indicates the communication direction of this PncComSignal.
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Range</b>	RX TX
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>ComMPncComSignalKind</b>
<b>Label</b>	PNC Com Signal Kind
<b>Description</b>	Indicates whether this PncComSignal represents EIRA or ERA PNC information.
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION



<b>Default value</b>	EIRA
<b>Range</b>	EIRA
	ERA
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>ComMPncComSignalChannelRef</b>
<b>Label</b>	PNC Com Signal Channel Reference
<b>Description</b>	Reference to the ComMChannel which is used to determine whether this Pnc-ComSignal shall participate in the active or passive role (via the parameter ComMPncGatewayType of the ComMChannel). Not applicable if ComMPnc-ComSignalKind is EIRA.  Dependencies: <ul style="list-style-type: none"><li>▶ The parameter ComMPncComSignalKind must be set to 'ERA'.</li><li>▶ The parameter ComMPncGatewayEnabled must be set to 'true'.</li></ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>ComMPncComSignalRef</b>
<b>Label</b>	PNC Com Signal Reference
<b>Description</b>	Reference to the ComSignal which is used to transport the partial network channel request information.
<b>Multiplicity</b>	1..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.3.1.8. ComMUser

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>

**Parameters included**

<a href="#">ComMUserIdentifier</a>	1..1
<a href="#">ComMUserEcucPartitionRef</a>	0..1

<b>Parameter Name</b>	<b>ComMUserIdentifier</b>	
<b>Label</b>	User Identifier	
<b>Description</b>	Numeric identifier of one user of the Communication Manager.  Dependencies:  ▶ The user identifier must be zero-based and consecutive.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ComMUserEcucPartitionRef</b>	
<b>Description</b>	Denotes in which "EcucPartition" the requester is executed. When the partition is stopped, the communication request shall be cancelled in the ComM to avoid a stay-aware situation of the bus due to a stopped partition.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

**5.3.1.9. ComMGeneral**

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">ComMServiceAPI</a>	1..1	<b>Label:</b> Service API Parameters  Container for configuration of the service API of ComM.

**Parameters included**

<b>Parameter name</b>	<b>Multiplicity</b>
-----------------------	---------------------



### Parameters included

<a href="#">ComMMultiCoreSupport</a>	1..1
<a href="#">ComMMasterCoreEcuCPartitionRef</a>	1..1
<a href="#">ComMDcmUsage</a>	1..1
<a href="#">ComM0PncVectorAvoidance</a>	1..1
<a href="#">ComMDevErrorDetect</a>	1..1
<a href="#">ComMDirectUserMapping</a>	0..1
<a href="#">ComMEcuGroupClassification</a>	1..1
<a href="#">ComMModeLimitationEnabled</a>	1..1
<a href="#">ComMPncGatewayEnabled</a>	0..1
<a href="#">ComMPncPrepareSleepTimer</a>	0..1
<a href="#">ComMPncSupport</a>	1..1
<a href="#">ComMPnInfoOffset</a>	0..1
<a href="#">ComMRelocatablePbcfgEnable</a>	1..1
<a href="#">ComMRteUsage</a>	1..1
<a href="#">ComMSynchronousWakeUp</a>	1..1
<a href="#">ComMTMinFullComModeDuration</a>	1..1
<a href="#">ComMVersionInfoApi</a>	1..1
<a href="#">ComMWakeupInhibitionEnabled</a>	1..1
<a href="#">ComMGlobalNvMBlockDescriptor</a>	0..1
<a href="#">ComMResetAfterForcingNoComm</a>	1..1
<a href="#">ComMReleaseNoneChannels</a>	1..1
<a href="#">ComMPncReleaseChannel</a>	1..1
<a href="#">ComMDefineOrder</a>	1..1

<b>Parameter Name</b>	<a href="#">ComMMultiCoreSupport</a>	
<b>Label</b>	ComM multicore support	
<b>Description</b>	Enables MultiCoreSupport.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<a href="#">VariantPostBuild:</a>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	



<b>Parameter Name</b>	<b>ComMMasterCoreEcuCPartitionRef</b>	
<b>Label</b>	Master Core EcuC Partition Reference	
<b>Description</b>	Master Core EcuC Partition Ref	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ComMDcmUsage</b>	
<b>Label</b>	Enable Dcm Usage	
<b>Description</b>	<p>This parameter enables the usage of the Dcm. In case this parameter is set to false:</p> <ul style="list-style-type: none"> <li>▶ ComM does not provide the API functions <code>ComM_DCM_ActiveDiagnostic()</code> and <code>ComM_DCM_InactiveDiagnostic</code></li> <li>▶ ComM does not call the API functions <code>Dcm_ComM_FullComModeEntered()</code>, <code>Dcm_ComM_SilentComModeEntered()</code> and <code>Dcm_ComM_NoComModeEntered()</code></li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>ComM0PncVectorAvoidance</b>	
<b>Label</b>	Enable ComM0PncVectorAvoidance	
<b>Description</b>	<p>Avoids sending of 0-PNC-Vectors in case ComMPncGatewayEnabled is enabled</p> <ul style="list-style-type: none"> <li>▶ TRUE: Enabled</li> <li>▶ FALSE: Disabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ComMDevErrorDetect</b>	



<b>Label</b>	Enable Development Error Detection	
<b>Description</b>	Switches the Development Error Detection and Notification ON or OFF. ▶ TRUE: Enabled ▶ FALSE: Disabled	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	true	
<b>Configuration class</b>	VariantPostBuild:	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	
<b>Parameter Name</b>	<b>ComMDirectUserMapping</b>	
<b>Label</b>	Direct User Mapping	
<b>Description</b>	<p>If this parameter is set to true the configuration tool shall automatically create a ComMUser per ComMPnc and a ComMUser per ComMChannel.</p> <p>The shortName of the generated ComMUsers shall follow the following naming convention:</p> <ul style="list-style-type: none"> <li>▶ PNCUser_ComMPncId, e.g. PNCUser_13</li> <li>▶ ChannelUser_ComMChannelId, e.g. ChannelUser_25</li> </ul> <p>Restriction: ComMUser, which are created due to this configuration parameter, shall not be used by SWCs (only available for BswM).</p>	
<b>Multiplicity</b>	0..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	PreCompile:	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	
<b>Parameter Name</b>	<b>ComMEcuGroupClassification</b>	
<b>Label</b>	ECU Group Classification	
<b>Description</b>	<p>This parameter is interpreted as a bitfield where each bit enables/disables a mode inhibition.</p> <ul style="list-style-type: none"> <li>▶ Bit 0: Wake Up inhibition active</li> <li>▶ Bit 1: Limit to <i>No Communication</i> mode</li> </ul> <p>For Example:</p>	



	00000011: Wake up inhibition and limitation to <i>No Communication</i> mode active  Dependencies:  ► Mode Limitation must be enabled.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	3
<b>Range</b>	<=255  >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>ComMModeLimitationEnabled</b>
<b>Label</b>	Mode Limitation
<b>Description</b>	This parameter enables the mode limitation functionality.  ► TRUE: Enabled  ► FALSE: Disabled
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>ComMPncGatewayEnabled</b>
<b>Label</b>	Support for Partial Network Gateway
<b>Description</b>	Enables or disables support of Partial Network Gateway.  ► False: Partial Networking Gateway is disabled  ► True: Partial Networking Gateway is enabled
<b>Multiplicity</b>	0..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC



<b>Parameter Name</b>	<b>ComMPncPrepareSleepTimer</b>	
<b>Label</b>	PNC PrepareSleep Timer	
<b>Description</b>	Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Range</b>	<=63.0 >=0.0	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ComMPncSupport</b>	
<b>Label</b>	Support for Partial Network Cluster (PNC)	
<b>Description</b>	Enables or disables support of partial networking.  ▶ False: Partial Networking is disabled ▶ True: Partial Networking is enabled	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ComMPnInfoOffset</b>	
<b>Label</b>	PnInfoOffset	
<b>Description</b>	Position of partial network information inside NM frame.  Ex. For a NM frame CBV BYTE1 BYTE2 ...BYTE7.  ▶ A value 1 for ComMPnInfoOffset, lead to interpreting BYTE1 as holding PN ID 8 to 15. ▶ A value 2 for ComMPnInfoOffset, lead to interpreting BYTE2 as holdign PN ID 16 to 23.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	1	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild



<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

<b>Parameter Name</b>	<b>ComMRelocatablePbcfgEnable</b>	
<b>Label</b>	ComMRelocatablePbcfgEnable	
<b>Description</b>	<p>Enables/disables support for relocatable postbuild configuration.</p> <ul style="list-style-type: none"> <li>▶ True: Postbuild configuration relocatable in memory.</li> <li>▶ False: Postbuild configuration not relocatable in memory.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	true	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>ComMRteUsage</b>	
<b>Label</b>	Enable Rte Usage	
<b>Description</b>	<p>This parameter enables the usage of the RTE for this module.</p> <p>For an easy integration it is recommended to disable the usage of the RTE at the beginning of the integration work.</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>ComMSynchronousWakeUp</b>	
<b>Label</b>	Synchronous Wake Up	
<b>Description</b>	<p>If this parameter is enabled, a wake up of one channel shall lead to a wake up of all network channels.</p> <ul style="list-style-type: none"> <li>▶ TRUE: Enabled</li> <li>▶ FALSE: Disabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	true	



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>ComMTMinFullComModeDuration</b>	
<b>Label</b>	Minimum Full Communication Mode Duration Time	
<b>Description</b>	Minimum time duration in seconds the ComM spends in the Full Communication mode.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	5	
<b>Range</b>	<=65.000 >=0.001	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>ComMVersionInfoApi</b>	
<b>Label</b>	Enable Version Info API	
<b>Description</b>	Switches the possibility to read the published information with the service ComM_GetPublishedInformation().  ► TRUE: Enabled ► FALSE: Disabled	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>ComMWakeupInhibitionEnabled</b>	
<b>Label</b>	Wakeup Inhibition	
<b>Description</b>	Enable wake up inhibition functionality.  ► TRUE: Enabled ► FALSE: Disabled	
<b>Multiplicity</b>	1..1	



<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>ComMGlobalNvMBlockDescriptor</b>
<b>Label</b>	Global NVRAM Block Descriptor
<b>Description</b>	<p>Reference to NVRAM block containing the non volatile data.</p> <p>Index of the NvMBlockDescriptor will be used by the ComM module for the identification of the NvM block</p>
<b>Multiplicity</b>	0..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>ComMResetAfterForcingNoComm</b>
<b>Label</b>	Reset enable
<b>Description</b>	<p>This parameter enables resetting ECU when entering "No Communication" mode because of an active mode limitation to "No Communication" mode.</p> <ul style="list-style-type: none"> <li>▶ TRUE: Enabled</li> <li>▶ FALSE: Disabled</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>ComMReleaseNoneChannels</b>
<b>Label</b>	ComMReleaseNoneChannels
<b>Description</b>	<p>Specifies if all channels with attribute ComMBusType set to COMM_BUS_TYPE_INTERNAL shall switch from Ready Sleep state to No Communication.</p> <ul style="list-style-type: none"> <li>▶ If ComMBusType = (COMM_BUS_TYPE_CAN or COMM_BUS_TYPE_FR or COMM_BUS_TYPE_LIN or COMM_BUS_TYPE_ETH) and ComMNmVariant = NONE, state transition from Ready Sleep state to No</li> </ul>



	<p>Communication is not performed. The transition from Ready Sleep state to No Communication state is performed only in case ComMBusType = COMM_BUS_TYPE_INTERNAL.</p> <ul style="list-style-type: none"> <li>▶ FALSE When entering ready sleep state all channels with ComMNmVariant = NONE go immediately to No Communication.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>ComMPncReleaseChannel</b>	
<b>Label</b>	ComMPncReleaseChannel	
<b>Description</b>	<p>Configures when the Full Communication requests shall be released.</p> <p>When this parameter is set to COMM_NO_INTERNAL_AND_EXTERNAL_REQUESTS, all Full Communication requests shall be released in the PNC_NO_COMMUNICATION state (Autosar 4.1.1 and Autosar 3.2.3).</p> <p>When this parameter is set to COMM_NO_INTERNAL_REQUESTS, all Full Communication requests shall be released after leaving the PNC_REQUESTED state (Autosar 4.2.1, see Rfc #63643).</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	COMM_NO_INTERNAL_REQUESTS	
<b>Range</b>	<p>C<small>OMM</small>_N<small>O</small>_I<small>N</small>T<small>E</small>R<small>A</small>L_<small>R</small>E<small>Q</small>U<small>E</small>S<small>T</small>S</p> <p>C<small>OMM</small>_N<small>O</small>_I<small>N</small>T<small>E</small>R<small>A</small>L_<small>R</small>E<small>Q</small>U<small>E</small>S<small>T</small>S AND C<small>OMM</small>_N<small>O</small>_E<small>X</small>T<small>E</small>R<small>N</small>A<small>L_<small>R</small>E<small>Q</small>U<small>E</small>S<small>T</small>S</small></p>	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>ComMDefineOrder</b>	
<b>Label</b>	ComMDefineOrder	
<b>Description</b>	<p>Defines the order to be used for mode declaration group ComMode.</p> <ul style="list-style-type: none"> <li>▶ EXPLICIT_ORDER - Explicit order is used.</li> <li>▶ ALPHABETIC_ORDER - Alphabetical order is used.</li> </ul>	
<b>Multiplicity</b>	1..1	



Type	ENUMERATION	
Default value	EXPLICIT_ORDER	
Range	EXPLICIT_ORDER ALPHABETIC_ORDER	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

### 5.3.1.10. ComMServiceAPI

Parameters included	
Parameter name	Multiplicity
<a href="#">ComMEnableASR32ServiceAPI</a>	1..1
<a href="#">ComMEnableASR40ServiceAPI</a>	1..1
<a href="#">ComMDefaultASRServiceAPI</a>	1..1

Parameter Name	<b>ComMEnableASR32ServiceAPI</b>	
Label	Enable AUTOSAR 3.2 service API	
Description	<p>Configures whether the AUTOSAR 3.2 service API shall be provided.</p> <ul style="list-style-type: none"> <li>▶ TRUE = Enables AUTOSAR 3.2 service API.</li> <li>▶ FALSE = Disables AUTOSAR 3.2 service API.</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>ComMEnableASR40ServiceAPI</b>	
Label	Enable AUTOSAR 4.0 service API	
Description	<p>Configures whether the AUTOSAR 4.0 service API shall be provided.</p> <ul style="list-style-type: none"> <li>▶ TRUE = Enables AUTOSAR 4.0 service API.</li> <li>▶ FALSE = Disables AUTOSAR 4.0 service API.</li> </ul> <p>:</p>	



<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	ComMDefaultASRServiceAPI
<b>Label</b>	Default AUTOSAR service API
<b>Description</b>	<p>Defines the default AUTOSAR service API.</p> <ul style="list-style-type: none"> <li>▶ AUTOSAR_32 = AUTOSAR 3.2 service API is the default one.</li> <li>▶ AUTOSAR_40 = AUTOSAR 4.0 service API is the default one.</li> <li>▶ NONE = No default AUTOSAR service API is provided.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	AUTOSAR_40
<b>Range</b>	AUTOSAR_32 AUTOSAR_40 NONE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.3.1.11. CommonPublishedInformation

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">ArMajorVersion</a>	1..1
<a href="#">ArMinorVersion</a>	1..1
<a href="#">ArPatchVersion</a>	1..1
<a href="#">SwMajorVersion</a>	1..1
<a href="#">SwMinorVersion</a>	1..1
<a href="#">SwPatchVersion</a>	1..1
<a href="#">ModuleId</a>	1..1

**Parameters included**

<a href="#">VendorId</a>	1..1
<a href="#">Release</a>	1..1

<b>Parameter Name</b>	<b>ArMajorVersion</b>
<b>Label</b>	AUTOSAR Major Version
<b>Description</b>	Major version number of AUTOSAR specification on which the appropriate implementation is based on.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	4
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>ArMinorVersion</b>
<b>Label</b>	AUTOSAR Minor Version
<b>Description</b>	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	1
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>ArPatchVersion</b>
<b>Label</b>	AUTOSAR Patch Version
<b>Description</b>	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	1
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SwMajorVersion</b>
-----------------------	-----------------------



<b>Label</b>	Software Major Version
<b>Description</b>	Major version number of the vendor specific implementation of the module.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	5
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SwMinorVersion</b>
<b>Label</b>	Software Minor Version
<b>Description</b>	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	19
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SwPatchVersion</b>
<b>Label</b>	Software Patch Version
<b>Description</b>	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	8
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>ModuleId</b>
<b>Label</b>	Numeric Module ID
<b>Description</b>	Module ID of this module from Module List
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	12



<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>VendorId</b>
<b>Label</b>	Vendor ID
<b>Description</b>	Vendor ID of the dedicated implementation of this module according to the AU-TOSAR vendor list
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	1
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>Release</b>
<b>Label</b>	Release Information
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING_LABEL
<b>Default value</b>	
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.3.1.12. PublishedInformation

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
PbcfgMSupport	1..1

<b>Parameter Name</b>	<b>PbcfgMSupport</b>
<b>Label</b>	PbcfgM support
<b>Description</b>	Specifies whether or not the ComM can use the PbcfgM module for post-build support.
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	true



<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

## 5.3.2. Application programming interface (API)

### 5.3.2.1. Type definitions

#### 5.3.2.1.1. ComM\_ASR32\_InhibitionStatusType

<b>Purpose</b>	Inhibition status of ComM.
<b>Type</b>	uint8
<b>Description</b>	<p>Defines whether a mode inhibition is active or not.</p> <ul style="list-style-type: none"><li>▶ Bit 0 (LSB): Wake Up inhibition active.</li><li>▶ Bit 1 (MSB): Limit to "No Communication" mode.</li></ul>

#### 5.3.2.1.2. ComM\_ASR32\_ModeType

<b>Purpose</b>	Current mode of the Communication Manager (main state of the state machine).
<b>Type</b>	uint8
<b>Description</b>	If Rte usage is enabled in the ComM, the type ComM_ModeType is defined in Rte_Types.h and is made public indirectly by including Rte_Types.h in ComM.h.

#### 5.3.2.1.3. ComM\_ASR32\_UserHandleType

<b>Purpose</b>	Type for user of the communication manager.
<b>Type</b>	uint8
<b>Description</b>	<p>Wraps the ComM_UserHandleType.</p> <p>Range:</p> <ul style="list-style-type: none"><li>▶ 0 ... 255</li></ul>



#### 5.3.2.1.4. ComM\_ASR40\_InhibitionStatusType

<b>Purpose</b>	Inhibition status of ComM.
<b>Type</b>	uint8
<b>Description</b>	<p>Defines whether a mode inhibition is active or not.</p> <ul style="list-style-type: none"> <li>▶ Bit 0 (LSB): Wake Up inhibition active.</li> <li>▶ Bit 1 (MSB): Limit to "No Communication" mode.</li> </ul>

#### 5.3.2.1.5. ComM\_ASR40\_ModeType

<b>Purpose</b>	Current mode of the Communication Manager (main state of the state machine).
<b>Type</b>	uint8
<b>Description</b>	If Rte usage is enabled in the ComM, the type ComM_ModeType is defined in Rte_Types.h and is made public indirectly by including Rte_Types.h in ComM.h.

#### 5.3.2.1.6. ComM\_ASR40\_UserHandleType

<b>Purpose</b>	Type for user of the communication manager.
<b>Type</b>	uint8
<b>Description</b>	<p>Wraps the ComM_UserHandleType.</p> <p>Range:</p> <ul style="list-style-type: none"> <li>▶ 0 ... 255</li> </ul>

#### 5.3.2.1.7. ComM\_BusSM\_RequestComMode\_FctPtr

<b>Purpose</b>	Function pointer type for BusSM Request.
<b>Type</b>	Std_ReturnType (*) (uint8, uint8)

#### 5.3.2.1.8. ComM\_Dcm\_Notify\_FctPtr

<b>Purpose</b>	Function pointer type for Dcm notifications.
----------------	--



Type	Std_ReturnType (*) (uint8)
------	----------------------------

### 5.3.2.1.9. ComM\_Dcm\_Notif\_Type

Purpose	Type for Comm to Dcm notification function pointers.	
Type	struct	
Members	ComM_Dcm_Notif_FctPtr FullCom-Notif	
	ComM_Dcm_Notif_FctPtr Silent-ComNotif	
	ComM_Dcm_Notif_FctPtr NoComNo-Notif	

### 5.3.2.1.10. ComM\_InhibitionStatusType

Purpose	Inhibition Status Type.
Type	<a href="#">ComM_AS40_InhibitionStatusType</a>
Description	Provide AUTOSAR 4.0 InhibitionStatusType as default to other BSW modules

### 5.3.2.1.11. ComM\_InitStatusType

Purpose	Initialization status of ComM.
Type	uint8

### 5.3.2.1.12. ComM\_ModeType

Purpose	Mode Type.
Type	<a href="#">ComM_AS40_ModeType</a>
Description	Provide AUTOSAR 4.0 ComM_ModeType as default to other BSW modules

### 5.3.2.1.13. ComM\_Multicore\_SchmSend\_FctPtr

Purpose	Function pointer type Master Core send signal.
---------	--



Type	Std_ReturnType(*) (const uint8 *data)
------	---------------------------------------

#### 5.3.2.1.14. ComM\_NvDataType

Purpose	Inhibition status of ComM to be stored in NvM.	
Type	struct	
Members	uint16 InhibitionCounter	number of rejected Full ComMode request
	ComM_ASR40_InhibitionStatusType ECUGroupClassification	Current ECU group classification.
	uint8 ChannelWakeUpInhibition	Defines whether a WakeUp inhibition is active or not to Persistently stored.
		This array is derived by calculating the number of channels configured in ComM. Each bit corresponds to a channel. E.g : If there are more than 8 channels, size of COMM_NUM_BYTES_NVM will greater than 1

#### 5.3.2.1.15. ComM\_PncModeType

Purpose	Current mode of a PNC.
Type	uint8

#### 5.3.2.1.16. ComM\_RxSignal\_Struct\_Type

Purpose	Type for Rx signals configuration.	
Type	struct	
Members	Com_SignalIdType ComHandleId	

#### 5.3.2.1.17. ComM\_StateType

Purpose	State and sub-state of ComM state machine.
Type	uint8



<b>Description</b>	Range: 0 : COMM_NO_COM_NO_PENDING_REQUEST 1 : COMM_NO_COM_REQUEST_PENDING 2 : COMM_FULL_COM_NETWORK_REQUESTED 3 : COMM_FULL_COM_READY_SLEEP 4 : COMM_SILENT_COM
--------------------	---

### 5.3.2.1.18. ComM\_TxSignal\_Struct\_Type

<b>Purpose</b>	Type for Tx signals configuration.
<b>Type</b>	struct
<b>Members</b>	Com_SignalIdType ComHandleId
	uint8 TxSignalKind

### 5.3.2.1.19. ComM\_UserHandleType

<b>Purpose</b>	User Handle Type.
<b>Type</b>	<a href="#">ComM_ASR40_UserHandleType</a>
<b>Description</b>	Provide AUTOSAR 4.0 UserHandleType as default to other BSW modules

## 5.3.2.2. Macro constants

### 5.3.2.2.1. COMM\_AR\_RELEASE\_MAJOR\_VERSION

<b>Purpose</b>	AUTOSAR release major version.
<b>Value</b>	4U

### 5.3.2.2.2. COMM\_AR\_RELEASE\_MINOR\_VERSION

<b>Purpose</b>	AUTOSAR release minor version.
<b>Value</b>	1U

### 5.3.2.2.3. COMM\_AR\_RELEASE\_REVISION\_VERSION

<b>Purpose</b>	AUTOSAR release revision version.
----------------	-----------------------------------



<b>Value</b>	3U
--------------	----

#### 5.3.2.2.4. COMM\_BUS\_TYPE\_CAN

<b>Purpose</b>	Bus types used as values for COMM_BUSTYPE_OF_CHANNEL().
<b>Value</b>	0U

#### 5.3.2.2.5. COMM\_BUS\_TYPE\_ETH

<b>Purpose</b>	
<b>Value</b>	1U

#### 5.3.2.2.6. COMM\_BUS\_TYPE\_FR

<b>Purpose</b>	
<b>Value</b>	2U

#### 5.3.2.2.7. COMM\_BUS\_TYPE\_INTERNAL

<b>Purpose</b>	
<b>Value</b>	3U

#### 5.3.2.2.8. COMM\_BUS\_TYPE\_LIN

<b>Purpose</b>	
<b>Value</b>	4U

#### 5.3.2.2.9. COMM\_DET\_REPORT\_ERROR

<b>Purpose</b>	Macro for reporting an error to DET.
<b>Value</b>	((void)Det_ReportError(COMM_MODULE_ID, (InstanceId), (Apild), (ErrorId)))



#### 5.3.2.2.10. COMM\_EIRA\_ERA\_ACTIVE

Purpose	
Value	0U

#### 5.3.2.2.11. COMM\_ERA\_PASSIVE

Purpose	
Value	1U

#### 5.3.2.2.12. COMM\_E\_BUSSM\_REQUESTCOMMODE

Purpose	Master BusSm Request Com Mode error Id.
Value	246U

#### 5.3.2.2.13. COMM\_E\_DCM\_COMM\_FULLCOMMODEENTERED

Purpose	Dcm Full Com Entered Notification error Id.
Value	245U

#### 5.3.2.2.14. COMM\_E\_DCM\_COMM\_NOCOMMODEENTERED

Purpose	Dcm No Com Entered Notification error Id.
Value	243U

#### 5.3.2.2.15. COMM\_E\_DCM\_COMM\_SILENTCOMMODEENTERED

Purpose	Dcm Silent Com Entered Notification error Id.
Value	244U

#### 5.3.2.2.16. COMM\_E\_MASTER\_DATA\_GET

Purpose	Master Data Get error Id.
---------	---------------------------



<b>Value</b>	247U
--------------	------

#### **5.3.2.2.17. COMM\_E\_MASTER\_DATA\_SET**

<b>Purpose</b>	Master Data Set error Id.
<b>Value</b>	248U

#### **5.3.2.2.18. COMM\_E\_MODE\_LIMITATION**

<b>Purpose</b>	Function call has been successfully but mode can not be granted because of mode inhibition.
<b>Value</b>	0x02U

#### **5.3.2.2.19. COMM\_E\_MULTIPLE\_PNC\_ASSIGNED**

<b>Purpose</b>	Return value when ComM when multiple PNCs are assigned to a user.
<b>Value</b>	3U

#### **5.3.2.2.20. COMM\_E\_NOT\_INITED**

<b>Purpose</b>	Error Code.
<b>Value</b>	1U
<b>Description</b>	API service used without module initialization.

#### **5.3.2.2.21. COMM\_E\_NO\_PNC\_ASSIGNED**

<b>Purpose</b>	Return value when ComM when no PNCs are assigned to a user.
<b>Value</b>	4U

#### **5.3.2.2.22. COMM\_E\_SLAVE\_DATA\_GET**

<b>Purpose</b>	Slave Data Get error Id.
----------------	--------------------------



<b>Value</b>	249U
--------------	------

#### 5.3.2.2.23. COMM\_E\_SLAVE\_DATA\_SET

<b>Purpose</b>	Slave Data Set error Id.
<b>Value</b>	250U

#### 5.3.2.2.24. COMM\_E\_UNINIT

<b>Purpose</b>	Return value when ComM not initialized.
<b>Value</b>	3U

#### 5.3.2.2.25. COMM\_E\_WRONG\_PARAMETERS

<b>Purpose</b>	Error Code.
<b>Value</b>	2U
<b>Description</b>	API service used with wrong parameters, e.g. a NULL pointer.

#### 5.3.2.2.26. COMM\_FULL\_COMMUNICATION

<b>Purpose</b>	Full Communication mode value for <a href="#">ComM_ModeType</a> .
<b>Value</b>	2U

#### 5.3.2.2.27. COMM\_FULL\_COM\_NETWORK\_REQUESTED

<b>Purpose</b>	Full_Com_Network_Requested value for <a href="#">ComM_StateType</a> .
<b>Value</b>	2U

#### 5.3.2.2.28. COMM\_FULL\_COM\_READY\_SLEEP

<b>Purpose</b>	Full_Com_Ready_Sleep value for <a href="#">ComM_StateType</a> .
----------------	---



<b>Value</b>	3U
--------------	----

#### 5.3.2.2.29. COMM\_INIT

<b>Purpose</b>	COM Manager is initialized and usable.
<b>Value</b>	1U

#### 5.3.2.2.30. COMM\_INSTANCE\_ID

<b>Purpose</b>	Id of instance of ComM.
<b>Value</b>	COMM_MASTER_CORE_ID

#### 5.3.2.2.31. COMM\_LIMIT\_NOCOMM

<b>Purpose</b>	Macros for Communication Inhibition.
<b>Value</b>	2U

#### 5.3.2.2.32. COMM\_LIMIT\_NOCOMM\_MASK

<b>Purpose</b>	
<b>Value</b>	253U

#### 5.3.2.2.33. COMM\_MODULE\_ID

<b>Purpose</b>	AUTOSAR module identification.
<b>Value</b>	12U

#### 5.3.2.2.34. COMM\_NM\_FULL\_VARIANT

<b>Purpose</b>	Nm variant types used for values of COMM_NM_VARIANT_OF_CHANNEL().
<b>Value</b>	0U



#### 5.3.2.2.35. COMM\_NM\_LIGHT\_VARIANT

Purpose	
Value	1U

#### 5.3.2.2.36. COMM\_NM\_NONE\_VARIANT

Purpose	
Value	2U

#### 5.3.2.2.37. COMM\_NM\_PASSIVE\_VARIANT

Purpose	
Value	3U

#### 5.3.2.2.38. COMM\_NM\_SLAVE\_ACTIVE\_VARIANT

Purpose	
Value	4U

#### 5.3.2.2.39. COMM\_NM\_SLAVE\_PASSIVE\_VARIANT

Purpose	
Value	5U

#### 5.3.2.2.40. COMM\_NO\_COMMUNICATION

Purpose	No Communication mode value for <a href="#">ComM_ModeType</a> .
Value	0U

#### 5.3.2.2.41. COMM\_NO\_COM\_NO\_PENDING\_REQUEST

Purpose	No_Com_No_Pending_Request value for <a href="#">ComM_StateType</a> .
---------	--



<b>Value</b>	0U
--------------	----

#### 5.3.2.2.42. COMM\_NO\_COM\_REQUEST\_PENDING

<b>Purpose</b>	No_Com_Pending_Request value for <a href="#">ComM_StateType</a> .
<b>Value</b>	1U

#### 5.3.2.2.43. COMM\_NO\_INTERNAL\_AND\_EXTERNAL\_REQUESTS

<b>Purpose</b>	
<b>Value</b>	0U

#### 5.3.2.2.44. COMM\_NO\_INTERNAL\_REQUESTS

<b>Purpose</b>	
<b>Value</b>	1U

#### 5.3.2.2.45. COMM\_NUM\_CHANNELS\_BYTES

<b>Purpose</b>	
<b>Value</b>	$((COMM\_NUM\_CHANNELS - 1U) / 8U) + 1U)$

#### 5.3.2.2.46. COMM\_NUM\_PNC\_BYTES

<b>Purpose</b>	
<b>Value</b>	$((COMM\_NUM\_PNC - 1U) / 8U) + 1U)$

#### 5.3.2.2.47. COMM\_PREVENT\_WAKEUP

<b>Purpose</b>	
<b>Value</b>	1U



#### 5.3.2.2.48. COMM\_PREVENT\_WAKEUP\_MASK

Purpose	
Value	254U

#### 5.3.2.2.49. COMM\_SID\_BUS\_SM\_BUS\_SLEEP\_MODE

Purpose	AUTOSAR API service ID.
Value	246U
Description	Definition of service ID for <a href="#">ComM_BusSM_BusSleepMode()</a> .

#### 5.3.2.2.50. COMM\_SID\_BUS\_SM\_INDICATION

Purpose	AUTOSAR API service ID.
Value	51U
Description	Definition of service ID for <a href="#">ComM_BusSM_ModelIndication()</a> .

#### 5.3.2.2.51. COMM\_SID\_COMMUNICATIONALLOWED

Purpose	AUTOSAR API service ID.
Value	53U
Description	Definition of service ID for <a href="#">ComM_CommunicationAllowed()</a> .

#### 5.3.2.2.52. COMM\_SID\_DCM\_ACTIVE\_DIAGNOSTIC

Purpose	AUTOSAR API service ID.
Value	31U
Description	Definition of service ID for <a href="#">ComM_DCM_ActiveDiagnostic()</a> .

#### 5.3.2.2.53. COMM\_SID\_DCM\_INACTIVE\_DIAGNOSTIC

Purpose	AUTOSAR API service ID.
---------	-------------------------



<b>Value</b>	32U
<b>Description</b>	Definition of service ID for <a href="#">ComM_DCM_InactiveDiagnostic()</a> .

#### 5.3.2.2.54. COMM\_SID\_DEINIT

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	2U
<b>Description</b>	Definition of service ID for <a href="#">ComM_DeInit()</a> .

#### 5.3.2.2.55. COMM\_SID\_ECUM\_PNC\_WAKE\_UP\_INDICATION

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	55U
<b>Description</b>	Definition of service ID for <a href="#">ComM_EcuM_WakeupIndication()</a> .

#### 5.3.2.2.56. COMM\_SID\_ECUM\_WAKE\_UP\_INDICATION

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	42U
<b>Description</b>	Definition of service ID for <a href="#">ComM_EcuM_WakeupIndication()</a> .

#### 5.3.2.2.57. COMM\_SID\_GET\_CURRENT\_COM\_MODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	8U
<b>Description</b>	Definition of service ID for <a href="#">ComM_GetCurrentComMode()</a> .

#### 5.3.2.2.58. COMM\_SID\_GET\_CURRENT\_PNC\_COM\_MODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	254U
<b>Description</b>	Definition of service ID for <a href="#">ComM_GetCurrentPNCComMode()</a> .



#### 5.3.2.2.59. COMM\_SID\_GET\_INHIBITION\_STATUS

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	4U
<b>Description</b>	Definition of service ID for <a href="#">ComM_GetInhibitionStatus()</a> .

#### 5.3.2.2.60. COMM\_SID\_GET\_MAX\_COM\_MODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	6U
<b>Description</b>	Definition of service ID for <a href="#">ComM_GetMaxComMode()</a> .

#### 5.3.2.2.61. COMM\_SID\_GET\_REQUESTED\_COM\_MODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	7U
<b>Description</b>	Definition of service ID for <a href="#">ComM_GetRequestedComMode()</a> .

#### 5.3.2.2.62. COMM\_SID\_GET\_STATE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	52U
<b>Description</b>	Definition of service ID for <a href="#">ComM_GetState()</a> .

#### 5.3.2.2.63. COMM\_SID\_GET\_STATUS

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	3U
<b>Description</b>	Definition of service ID for <a href="#">ComM_GetStatus()</a> .

#### 5.3.2.2.64. COMM\_SID\_GET\_VERSION\_INFO

<b>Purpose</b>	AUTOSAR API service ID.
----------------	-------------------------



<b>Value</b>	16U
<b>Description</b>	Definition of service ID for <a href="#">ComM_GetVersionInfo()</a> .

#### 5.3.2.2.65. COMM\_SID\_INIT

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	1U
<b>Description</b>	Definition of service ID for <a href="#">ComM_Init()</a> .

#### 5.3.2.2.66. COMM\_SID\_LIMIT\_CHANNEL\_TO\_NO\_COM

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	11U
<b>Description</b>	Definition of service ID for <a href="#">ComM_LimitChannelToNoComMode()</a> .

#### 5.3.2.2.67. COMM\_SID\_LIMIT\_ECU\_TO\_NO\_COM

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	12U
<b>Description</b>	Definition of service ID for <a href="#">ComM_LimitECUToNoComMode()</a> .

#### 5.3.2.2.68. COMM\_SID\_MAIN\_FUNCTION

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x60U
<b>Description</b>	Definition of service ID for <a href="#">ComM_MainFunction_&lt;Channel_ID&gt;()</a> .

#### 5.3.2.2.69. COMM\_SID\_MASTER\_DATA\_GET

<b>Purpose</b>	Master Data Get SID.
<b>Value</b>	247U



#### 5.3.2.2.70. COMM\_SID\_MASTER\_DATA\_SET

<b>Purpose</b>	Master Data Set SID.
<b>Value</b>	248U

#### 5.3.2.2.71. COMM\_SID\_NM\_BUS\_SLEEP\_MODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	26U
<b>Description</b>	Definition of service ID for <a href="#">ComM_Nm_BusSleepMode()</a> .

#### 5.3.2.2.72. COMM\_SID\_NM\_NETWORK\_MODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	24U
<b>Description</b>	Definition of service ID for <a href="#">ComM_Nm_NetworkMode()</a> .

#### 5.3.2.2.73. COMM\_SID\_NM\_NETWORK\_START\_IND

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	21U
<b>Description</b>	Definition of service ID for <a href="#">ComM_Nm_NetworkStartIndication()</a> .

#### 5.3.2.2.74. COMM\_SID\_NM\_PREPARE\_BUS\_SLEEP

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	25U
<b>Description</b>	Definition of service ID for <a href="#">ComM_Nm_PrepBusSleepMode()</a> .

#### 5.3.2.2.75. COMM\_SID\_NM\_RESTART\_IND

<b>Purpose</b>	AUTOSAR API service ID.
----------------	-------------------------



<b>Value</b>	27U
<b>Description</b>	Definition of service ID for <a href="#">ComM_Nm_RestartIndication()</a> .

#### 5.3.2.2.76. COMM\_SID\_PREVENT\_WAKE\_UP

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	9U
<b>Description</b>	Definition of service ID for <a href="#">ComM_PreventWakeUp()</a> .

#### 5.3.2.2.77. COMM\_SID\_READ\_INHIBIT\_COUNTER

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	13U
<b>Description</b>	Definition of service ID for <a href="#">ComM_ReadInhibitCounter()</a> .

#### 5.3.2.2.78. COMM\_SID\_REQUEST\_COM\_MODE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	5U
<b>Description</b>	Definition of service ID for <a href="#">ComM_GetRequestComMode()</a> .

#### 5.3.2.2.79. COMM\_SID\_RESET\_INHIBIT\_COUNTER

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	14U
<b>Description</b>	Definition of service ID for <a href="#">ComM_ResetInhibitCounter()</a> .

#### 5.3.2.2.80. COMM\_SID\_SET\_ECU\_GROUP\_CLASS

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	15U



Description	Definition of service ID for <a href="#">ComM_SetECUGroupClassification()</a> .
-------------	---

#### 5.3.2.2.81. COMM\_SID\_SLAVE\_DATA\_GET

Purpose	Slave Data Get SID.
Value	249U

#### 5.3.2.2.82. COMM\_SID\_SLAVE\_DATA\_SET

Purpose	Slave Data Set SID.
Value	250U

#### 5.3.2.2.83. COMM\_SILENT\_COM

Purpose	Silent Communication value for <a href="#">ComM_StateType</a> .
Value	4U

#### 5.3.2.2.84. COMM\_SILENT\_COMMUNICATION

Purpose	Silent Communication mode value for <a href="#">ComM_ModeType</a> .
Value	1U

#### 5.3.2.2.85. COMM\_SW\_MAJOR\_VERSION

Purpose	AUTOSAR module major version.
Value	5U

#### 5.3.2.2.86. COMM\_SW\_MINOR\_VERSION

Purpose	AUTOSAR module minor version.
Value	19U



#### 5.3.2.2.87. COMM\_SW\_PATCH\_VERSION

<b>Purpose</b>	AUTOSAR module patch version.
<b>Value</b>	8U

#### 5.3.2.2.88. COMM\_UNINIT

<b>Purpose</b>	The COM Manager is not initialized or not usable. Default value after reset.
<b>Value</b>	0U

#### 5.3.2.2.89. COMM\_VENDOR\_ID

<b>Purpose</b>	AUTOSAR vendor identification: Elektrobit Automotive GmbH.
<b>Value</b>	1U

#### 5.3.2.2.90. ComM\_GetCurrentComMode

<b>Purpose</b>	Wrapping macro for ComM_GetCurrentComMode to provide AUTOSAR 4.0 API as default to other BSW modules.
<b>Value</b>	ComM_AS40_GetCurrentComMode

#### 5.3.2.2.91. ComM\_GetCurrentPNCComMode

<b>Purpose</b>	Wrapping macro for ComM_GetCurrentPNCComMode to provide AUTOSAR 4.0 API as default to other BSW modules.
<b>Value</b>	ComM_AS40_GetCurrentPNCComMode

#### 5.3.2.2.92. ComM\_GetInhibitionStatus

<b>Purpose</b>	Wrapping macro for ComM_GetInhibitionStatus to provide AUTOSAR 4.0 API as default to other BSW modules.
<b>Value</b>	ComM_AS40_GetInhibitionStatus



### 5.3.2.2.93. ComM\_GetMaxComMode

<b>Purpose</b>	Wrapping macro for ComM_GetMaxComMode to provide AUTOSAR 4.0 API as default to other BSW modules.
<b>Value</b>	ComM_ASR40_GetMaxComMode

### 5.3.2.2.94. ComM\_GetRequestedComMode

<b>Purpose</b>	Wrapping macro for ComM_GetRequestedComMode to provide AUTOSAR 4.0 API as default to other BSW modules.
<b>Value</b>	ComM_ASR40_GetRequestedComMode

### 5.3.2.2.95. ComM\_LimitChannelToNoComMode

<b>Purpose</b>	Wrapping macro for ComM_LimitChannelToNoComMode to provide AUTOSAR 4.0 API as default to other BSW modules.
<b>Value</b>	ComM_ASR40_LimitChannelToNoComMode

### 5.3.2.2.96. ComM\_LimitECUToNoComMode

<b>Purpose</b>	Wrapping macro for ComM_LimitECUToNoComMode to provide AUTOSAR 4.0 API as default to other BSW modules.
<b>Value</b>	ComM_ASR40_LimitECUToNoComMode

### 5.3.2.2.97. ComM\_PreventWakeUp

<b>Purpose</b>	Wrapping macro for ComM_PreventWakeUp to provide AUTOSAR 4.0 API as default to other BSW modules.
<b>Value</b>	ComM_ASR40_PreventWakeUp

### 5.3.2.2.98. ComM\_ReadInhibitCounter

<b>Purpose</b>	Wrapping macro for ComM_ReadInhibitCounter to provide AUTOSAR 4.0 API as default to other BSW modules.
<b>Value</b>	ComM_ASR40_ReadInhibitCounter



#### 5.3.2.2.99. ComM\_RequestComMode

<b>Purpose</b>	Wrapping macro for ComM_RequestComMode to provide AUTOSAR 4.0 API as default to other BSW modules.
<b>Value</b>	ComM_AS40_RequestComMode

#### 5.3.2.2.100. ComM\_ResetInhibitCounter

<b>Purpose</b>	Wrapping macro for ComM_ResetInhibitCounter to provide AUTOSAR 4.0 API as default to other BSW modules.
<b>Value</b>	ComM_AS40_ResetInhibitCounter

#### 5.3.2.2.101. ComM\_SetECUGroupClassification

<b>Purpose</b>	Wrapping macro for ComM_SetECUGroupClassification to provide AUTOSAR 4.0 API as default to other BSW modules.
<b>Value</b>	ComM_AS40_SetECUGroupClassification

#### 5.3.2.2.102. PNC\_FULL\_COMMUNICATION

<b>Purpose</b>	PNC is able to communicate.
<b>Value</b>	4U

#### 5.3.2.2.103. PNC\_NO\_COMMUNICATION

<b>Purpose</b>	PNC does not communicate.
<b>Value</b>	3U

#### 5.3.2.2.104. PNC\_PREPARE\_SLEEP

<b>Purpose</b>	PNC is active with no deadline monitoring.
<b>Value</b>	2U



#### 5.3.2.2.105. PNC\_READY\_SLEEP

Purpose	PNC is requested by a remote ComM user.
Value	1U

#### 5.3.2.2.106. PNC\_REQUESTED

Purpose	PNC is requested by a local ComM user.
Value	0U

### 5.3.2.3. Objects

#### 5.3.2.3.1. ComM\_BusSmRequestComModeMcFpList

Purpose	
Type	const <a href="#">ComM_BusSM_RequestComMode_FctPtr</a>

#### 5.3.2.3.2. ComM\_BusTypeOfChannel

Purpose	Bus type of channel given by index.
Type	const uint8

#### 5.3.2.3.3. ComM\_ChanState

Purpose	ComM Channel State.
Type	<a href="#">ComM_StateType</a>
Description	This variable holds the current state of the ComM Channel.

#### 5.3.2.3.4. ComM\_ChannellInhibitionStatusInit

Purpose	Default ComMNoCom.
Type	const <a href="#">ComM_ASR40_InhibitionStatusType</a>



<b>Description</b>	Default value configured for ComMNoCom
--------------------	--

### 5.3.2.3.5. ComM\_ChannelMode

<b>Purpose</b>	ComM Channel Mode.
Type	<a href="#">ComM_ASR40_ModeType</a>
<b>Description</b>	This variable holds the current mode of the ComM Channel.

### 5.3.2.3.6. ComM\_ChannelModePrevious

<b>Purpose</b>	ComM Channel Mode.
Type	<a href="#">ComM_ASR40_ModeType</a>
<b>Description</b>	This variable holds the current mode of the ComM Channel before ComM_ClearUser-Request is called.

### 5.3.2.3.7. ComM\_Communication\_Allowed

<b>Purpose</b>	number of bytes needed to store CommunicationAllowed flags
Type	uint8

### 5.3.2.3.8. ComM\_DcmActiveStatus

<b>Purpose</b>	Flag to be set if DCM indicated active communication.
Type	uint8

### 5.3.2.3.9. ComM\_Dcm\_Notify\_FpList

<b>Purpose</b>	
Type	const <a href="#">ComM_Dcm_Notify_Type</a>

### 5.3.2.3.10. ComM\_EB\_NoWakeupNvStorage

<b>Purpose</b>	Persistent storage of WakeUp inhibition status of channels in NvM.
----------------	--



Type	const uint8
------	-------------

#### 5.3.2.3.11. ComM\_EB\_NvRAM

Purpose	
Type	<a href="#">ComM_NvDataType</a>

#### 5.3.2.3.12. ComM\_EB\_NvROM

Purpose	Inhibition status of ComM in ROM.
Type	const <a href="#">ComM_NvDataType</a>
Description	Default Inhibition status of ComM to be stored in NvM

#### 5.3.2.3.13. ComM\_InitStatus

Purpose	ComM Initialization status.
Type	<a href="#">ComM_InitStatusType</a>
Description	This variable holds the current initialization status of the ComM module.

#### 5.3.2.3.14. ComM\_NetReqNoNmTimeoutMs

Purpose	NetReqNoNmTimeout.
Type	const uint16
Description	These variables hold the timeout value for Full Com Network Request state in ms. After this time the Full Com Network Request state is left and the transition to the state "ready sleep" is performed. This timeout is only relevant and used for channels of Nm variant NONE and LIGHT.

#### 5.3.2.3.15. ComM\_NmVariantOfChannel

Purpose	NM variant of channel given by index.
Type	const uint8



#### 5.3.2.3.16. ComM\_PNCMode

<b>Purpose</b>	ComM PNC communication mode.
Type	<a href="#">ComM_ASR40_ModeType</a>
Description	This variable holds the current PNC mode.

#### 5.3.2.3.17. ComM\_PNCRequestedComMode

<b>Purpose</b>	ComM PNC requested mode.
Type	<a href="#">ComM_ASR40_ModeType</a>
Description	This variable holds the current PNC requested mode.

#### 5.3.2.3.18. ComM\_PNCRequestedComMode0PncVectorAvoidance

<b>Purpose</b>	ComM PNC requested mode.
Type	<a href="#">ComM_ASR40_ModeType</a>
Description	This variable holds the current PNC requested mode.

#### 5.3.2.3.19. ComM\_PassiveStartupStatus

<b>Purpose</b>	Flag to be set if Passive startup happens before active request (user, pnc, managed channels).
Type	uint8

#### 5.3.2.3.20. ComM\_PncRxActiveEraMask

<b>Purpose</b>	Bit array of Rx Era Com signal masks.
Type	const uint8 *const
Description	A PNC may not have references to all RX ERA Com signals. Therefore, the bits corresponding to PNCs which do not reference a Com signal have to be masked out from the newly received ERA before being merged to the existing ERA bits. This array contains references to bit arrays for each RX ERA Com signal. If a Com Signal is associated to a PNC, then the bit corresponding to that PNC in the bit array will be set to 1.



	If a Com Signal is not associated to a PNC, then the corresponding bit in the bit array will be set to 0.
--	---

### 5.3.2.3.21. ComM\_PncRxEiraMask

<b>Purpose</b>	Bit array of Rx Eira Com signal masks.
<b>Type</b>	<code>const uint8 *const</code>
<b>Description</b>	A PNC may not have references to all RX EIRA Com signals. Therefore, the bits corresponding to PNCs which do not reference a Com signal have to be masked out from the newly received EIRA before being merged to the existing EIRA bits. This array contains references to bit arrays for each RX EIRA Com signal. If a Com Signal is associated to a PNC, then the bit corresponding to that PNC in the bit array will be set to 1. If a Com Signal is not associated to a PNC, then the corresponding bit in the bit array will be set to 0.

### 5.3.2.3.22. ComM\_PncRxUnfilteredEraMask

<b>Purpose</b>	
<b>Type</b>	<code>const uint8 *const</code>

### 5.3.2.3.23. ComM\_PncStateGWE

<b>Purpose</b>	Current state of PNC only available if PNC Gateway is enabled.
<b>Type</b>	<a href="#">ComM_PncModeType</a>

### 5.3.2.3.24. ComM\_PncZeroVectorAvoidanceRelease

<b>Purpose</b>	Request the network every time when entering Pnc_Requested state if ComMOPncVectorAvoidance configured as TRUE.
<b>Type</b>	<code>const uint8</code>

### 5.3.2.3.25. ComM\_ReadySleepNoNmTimeoutMs

<b>Purpose</b>	Ready Sleep Timeout.
----------------	----------------------



Type	const uint16
Description	These variables hold the timeout value for ready sleep state in ms. After this time the ready sleep state is left and the transition to the state "no communication" is performed. This timeout is only relevant and used for channels of Nm variant NONE and LIGHT. For channels of NmVariant LIGHT the timeout value is given by the configuration parameter ComMNmLightTimeout.

### 5.3.2.3.26. ComM\_RequestComMode

Purpose	ComM Requested Mode.
Type	<a href="#">ComM_ASR40_ModeType</a>
Description	This variable holds the currently requested mode.

### 5.3.2.3.27. ComM\_RequestComModeWithoutPnc

Purpose	ComM Requested Mode.
Type	<a href="#">ComM_ASR40_ModeType</a>
Description	This variable holds the currently requested mode without pnc request (user + managed, mangning).

### 5.3.2.3.28. ComM\_RxActiveEra

Purpose	This variable holds the received ERA signal.
Type	uint8

### 5.3.2.3.29. ComM\_RxActiveEraSignalStatus

Purpose	Holds the Status flag indicating changes in the ERA Active signals.
Type	boolean

### 5.3.2.3.30. ComM\_RxComSignalActiveEraCfg

Purpose	Array of Rx ERA ComSignal Handle Ids sorted in the ascending order.
---------	---



Type	const <a href="#">ComM_RxSignal_Struct_Type</a>
------	---

### 5.3.2.3.31. ComM\_RxComSignalEiraCfg

Purpose	Array of Rx EIRA ComSignal Handle Ids sorted in the ascending order.
Type	const <a href="#">ComM_RxSignal_Struct_Type</a>

### 5.3.2.3.32. ComM\_RxComSignalUnfilteredEraCfg

Purpose	Array of Rx ERA ComSignal Handle Ids sorted in the ascending order.
Type	const <a href="#">ComM_RxSignal_Struct_Type</a>

### 5.3.2.3.33. ComM\_RxEira

Purpose	This variable holds the received EIRA signal.
Type	uint8

### 5.3.2.3.34. ComM\_RxEiraSignalStatus

Purpose	Holds the Status flag indicating changes in the EIRA signals.
Type	boolean

### 5.3.2.3.35. ComM\_RxUnfilteredEra

Purpose	
Type	uint8

### 5.3.2.3.36. ComM\_RxUnfilteredEraSignalStatus

Purpose	Holds the Status flag indicating changes in the ERA Passive signals.
Type	boolean



### 5.3.2.3.37. ComM\_StoreNoWakeUpInhibition

<b>Purpose</b>	Storage for NoWakeUpInhibition in the NvM block pointed by ComMGlobalNvmBlock-Descriptor.
Type	const <a href="#">ComM_ASRS40_InhibitionStatusType</a>

### 5.3.2.3.38. ComM\_TxEiraEraActive

<b>Purpose</b>	This variable holds data for sending Eira signal.
Type	uint8

### 5.3.2.3.39. ComM\_TxEraPassive

<b>Purpose</b>	This variable holds data for sending Eira signal.
Type	uint8

### 5.3.2.3.40. ComM\_UserRequestedComMode

<b>Purpose</b>	ComM User Requested Mode.
Type	<a href="#">ComM_ASRS40_ModeType</a>
Description	This variable holds the current user requested mode.

## 5.3.2.4. Functions

### 5.3.2.4.1. ComM\_ASRS40\_GetCurrentComMode

<b>Purpose</b>	Retrieve the current mode of ComM via AUTOSAR 3.2 API.
<b>Synopsis</b>	Std_ReturnType <b>ComM_ASRS40_GetCurrentComMode</b> ( ComM_ASRS40_UserHandleType User , ComM_ASRS40_ModeType * ComMode );
<b>Service ID</b>	<a href="#">COMM_SID_GET_CURRENT_COM_MODE</a>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Reentrant



<b>Parameters (in)</b>	User	Handle of the user for which the query shall be performed.
<b>Parameters (out)</b>	ComMode	Current communication mode.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Getting of Current Communication Mode failed.
<b>Description</b>	This function retrieves the current communication mode of the ComM state machine.	

#### 5.3.2.4.2. ComM\_ASR32\_GetInhibitionStatus

<b>Purpose</b>	Retrieve inhibition status via AUTOSAR 3.2 API.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_ASR32_GetInhibitionStatus</b> ( NetworkHandleType Channel , ComM_ASR32_InhibitionStatusType * Status );	
<b>Service ID</b>	<a href="#"><u>COMM_SID_GET_INHIBITION_STATUS</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	Channel	Network channel on which the mode transition has occurred.
<b>Parameters (out)</b>	Status	Current inhibition status of ComM.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Getting of Inhibition Status failed.
<b>Description</b>	This function returns the inhibition status of a ComM channel.	

#### 5.3.2.4.3. ComM\_ASR32\_GetMaxComMode

<b>Purpose</b>	Retrieve the maximum allowed communication mode via AUTOSAR 3.2 API.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_ASR32_GetMaxComMode</b> ( ComM_ASR32_UserHandleType User , ComM_ASR32_ModeType * ComMode );	
<b>Service ID</b>	<a href="#"><u>COMM_SID_GET_MAX_COM_MODE</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	



<b>Parameters (in)</b>	User	Handle of the user for which the query shall be performed.
<b>Parameters (out)</b>	ComMode	Maximum allowed communication mode.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Getting of Maximum Communication Mode failed.
<b>Description</b>	This function retrieves the maximum allowed communication mode of the corresponding user.	

#### 5.3.2.4.4. ComM\_AS32\_GetRequestedMode

<b>Purpose</b>	Retrieve the communication mode requested by a user via AUTOSAR 3.2 API.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_AS32_GetRequestedMode</b> ( ComM_AS32_UserHandleType User , ComM_AS32_ModeType * ComMode );	
<b>Service ID</b>	<a href="#">COMM_SID_GET_REQUESTED_COM_MODE</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	User	Handle of the user for which the query shall be performed.
<b>Parameters (out)</b>	ComMode	Communication mode that has been requested most recently by that user.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Getting of the Requested Communication Mode failed.
<b>Description</b>	This function retrieves the communication mode that has been requested most recently by the given user.	

#### 5.3.2.4.5. ComM\_AS32\_LimitChannelToNoComMode

<b>Purpose</b>	Limit channel to "No Communication" via AUTOSAR 3.2 API.
<b>Synopsis</b>	Std_ReturnType <b>ComM_AS32_LimitChannelToNoComMode</b> ( NetworkHandleType Channel , boolean Status );



<b>Service ID</b>	<a href="#"><u>COMM_SID_LIMIT_CHANNEL_TO_NO_COM</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	Channel	Channel on which a limitations will be en(-dis)abled.
	Status	En(-dis)ables channel limitation.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Limiting of channel failed.
<b>Description</b>	This function en(-dis)ables a limitation for a channel.	

#### 5.3.2.4.6. ComM\_ASR32\_LimitECUToNoComMode

<b>Purpose</b>	Limit ECU to "No Communication" via AUTOSAR 3.2 API.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_ASR32_LimitECUToNoComMode</b> ( boolean Status ) ;	
<b>Service ID</b>	<a href="#"><u>COMM_SID_LIMIT_ECU_TO_NO_COM</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	Status	En(-dis)ables limitation.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Limiting of ECU failed.
<b>Description</b>	This function en(-dis)ables a limitation for a ECU.	

#### 5.3.2.4.7. ComM\_ASR32\_PreventWakeUp

<b>Purpose</b>	Changes Wakeup inhibition status for requested channel via AUTOSAR 3.2 API.
<b>Synopsis</b>	Std_ReturnType <b>ComM_ASR32_PreventWakeUp</b> ( NetworkHandleType Channel , boolean Status ) ;
<b>Service ID</b>	<a href="#"><u>COMM_SID_PREVENT_WAKE_UP</u></a>
<b>Sync/Async</b>	Synchronous



<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	Channel	on which a Wake Up inhibition will be en(-dis)abled
	Status	En(-dis)ables Wake up Inhibition.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Change of Wakeup Inhibition Status failed
<b>Description</b>	This function allows user to enable or disable the Wake Up inhibition for a requested channel.	

#### 5.3.2.4.8. ComM\_AS32\_ReadInhibitCounter

<b>Purpose</b>	Return the amount of rejected user requests via AUTOSAR 3.2 API.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_AS32_ReadInhibitCounter</b> ( uint16 * CounterValue );	
<b>Service ID</b>	<a href="#">COMM_SID_READ_INHIBIT_COUNTER</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (out)</b>	CounterValue	Amount of rejected user requests.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Reading of Inhibit Counter failed.
<b>Description</b>	This function returns the amount of rejected "Full Communication" user requests.	

#### 5.3.2.4.9. ComM\_AS32\_RequestComMode

<b>Purpose</b>	Request of a communication mode by a user via AUTOSAR 3.2 API.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_AS32_RequestComMode</b> ( ComM_AS32_UserHandleType User , ComM_AS32_ModeType ComMode );	
<b>Service ID</b>	<a href="#">COMM_SID_REQUEST_COM_MODE</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	



<b>Parameters (in)</b>	User	Handle of the user requesting the communication mode.
	ComMode	Desired communication mode.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Requesting of Communication Mode failed.
<b>Description</b>	<p>This function requests the given communication mode for the given user.</p> <p>If Rte usage is enables in the ComM configuration the declaration of this function is provided by Rte_ComM.h.</p>	

#### 5.3.2.4.10. ComM\_ASR32\_ResetInhibitCounter

<b>Purpose</b>	Reset inhibition counter via AUTOSAR 3.2 API.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_ASR32_ResetInhibitCounter</b> ( void );	
<b>Service ID</b>	<a href="#"><u>COMM_SID_RESET_INHIBIT_COUNTER</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Resetting of Inhibit Counter failed.
<b>Description</b>	This function resets the "Inhibited Full Communication Request Counter".	

#### 5.3.2.4.11. ComM\_ASR32\_SetECUGroupClassification

<b>Purpose</b>	Change the ECU Group Classification status via AUTOSAR 3.2 API.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_ASR32_SetECUGroupClassification</b> ( ComM_ASR32_InhibitionStatusType Status );	
<b>Service ID</b>	<a href="#"><u>COMM_SID_SET_ECU_GROUP_CLASS</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	Status	Inhibition status.



<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Setting of the ECU Group Classification failed.
<b>Description</b>	This function changes the ECU Group Classification status.	

#### 5.3.2.4.12. ComM\_AS40\_GetCurrentComMode

<b>Purpose</b>	Retrieve the current mode of ComM.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_AS40_GetCurrentComMode</b> ( ComM_AS40_UserHandleType User , ComM_AS40_ModeType * ComMode );	
<b>Service ID</b>	<a href="#"><u>COMM_SID_GET_CURRENT_COM_MODE</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	User	Handle of the user for which the query shall be performed.
<b>Parameters (out)</b>	ComMode	Current communication mode.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Getting of Current Communication Mode failed.
<b>Description</b>	This function retrieves the current communication mode of the ComM state machine.	

#### 5.3.2.4.13. ComM\_AS40\_GetCurrentPNCComMode

<b>Purpose</b>	Retrieve the current mode of a PNC.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_AS40_GetCurrentPNCComMode</b> ( ComM_AS40_UserHandleType User , ComM_AS40_ModeType * ComMode );	
<b>Service ID</b>	<a href="#"><u>COMM_SID_GET_CURRENT_PNC_COM_MODE</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	User	Handle of the user for which the query shall be performed.



<b>Parameters (out)</b>	ComMode	Current communication mode.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Getting of Current Communication Mode failed.
	E_MULTIPLE_PNC_ASSIGNED	multiple PNCs attached to the user
	E_NO_PNC_ASSIGNED	no PNC attached to the user
<b>Description</b>	This function retrieves the current communication mode of the ComM PNC state machine.	

#### 5.3.2.4.14. ComM\_AS40\_GetInhibitionStatus

<b>Purpose</b>	Retrieve inhibition status.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_AS40_GetInhibitionStatus</b> ( NetworkHandleType Channel , ComM_AS40_InhibitionStatusType * Status );	
<b>Service ID</b>	<a href="#">COMM_SID_GET_INHIBITION_STATUS</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	Channel	Network channel on which the mode transition has occurred.
<b>Parameters (out)</b>	Status	Current inhibition status of ComM.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Getting of Inhibition Status failed.
<b>Description</b>	This function returns the inhibition status of a ComM channel.	

#### 5.3.2.4.15. ComM\_AS40\_GetMaxComMode

<b>Purpose</b>	Retrieve the maximum allowed communication mode.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_AS40_GetMaxComMode</b> ( ComM_AS40_UserHandleType User , ComM_AS40_ModeType * ComMode );	
<b>Service ID</b>	<a href="#">COMM_SID_GET_MAX_COM_MODE</a>	
<b>Sync/Async</b>	Synchronous	



<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	User	Handle of the user for which the query shall be performed.
<b>Parameters (out)</b>	ComMode	Maximum allowed communication mode.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Getting of Maximum Communication Mode failed.
<b>Description</b>	This function retrieves the maximum allowed communication mode of the corresponding user.	

#### 5.3.2.4.16. ComM\_AS40\_GetRequestedComMode

<b>Purpose</b>	Retrieve the communication mode requested by a user.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_AS40_GetRequestedComMode</b> ( ComM_AS40_-UserHandleType User , ComM_AS40_ModeType * ComMode );	
<b>Service ID</b>	<a href="#">COMM_SID_GET_REQUESTED_COM_MODE</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	User	Handle of the user for which the query shall be performed.
<b>Parameters (out)</b>	ComMode	Communication mode that has been requested most recently by that user.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Getting of the Requested Communication Mode failed.
<b>Description</b>	This function retrieves the communication mode that has been requested most recently by the given user.	

#### 5.3.2.4.17. ComM\_AS40\_LimitChannelToNoComMode

<b>Purpose</b>	Limit channel to "No Communication".
----------------	--------------------------------------



<b>Synopsis</b>	Std_ReturnType <b>ComM_ASR40_LimitChannelToNoComMode</b> ( NetworkHandleType Channel , boolean Status );	
<b>Service ID</b>	<a href="#"><u>COMM_SID_LIMIT_CHANNEL_TO_NO_COM</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	Channel	Channel on which a limitations will be en(-dis)abled.
	Status	En(-dis)ables channel limitation.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Limiting of channel failed.
<b>Description</b>	This function en(-dis)ables a limitation for a channel.	

#### 5.3.2.4.18. ComM\_ASR40\_LimitECUToNoComMode

<b>Purpose</b>	Limit ECU to "No Communication".	
<b>Synopsis</b>	Std_ReturnType <b>ComM_ASR40_LimitECUToNoComMode</b> ( boolean Status );	
<b>Service ID</b>	<a href="#"><u>COMM_SID_LIMIT_ECU_TO_NO_COM</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	Status	En(-dis)ables limitation.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Limiting of ECU failed.
<b>Description</b>	This function en(-dis)ables a limitation for a ECU.	

#### 5.3.2.4.19. ComM\_ASR40\_PreventWakeUp

<b>Purpose</b>	Changes Wakeup inhibition status for requested channel.
<b>Synopsis</b>	Std_ReturnType <b>ComM_ASR40_PreventWakeUp</b> ( NetworkHandleType Channel , boolean Status );
<b>Service ID</b>	<a href="#"><u>COMM_SID_PREVENT_WAKE_UP</u></a>



<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	Channel	on which a Wake Up inhibition will be en(-dis)abled
	Status	En(-dis)ables Wake up Inhibition.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Change of Wakeup Inhibition Status failed
<b>Description</b>	This function allows user to enable or disable the Wake Up inhibition for a requested channel.	

#### 5.3.2.4.20. ComM\_AS40\_ReadInhibitCounter

<b>Purpose</b>	Return the amount of rejected user requests.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_AS40_ReadInhibitCounter</b> ( uint16 * CounterValue );	
<b>Service ID</b>	<a href="#"><u>COMM_SID_READ_INHIBIT_COUNTER</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (out)</b>	CounterValue	Amount of rejected user requests.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Reading of Inhibit Counter failed.
<b>Description</b>	This function returns the amount of rejected "Full Communication" user requests.	

#### 5.3.2.4.21. ComM\_AS40\_RequestComMode

<b>Purpose</b>	Request of a communication mode by a user.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_AS40_RequestComMode</b> ( ComM_AS40_UserHandleType User , ComM_AS40_ModeType ComMode );	
<b>Service ID</b>	<a href="#"><u>COMM_SID_REQUEST_COM_MODE</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	



<b>Parameters (in)</b>	User	Handle of the user requesting the communication mode.
	ComMode	Desired communication mode.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Requesting of Communication Mode failed.
<b>Description</b>	<p>This function requests the given communication mode for the given user.</p> <p>If Rte usage is enables in the ComM configuration the declaration of this function is provided by Rte_ComM.h.</p>	

#### 5.3.2.4.22. ComM\_AS40\_ResetInhibitCounter

<b>Purpose</b>	Reset inhibition counter.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_AS40_ResetInhibitCounter</b> ( void );	
<b>Service ID</b>	<a href="#"><u>COMM_SID_RESET_INHIBIT_COUNTER</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Resetting of Inhibit Counter failed.
<b>Description</b>	This function resets the "Inhibited Full Communication Request Counter".	

#### 5.3.2.4.23. ComM\_AS40\_SetECUGroupClassification

<b>Purpose</b>	Change the ECU Group Classification status.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_AS40_SetECUGroupClassification</b> ( ComM_AS40_InhibitionStatusType Status );	
<b>Service ID</b>	<a href="#"><u>COMM_SID_SET_ECU_GROUP_CLASS</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	Status	Inhibition status.



<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	Setting of the ECU Group Classification failed.
<b>Description</b>	This function changes the ECU Group Classification status.	

#### 5.3.2.4.24. ComM\_BitArrayClear

<b>Purpose</b>	Clear the x-th bit in a bit array.	
<b>Synopsis</b>	<code>void ComM_BitArrayClear ( uint8 * arr , uint16 idx );</code>	
<b>Parameters (in)</b>	arr	array of uint8
	idx	number of bit to be set to 0

#### 5.3.2.4.25. ComM\_BitArrayIsBitSet

<b>Purpose</b>	Test the x-th bit in a bit array.	
<b>Synopsis</b>	<code>boolean ComM_BitArrayIsBitSet ( const uint8 * arr , uint16 idx );</code>	
<b>Parameters (in)</b>	arr	array of uint8
	idx	number of bit to be tested
<b>Return Value</b>	TRUE or FALSE whether x-th bit is set or not.	

#### 5.3.2.4.26. ComM\_BitArraySet

<b>Purpose</b>	Set the x-th bit in a bit array.	
<b>Synopsis</b>	<code>void ComM_BitArraySet ( uint8 * arr , uint16 idx );</code>	
<b>Parameters (in)</b>	arr	array of uint8
	idx	number of bit to be set to 1

#### 5.3.2.4.27. ComM\_BusSM\_BusSleepMode

<b>Purpose</b>	Notify that the bus state management has entered Bus Sleep Mode.
----------------	--



<b>Synopsis</b>	<code>void ComM_BusSM_BusSleepMode ( NetworkHandleType Channel );</code>	
<b>Service ID</b>	<a href="#"><u>COMM_SID_BUSSM_BUS_SLEEP_MODE</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	Channel	Network channel on which the mode transition has occurred.
<b>Description</b>	This function provides a notification that the bus state management has entered Bus Sleep Mode.	

#### 5.3.2.4.28. ComM\_BusSM\_ModeIndication

<b>Purpose</b>	Indicate that BusSM has changed its communication mode.	
<b>Synopsis</b>	<code>void ComM_BusSM_ModeIndication ( NetworkHandleType Channel , uint8 * ComMode );</code>	
<b>Service ID</b>	<a href="#"><u>COMM_SID_BUS_SM_INDICATION</u></a>	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	Channel	Network channel on which the mode transition has occurred.
	ComMode	New communication mode of the BusSM.
<b>Description</b>	This function provides an indication that the BusSM has changed its communication mode. The new mode is supplied as a parameter.	

#### 5.3.2.4.29. ComM\_ChannelNoComInhibitionStatus

<b>Purpose</b>	Evaluate NoCom Inhibition for ComM Channels This function checks the ComM channel NoCom Inhibition is set.	
<b>Synopsis</b>	<code>boolean ComM_ChannelNoComInhibitionStatus ( uint8 instIdx );</code>	
<b>Parameters (in)</b>	instIdx	index of state machine instance
<b>Return Value</b>	Status of NoCom inhibition setting.	
	TRUE	if inhibition is set.
	FALSE	if inhibition is not set.



#### 5.3.2.4.30. ComM\_ChannelWakeUpInhibitionStatus

<b>Purpose</b>	Evaluate WakeUp Inhibition for ComM Channels This function checks the ComM channel WakeUp Inhibition is set.	
<b>Synopsis</b>	<code>boolean ComM_ChannelWakeUpInhibitionStatus ( uint8 instIdx );</code>	
<b>Parameters (in)</b>	instIdx	index of state machine instance
<b>Return Value</b>	Status of WakeUp inhibition setting.	
	TRUE	if inhibition is set.
	FALSE	if inhibition is not set.

#### 5.3.2.4.31. ComM\_CommunicationAllowed

<b>Purpose</b>	Indication to ComM when communication is allowed .	
<b>Synopsis</b>	<code>void ComM_CommunicationAllowed ( NetworkHandleType Channel , boolean Allowed );</code>	
<b>Parameters (in)</b>	Channel	Network channel on which communication is allowed.
	Allowed	communication is allowed (true) or not (false)
<b>Description</b>	Ecum or BswM shall indicate to ComM when communication is allowed.	

#### 5.3.2.4.32. ComM\_DCM\_ActiveDiagnostic

<b>Purpose</b>	Indicate an active diagnostic session by DCM.	
<b>Synopsis</b>	<code>void ComM_DCM_ActiveDiagnostic ( NetworkHandleType Channel );</code>	
<b>Service ID</b>	<a href="#"><u>COMM_SID_DCM_ACTIVE_DIAGNOSTIC</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	Channel	Network channel needed for Diagnostic communication.
<b>Description</b>	<p>This function provides an indication that DCM is currently handling an active diagnostic session on the channel given by ComM's configuration.</p> <p>This implies that for the respective channels, network mode must be requested.</p>	



#### 5.3.2.4.33. ComM\_DCM\_InactiveDiagnostic

<b>Purpose</b>	Indicate an inactive diagnostic session by DCM.	
<b>Synopsis</b>	<code>void ComM_DCM_InactiveDiagnostic ( NetworkHandleType Channel );</code>	
<b>Service ID</b>	<a href="#"><u>COMM_SID_DCM_INACTIVE_DIAGNOSTIC</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	Channel	Network channel no longer needed for Diagnostic communication.
<b>Description</b>	<p>This function provides an indication that DCM is currently not handling an active diagnostic session on the channel given by ComM's configuration.</p> <p>This implies that for the respective channels, network mode can be released when no other user requires the respective channel.</p>	

#### 5.3.2.4.34. ComM\_DeInit

<b>Purpose</b>	Deinitialize ComM module.
<b>Synopsis</b>	<code>void ComM_DeInit ( void );</code>
<b>Service ID</b>	<a href="#"><u>COMM_SID_DEINIT</u></a>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	This function deinitializes (terminates) the ComM module. De-init shall only be performed if all channels controlled by ComM are in "No Communication" state. The function call will be ignored if the state is not "No Communication".

#### 5.3.2.4.35. ComM\_EcuM\_PNCWakeUpIndication

<b>Purpose</b>	Notification of a wake up on the corresponding partial network cluster.
<b>Synopsis</b>	<code>void ComM_EcuM_PNCWakeUpIndication ( PNCHandleType PNCid );</code>
<b>Service ID</b>	<a href="#"><u>COMM_SID_ECU_M_PNC_WAKE_UP_INDICATION</u></a>
<b>Sync/Async</b>	Synchronous



<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	PNCid	Partial network index
<b>Description</b>	This function provides an indication that the ECU State Manager has received a wakeup event for the given PNC.	

#### 5.3.2.4.36. ComM\_EcuM\_WakeUpIndication

<b>Purpose</b>	Indicate that the ECU State Manager has received a wakeup.	
<b>Synopsis</b>	<code>void ComM_EcuM_WakeUpIndication ( NetworkHandleType Channel );</code>	
<b>Service ID</b>	<a href="#"><u>COMM_SID_ECU_M_WAKE_UP_INDICATION</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	Channel	Network channel on which the mode transition has occurred.
<b>Description</b>	This function provides an indication that the ECU State Manager has received a wakeup event for the given channel.	

#### 5.3.2.4.37. ComM\_EvaluatePnc

<b>Purpose</b>	Evaluate ComM Partial Network Cluster This function checks the ComM PNC mode and emits specific event related to the mode of the PNC.	
<b>Synopsis</b>	<code>void ComM_EvaluatePnc ( uint8 Pnc , ComM_AS40_ModeType ComMode ) ;</code>	
<b>Parameters (in)</b>	Pnc	Partial Network Cluster on which the mode transition has occurred.
	ComMode	Maximum allowed communication mode.

#### 5.3.2.4.38. ComM\_GetState

<b>Purpose</b>	Return current state.
<b>Synopsis</b>	<code>Std_ReturnType ComM_GetState ( NetworkHandleType Channel , ComM_StateType * State ) ;</code>
<b>Service ID</b>	<a href="#"><u>COMM_SID_GET_STATE</u></a>



<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	Channel	
<b>Parameters (out)</b>	State	(state of ComM.)
<b>Return Value</b>	Standard Return Code	
	E_OK	Function Successfully return current state of ComM state machine.
	E_NOT_OK	Return of current state of ComM state machine failed .
	COMM_E_UNINIT	ComM not initialized
<b>Description</b>	This function returns current state, including sub-state, of the ComM state machine.	

#### 5.3.2.4.39. ComM\_GetStatus

<b>Purpose</b>	Return the initialization status of the ComM module.	
<b>Synopsis</b>	Std_ReturnType <b>ComM_GetStatus</b> ( ComM_InitStatusType * Status );	
<b>Service ID</b>	<a href="#">COMM_SID_GET_STATUS</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (out)</b>	Status	Current initialization status of ComM.
<b>Return Value</b>	Standard Return Code	
	E_OK	Function serviced successfully.
	E_NOT_OK	Function execution failed.
<b>Description</b>	This function returns the initialization status of the ComM module.	

#### 5.3.2.4.40. ComM\_GetVersionInfo

<b>Purpose</b>	Get version information of the ComM module.	
<b>Synopsis</b>	void <b>ComM_GetVersionInfo</b> ( Std_VersionInfoType * VersionInfoPtr );	
<b>Service ID</b>	<a href="#">COMM_SID_GET_VERSION_INFO</a>	
<b>Sync/Async</b>	Synchronous	



<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (out)</b>	VersionInfoPtr	Pointer to where to store the version information of this module.
<b>Description</b>	<p>This service returns the version information of this module. The version information includes:</p> <ul style="list-style-type: none"> <li>➤ Module Id</li> <li>➤ Vendor Id</li> <li>➤ Vendor specific version numbers</li> </ul>	

#### 5.3.2.4.41. ComM\_Init

<b>Purpose</b>	Initialize the ComM module.	
<b>Synopsis</b>	<pre>void <b>ComM_Init</b> ( const ComM_ConfigType * ConfigPtr );</pre>	
<b>Service ID</b>	<a href="#">COMM_SID_INIT</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	ConfigPtr	- Pointer to a selected configuration structure.
<b>Description</b>	<p>This function initializes the ComM module by setting all internal state variables to defined values.</p> <p>Precondition: This API function has to be called before any other function (except <a href="#">ComM_GetStatus()</a>) can be called.</p>	

#### 5.3.2.4.42. ComM\_IsValidConfig

<b>Purpose</b>	Validates the post-build configuration data structure.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>ComM_IsValidConfig</b> ( const void * voidConfigPtr );</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	voidConfigPtr	- pointer to a ComM post-build data structure. If a NULL_PTR is passed, the ComM will attempt to retrieve the ComM post-



		build configuration from the PbcfgM module.
<b>Return Value</b>	Standard Return Code	
	E_OK	When the pre-compile, link-time and platform hash values stored within the post-build structure correspond to the hash values of the compiled source files.
	E_NOT_OK	Otherwise, E_NOT_OK will be returned.
<b>Description</b>	This function validates the post-build configuration data structure passed to the ComM_Init function.	

#### 5.3.2.4.43. ComM\_Nm\_BusSleepMode

<b>Purpose</b>	Notify that the network management has entered Bus Sleep Mode.	
<b>Synopsis</b>	<code>void ComM_Nm_BusSleepMode ( NetworkHandleType Channel );</code>	
<b>Service ID</b>	<a href="#"><u>COMM_SID_NM_BUS_SLEEP_MODE</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	Channel	Network channel on which the mode transition has occurred.
<b>Description</b>	This function provides a notification that the network management has entered Bus Sleep Mode.	

#### 5.3.2.4.44. ComM\_Nm\_NetworkMode

<b>Purpose</b>	Notify that the network management has entered Network Mode.	
<b>Synopsis</b>	<code>void ComM_Nm_NetworkMode ( NetworkHandleType Channel );</code>	
<b>Service ID</b>	<a href="#"><u>COMM_SID_NM_NETWORK_MODE</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	Channel	Network channel on which the mode transition has occurred.
<b>Description</b>	This function provides a notification that the network management has entered Network Mode.	



#### 5.3.2.4.45. ComM\_Nm\_NetworkStartIndication

<b>Purpose</b>	Indicate reception of NM PDU in Bus Sleep Mode.	
<b>Synopsis</b>	<pre>void ComM_Nm_NetworkStartIndication ( NetworkHandleType Channel );</pre>	
<b>Service ID</b>	<a href="#">COMM_SID_NM_NETWORK_START_IND</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	Channel	Network channel on which the NM PDU has been received.
<b>Description</b>	This function provides an indication that a NM PDU has been received in the Bus Sleep Mode, which indicates that some nodes in the network have already entered the Network Mode.	

#### 5.3.2.4.46. ComM\_Nm\_PreparesBusSleepMode

<b>Purpose</b>	Notify that the network management has entered Prepare Bus Sleep Mode.	
<b>Synopsis</b>	<pre>void ComM_Nm_PreparesBusSleepMode ( NetworkHandleType Channel );</pre>	
<b>Service ID</b>	<a href="#">COMM_SID_NM_PREPARE_BUS_SLEEP</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	Channel	Network channel on which the mode transition has occurred.
<b>Description</b>	This function provides a notification that the network management has entered Prepare Bus Sleep Mode.	

#### 5.3.2.4.47. ComM\_Nm\_RestartIndication

<b>Purpose</b>	Indicate a restart of NM.	
<b>Synopsis</b>	<pre>void ComM_Nm_RestartIndication ( NetworkHandleType Channel );</pre>	
<b>Service ID</b>	<a href="#">COMM_SID_NM_RESTART_IND</a>	
<b>Sync/Async</b>	Synchronous	



<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	Channel	Network channel on which the NM PDU has been received.
<b>Description</b>	This function provides an indication that Nmlf has started to shut down the coordinated buses, and not all coordinated buses have indicated bus sleep state, and on at least one of the coordinated buses NM is restarted.	

#### 5.3.2.4.48. ComM\_PncMainFunction

<b>Purpose</b>	Main processing function of PNC.
<b>Synopsis</b>	<code>void ComM_PncMainFunction ( void );</code>

#### 5.3.2.4.49. ComM\_PncProcessTimers

<b>Purpose</b>	Timer processing function of PNC.
<b>Synopsis</b>	<code>void ComM_PncProcessTimers ( void );</code>

#### 5.3.2.4.50. ComM\_ProcessTimers

<b>Purpose</b>	Processes the timer for a channel.	
<b>Synopsis</b>	<code>void ComM_ProcessTimers ( uint8 Channel );</code>	
<b>Parameters (in)</b>	Channel	Channel for which the timers are processed
<b>Description</b>	This function decreases the timer of the specified channel and emits an event, when the timer expires.	

#### 5.3.2.4.51. ComM\_SetChannelComMIInhibitionStatus

<b>Purpose</b>	Set Inhibition status for ComM Channels This function set and resets the ComM channel Inhibition.	
<b>Synopsis</b>	<code>void ComM_SetChannelComMIInhibitionStatus ( uint8 instIdx , ComM_ASR40_InhibitionStatusType Inhibitiontype , boolean Status ) ;</code>	
<b>Parameters (in)</b>	instIdx	index of state machine instance



	Inhibition type	Inhibition type of ComM channel.
	Status	to be set.

## 5.3.3. Integration notes

### 5.3.3.1. Exclusive areas

This section describes the exclusive areas used by the ComM module.

#### 5.3.3.1.1. SCHM\_COMM\_EXCLUSIVE\_AREA\_0

<b>Protected data structures</b>	All shared data that shall be protected from mutual access.
<b>Recommended locking mechanism</b>	This exclusive area must always be protected by a locking mechanism. The options for locking are described in the EB tresos AutoCore Generic documentation. Refer to the section <a href="#">Mapping exclusive areas</a> in the basic software modules in the <a href="#">Integration notes</a> section for details.

### 5.3.3.2. Production errors

Production errors are not reported by the ComM module.

### 5.3.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section [Memory mapping and compiler abstraction](#) in the [Integration notes](#) section for details.

The following table provides the list of sections that may be mapped for this module:

Memory section
VAR_CLEARED_16
VAR_INIT_16
VAR_INIT_8



CONST\_32

CONST\_16

CONST\_8

VAR\_CLEARED\_8

VAR\_CLEARED\_UNSPECIFIED

VAR\_INIT\_UNSPECIFIED

VAR\_POWER\_ON\_CLEARED\_UNSPECIFIED

CONFIG\_DATA\_UNSPECIFIED

CONST\_UNSPECIFIED

CODE

#### 5.3.3.4. Integration requirements

**WARNING**

**Integration requirements list is not exhaustive**

The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

Integration requirements are not listed for the ComM module.

## 5.4. EcuM

### 5.4.1. Configuration parameters

Containers included		
Container name	Multiplicity	Description
<a href="#">CommonPublishedInformation</a>	1..1	<b>Label:</b> Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
<a href="#">EcuMConfiguration</a>	1..1	<b>Label:</b> Configuration This container contains the configuration (parameters) of the ECU State Manager

**Containers included**

<a href="#">EcuMDefensiveProgramming</a>	1..1	<b>Label:</b> Defensive Programming Options  Parameters for defensive programming
<a href="#">EcuMFixedGeneral</a>	0..1	<i>The functionality related to this parameter is not supported by the current implementation.</i>  This container holds the general, pre-compile configuration parameters for the EcuMFixed.
<a href="#">EcuMFlexGeneral</a>	1..1	This container holds the general, pre-compile configuration parameters for the flexible ECU management.
<a href="#">EcuMGeneral</a>	1..1	<b>Label:</b> Pre-Compile Configuration Parameter  This container holds the general, pre-compile configuration parameters.
<a href="#">ReportToDem</a>	1..1	<b>Label:</b> Production error handling Production error handling
<a href="#">PublishedInformation</a>	1..1	<b>Label:</b> EB Published Information Additional published parameters not covered by Common-PublishedInformation container.

**Parameters included**

Parameter name	Multiplicity
<a href="#">IMPLEMENTATION_CONFIG_VARIANT</a>	1..1

Parameter Name	<b>IMPLEMENTATION_CONFIG_VARIANT</b>	
Label	Config Variant	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	VariantPostBuild	
Range	VariantPostBuild	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild

**5.4.1.1. CommonPublishedInformation**

Parameters included	Multiplicity
Parameter name	Multiplicity

**Parameters included**

<a href="#">ArMajorVersion</a>	1..1
<a href="#">ArMinorVersion</a>	1..1
<a href="#">ArPatchVersion</a>	1..1
<a href="#">SwMajorVersion</a>	1..1
<a href="#">SwMinorVersion</a>	1..1
<a href="#">SwPatchVersion</a>	1..1
<a href="#">ModuleId</a>	1..1
<a href="#">VendorId</a>	1..1
<a href="#">Release</a>	1..1

**Parameter Name** [ArMajorVersion](#)

<b>Label</b>	AUTOSAR Major Version
<b>Description</b>	Major version number of AUTOSAR specification on which the appropriate implementation is based on.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	3
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

**Parameter Name** [ArMinorVersion](#)

<b>Label</b>	AUTOSAR Minor Version
<b>Description</b>	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	0
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

**Parameter Name** [ArPatchVersion](#)

<b>Label</b>	AUTOSAR Patch Version
<b>Description</b>	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.



<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	0
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SwMajorVersion</b>
<b>Label</b>	Software Major Version
<b>Description</b>	Major version number of the vendor specific implementation of the module.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	5
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SwMinorVersion</b>
<b>Label</b>	Software Minor Version
<b>Description</b>	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	15
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SwPatchVersion</b>
<b>Label</b>	Software Patch Version
<b>Description</b>	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	8
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH



<b>Parameter Name</b>	<b>ModuleId</b>
<b>Label</b>	Numeric Module ID
<b>Description</b>	Module ID of this module from Module List
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	10
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>VendorId</b>
<b>Label</b>	Vendor ID
<b>Description</b>	Vendor ID of the dedicated implementation of this module according to the AU-TOSAR vendor list
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	1
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>Release</b>
<b>Label</b>	Release Information
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING_LABEL
<b>Default value</b>	
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

#### 5.4.1.2. EcuMConfiguration

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">EcuMCommonConfiguration</a>	1..1	This container contains the common configuration (parameters) of the ECU State Manager



Containers included		
<a href="#">EcuMFixedConfiguration</a>	0..1	<p>This container contains the configuration (parameters) of the EcuMFixed. Only applicable if EcuMFixed is implemented.</p> <p>Dependency:</p> <ul style="list-style-type: none"> <li>▶ The parameter <code>EcuMRteUsage</code> shall be enabled.</li> </ul>
<a href="#">EcuMFlexConfiguration</a>	1..1	<p><b>Label:</b> EcuM Flex Configuration</p> <p>This container contains the configuration (parameters) of the EcuMFlex.</p>

#### 5.4.1.3. EcuMCommonConfiguration

Containers included		
Container name	Multiplicity	Description
<a href="#">EcuMDefaultShutdownTarget</a>	1..1	<p><b>Label:</b> Default Shutdown Target</p> <p>This container describes the default shutdown target to be selected by EcuM. The actual shutdown target may be overridden by the <code>EcuM_SelectShutdownTarget()</code> service.</p>
<a href="#">EcuMDriverInitListOne</a>	0..1	<p><b>Label:</b> Module Initialization List One</p> <p>Container for Init Block I. This container holds a list of module IDs that will be initialized. Each module in the list will be called for initialization in the list order. All modules in this list are initialized before the OS is started and so these modules require no OS support.</p>
<a href="#">EcuMDriverInitListZero</a>	0..1	<p><b>Label:</b> Module Initialization List Zero</p> <p>Container for Init Block 0. This container holds a list of module IDs that will be initialized. Each module in the list will be called for initialization in the list order. All modules in this list are initialized before the post-build configuration has been loaded and the OS is initialized. Therefore, these modules may not use post-build configuration.</p>
<a href="#">EcuMDriverRestartList</a>	0..1	<p>List of module IDs that shall be re-initialized during Wake-upRestart. Each module in the list will be called for initialization in the list order by the callout <code>EcuM_AL_DriverRestart()</code>.</p> <p>Restriction:</p>



### Containers included

		<ul style="list-style-type: none"> <li>▶ Entries in this list must appear in the same order as in the list of <code>EcuM_DriverInitListOne</code> and <code>EcuM_DriverInitListZero</code>. <code>EcuMDriverRestartList</code> shall only contain a subset of the combined list of init block 0 and init block 1 drivers.</li> </ul>
<a href="#">EcuMSleepMode</a>	1..n	<p>This container describes the configured sleep mode. The name of this container specifies the symbolic name of the sleep mode.</p>
<a href="#">EcuMWakeupSource</a>	5..32	<p>Configuration of Wakeup Sources.</p> <p>The short name of this container will be available as pre-processor define which expands to values corresponding to the bit positions(wakeup Source ID) of this wakeup Source.</p> <p><i>Attention:</i> For the first five wakeup sources standardized names will be used:</p> <ul style="list-style-type: none"> <li>▶ Wakeup Source ID: 0: ECUM_WKSOURCE_POWER</li> <li>▶ Wakeup Source ID: 1: ECUM_WKSOURCE_RESET</li> <li>▶ Wakeup Source ID: 2: ECUM_WKSOURCE_INTERNAL_RESET</li> <li>▶ Wakeup Source ID: 3: ECUM_WKSOURCE_INTERNAL_WDG</li> <li>▶ Wakeup Source ID: 4: ECUM_WKSOURCE_EXTERNAL_WDG</li> </ul>
<a href="#">EcuMDemEventParameter-Refs</a>	0..1	<p>References to DemEventParameter elements which shall be invoked using the API <code>Dem_ReportErrorStatus()</code> in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's <code>DemEventId</code> value. The standardized errors are provided in this container and can be extended by vendor specific error references.</p>

### Parameters included

Parameter name	Multiplicity
<a href="#">EcuMConfigConsistencyHash</a>	1..1
<a href="#">EcuMDefaultAppMode</a>	1..1
<a href="#">EcuMOSResource</a>	0..n

**Parameters included**

<a href="#">EcuMDefaultAppModeBaseId</a>	0..1
--	------

<b>Parameter Name</b>	<b>EcuMConfigConsistencyHash</b>
<b>Label</b>	Configuration Consistency Hash
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>A hash value is generated across all pre-compile and link-time parameters of all BSW modules. This hash value is compared against a field in the <code>EcuM_ConfigType</code> and hence allows checking the consistency of the entire configuration.</p> <p>Dependency:</p> <ul style="list-style-type: none"> <li>▶ The maximal value for <code>EcuMConfigConsistencyHash</code> is 18446744073709551615 (= 0xFFFFFFFFFFFFFF = <math>(2^{64}-1)</math>) according to EcuM R4.0, but EB tresos Studio limits the maximum value to 9223372036854775807 (= 0x7FFFFFFFFFFFF = <math>(2^{63}-1)</math>).</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Range</b>	$\leq 9223372036854775807$ $\geq 0$
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMDefaultAppMode</b>
<b>Label</b>	OS Application Mode
<b>Description</b>	Reference to the default application mode of the OS, loaded when the ECU comes out of reset.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMOSResource</b>
-----------------------	-----------------------



<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>
	Reference to the OS resource which is used to bring the ECU into sleep mode. In case of multi core Ecu's each core shall have an own OsResource.
<b>Multiplicity</b>	0..n
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMDefaultAppModeBaseId</b>	
<b>Label</b>	Offset OS Index Value	
<b>Description</b>	Parameter used to set the offset value for the EcuMDefaultAppMode.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Range</b>	<=255 >=0	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

#### 5.4.1.4. EcuMDefaultShutdownTarget

Parameters included		
Parameter name		Multiplicity
<a href="#">EcuMDefaultState</a>		1..1
<a href="#">EcuMDefaultResetModeRef</a>		0..1
<a href="#">EcuMDefaultSleepModeRef</a>		0..1

<b>Parameter Name</b>	<b>EcuMDefaultState</b>
<b>Label</b>	Default Shutdown State
<b>Description</b>	This parameter describes the state of the default shutdown target selected when the ECU comes out of reset. If EcuMStateSleep is selected, the parameter EcuMDefaultSleepModeRef selects the specific sleep mode.



	<ul style="list-style-type: none"> <li>▶ If EcuMStateReset is selected then the default reset mode implicitly selected as ECUM_RESET MCU since the current SWS do not have the configuration parameter reference to EcuM Reset Modes.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	EcuMStateOff
<b>Range</b>	<ul style="list-style-type: none"> <li>EcuMStateOff</li> <li>EcuMStateReset</li> <li>EcuMStateSleep</li> </ul>
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMDefaultResetModeRef</b>
<b>Label</b>	Default Reset Mode
<b>Description</b>	<p>Reference to the default reset mode.</p> <p>Dependency:</p> <ul style="list-style-type: none"> <li>▶ Applicable only if EcuMDefaultShutdownTarget is selected as EcuMStateReset.</li> </ul>
<b>Multiplicity</b>	0..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMDefaultSleepModeRef</b>
<b>Label</b>	Default Sleep Mode
<b>Description</b>	<p>Reference to the default sleep mode.</p> <p>Dependency:</p> <ul style="list-style-type: none"> <li>▶ Applicable only if EcuMDefaultShutdownTarget is selected as EcuMStateSleep.</li> </ul>
<b>Multiplicity</b>	0..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC



### 5.4.1.5. EcuMDriverInitListOne

Containers included		
Container name	Multiplicity	Description
EcuMDriverInitItem	1..n	This container describes one entry in a driver init list.

### 5.4.1.6. EcuMDriverInitItem

Parameters included	
Parameter name	Multiplicity
<a href="#">EcuMEnableVendorApiInfix</a>	1..1
<a href="#">EcuMModuleHeaderFile</a>	1..1
<a href="#">EcuMModuleID</a>	1..1
<a href="#">EcuMModuleInitConfigStr</a>	0..1
<a href="#">EcuMModuleService</a>	1..1
<a href="#">EcuMEnableDriver</a>	1..1

Parameter Name	<b>EcuMEnableVendorApiInfix</b>	
Label	EcuMEnableVendorApiInfix	
Description	<p>Enable this parameter if <code>VendorApiInfix</code> is available for a module. This generate the module service in the following format: &lt;function&gt;EcuMModuleID_VendorId_VendorApiInfix_EcuMModuleService()&lt;/function&gt; Also this considers the existance of header file: <code>EcuMModuleID_VendorId_VendorApiInfix.h</code></p> <p>Dependency:</p> <ul style="list-style-type: none"> <li>▶ The module shall be configured in <code>EcuMFlexModuleConfigurationRef</code></li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>EcuMModuleHeaderFile</b>	
Label	Module Header File	
Description	Base name of the header file to be included for the usage of the module.	



	<p>If this parameter is left blank, the EcuM will include the header file with base name as entered in <code>EcuModuleID</code>.</p> <p>E.g if the <code>Can</code> is configured to be initialized and the <code>EcuModuleHeaderFile</code> is kept blank, the EcuM will include <code>Can.h</code>, else the EcuM will include the header with the file name assembled from the base name defined by <code>EcuModuleHeaderFile</code> and the postfix <code>.h</code>.</p>
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<code>VariantPostBuild:</code> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<code>EcuModuleID</code>
<b>Label</b>	Module ID
<b>Description</b>	<p>Specify the module short name (aka module ID) of the module to be initialized; e.g. <code>Adc</code></p> <p>The name must not be empty and follow the naming conventions for valid C identifiers.</p> <p>It is assumed that there exists:</p> <ul style="list-style-type: none"> <li>▶ A header file named after the value of this parameter; e.g. <code>Adc.h</code></li> <li>▶ A initialization function (declared in the mentioned header file) prefixed with the value of this parameter; e.g. <code>Adc_Init()</code></li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<code>VariantPostBuild:</code> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<code>EcuModuleInitConfigStr</code>
<b>Label</b>	Module Config String
<b>Description</b>	<p>To find the init-configuration of most Autosar modules, the EcuM evaluates the <code>EcuModuleConfigurationRefs</code> given in this configuration. The name of the multiple-configuration container these references refer to also is the linker-symbol of the configuration defined in that container. That symbol is then passed to the Init-function.</p> <p>While initializing a module which does not use the multiple configuration container name as symbol for the configuration object, you cannot use the <code>EcuMod-</code></p>



	<p>uleConfigurationRef to determine the pointer, which is to be passed to the Init function. In this case, you <i>must</i> enable and provide the parameter. The parameter is then used as parameter for the Init-method of the very module.</p> <p>It is also possible to <i>override</i> the standard configuration of any module using a parameter in it's Init function using this parameter. If this parameter is given, the EcuMModuleConfigurationRef is ignored.</p> <p>The string given here must evaluate to a C address, e.g. as plain number, or as address operator followed by a constant name.</p> <p>Some Autosar modules do not have a parameter in the signature of their MOD_Init() method. For those modules that do not require a parameter, do not set this parameter but disable it.</p>
<b>Multiplicity</b>	0..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EcuMModuleService</b>
<b>Label</b>	Module Service
<b>Description</b>	<p>The service to be called to initialize that module, e.g. Init, PreInit, Start etc.</p> <p>If the service is Init and the parameter EcuMModuleConfigurationRef has been set for that module, the corresponding pointer to the init structure (&lt; Module &gt;_ConfigType) shall be passed as an argument.</p>
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMEnableDriver</b>
<b>Label</b>	Enable Driver
<b>Description</b>	If enabled, this DriverInitItem will be used by the EcuM_AL_DriverInitOne() function.
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	true



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

#### 5.4.1.7. EcuMDriverInitListZero

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">EcuMDriverInitItem</a>	1..n	This container describes one entry in a driver init list.

#### 5.4.1.8. EcuMDriverInitItem

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMEnableVendorApiInfix</a>	1..1
<a href="#">EcuMModuleHeaderFile</a>	1..1
<a href="#">EcuMModuleID</a>	1..1
<a href="#">EcuMModuleService</a>	1..1

<b>Parameter Name</b>	<b>EcuMEnableVendorApiInfix</b>
<b>Label</b>	EcuMEnableVendorApiInfix
<b>Description</b>	Enable this parameter if <code>VendorApiInfix</code> is available for a module. This generate the module service in the following format: <code>&lt;function&gt;EcuMModuleID_VendorId_VendorApiInfix_EcuMModuleService()&lt;/function&gt;</code> Also this considers the existance of header file: <code>EcuMModuleID_VendorId_VendorApiInfix.h</code>  Dependency: <ul style="list-style-type: none"><li>▶ The module shall be configured in <code>EcuMFlexModuleConfigurationRef</code></li></ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	Elektrobit Automotive GmbH
<b>Parameter Name</b>	<b>EcuMModuleHeaderFile</b>
<b>Label</b>	Module Header File



<b>Description</b>	Base name of the header file to be included for the usage of the module.  If this parameter is left blank, the EcuM will include the header file with base name as entered in <code>EcuModuleID</code> .  E.g if the <code>Can</code> is configured to be initialized and the <code>EcuModuleHeaderFile</code> is kept blank, the EcuM will include <code>Can.h</code> , else the EcuM will include the header with the file name assembled from the base name defined by <code>EcuModuleHeaderFile</code> and the postfix <code>.h</code> .
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EcuModuleID</b>
<b>Label</b>	Module ID
<b>Description</b>	Specify the module short name (aka module ID) of the module to be initialized; e.g. <code>Adc</code>  The name must not be empty and follow the naming conventions for valid C identifiers.  It is assumed that there exists: <ul style="list-style-type: none"><li>▶ A header file named after the value of this parameter; e.g. <code>Adc.h</code></li><li>▶ A initialization function (declared in the mentioned header file) prefixed with the value of this parameter; e.g. <code>Adc_Init()</code></li></ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuModuleService</b>
<b>Label</b>	Module Service
<b>Description</b>	The service to be called to initialize that module, e.g. Init, PreInit, Start etc.  If the service is Init and the parameter <code>EcuModuleConfigurationRef</code> has been set for that module, the corresponding pointer to the init structure (<Module>_ConfigType) and in case of multiple instantiation an uint8 value to identify the instance of the module(<MSN>_CtrlIdx) shall be passed as an argument.



<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.4.1.9. EcuMDriverRestartList

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">EcuMDriverInitItem</a>	1..n	<p>This container describes one entry in a driver init list.</p> <p>Restriction:</p> <ul style="list-style-type: none"> <li>▶ The <code>EcuMDriverInitItem</code> name in <code>EcuMDriverRestartList</code> shall be the same for all EcuM configuration containers</li> </ul>

#### 5.4.1.10. EcuMDriverInitItem

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMEnableVendorApiInfix</a>	1..1
<a href="#">EcuMModuleHeaderFile</a>	1..1
<a href="#">EcuMModuleID</a>	1..1
<a href="#">EcuMModuleInitConfigStr</a>	0..1
<a href="#">EcuMModuleService</a>	1..1
<a href="#">EcuMEnableDriver</a>	1..1

<b>Parameter Name</b>	<a href="#">EcuMEnableVendorApiInfix</a>
<b>Label</b>	EcuMEnableVendorApiInfix
<b>Description</b>	<p>Enable this parameter if <code>VendorApiInfix</code> is available for a module. This generate the module service in the following format: <code>&lt;function&gt;EcuMModuleID_VendorId_VendorApiInfix_EcuMModuleService()&lt;/function&gt;</code> Also this considers the existance of header file: <code>EcuMModuleID_VendorId_VendorApiInfix.h</code></p> <p>Dependency:</p>



	► The module shall be configured in <code>EcuMFlexModuleConfigurationRef</code>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EcuMModuleHeaderFile</b>
<b>Label</b>	Module Header File
<b>Description</b>	<p>Base name of the header file to be included for the usage of the module.</p> <p>If this parameter is left blank, the EcuM will include the header file with base name as entered in <code>EcuMModuleID</code>.</p> <p>E.g if the <code>Can</code> is configured to be initialized and the <code>EcuMModuleHeaderFile</code> is kept blank, the EcuM will include <code>Can.h</code>, else the EcuM will include the header with the file name assembled from the base name defined by <code>EcuMModuleHeaderFile</code> and the postfix <code>.h</code>.</p>
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EcuMModuleID</b>
<b>Label</b>	Module ID
<b>Description</b>	<p>Specify the module short name (aka module ID) of the module to be initialized; e.g. <code>Adc</code></p> <p>The name must not be empty and follow the naming conventions for valid C identifiers.</p> <p>It is assumed that there exists:</p> <ul style="list-style-type: none"> <li>► A header file named after the value of this parameter; e.g. <code>Adc.h</code></li> <li>► A initialization function (declared in the mentioned header file) prefixed with the value of this parameter; e.g. <code>Adc_Init()</code></li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>EcuMModuleInitConfigStr</b>
<b>Label</b>	Module Config String
<b>Description</b>	<p>To find the init-configuration of most Autosar modules, the EcuM evaluates the <code>EcuMModuleConfigurationRefs</code> given in this configuration. The name of the multiple-configuration container these references refer to also is the linker-symbol of the configuration defined in that container. That symbol is then passed to the Init-function.</p> <p>While initializing a module which does not use the multiple configuration container name as symbol for the configuration object, you cannot use the <code>EcuMModuleConfigurationRef</code> to determine the pointer, which is to be passed to the Init function. In this case, you <i>must</i> enable and provide the parameter. The parameter is then used as parameter for the Init-method of the very module.</p> <p>It is also possible to <i>override</i> the standard configuration of any module using a parameter in its Init function using this parameter. If this parameter is given, the <code>EcuMModuleConfigurationRef</code> is ignored.</p> <p>The string given here must evaluate to a C address, e.g. as plain number, or as address operator followed by a constant name.</p> <p>Some Autosar modules do not have a parameter in the signature of their <code>MOD_Init()</code> method. For those modules that do not require a parameter, do not set this parameter but disable it.</p>
<b>Multiplicity</b>	0..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>PreCompile:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EcuMModuleService</b>
<b>Label</b>	Module Service
<b>Description</b>	<p>The service to be called to initialize that module, e.g. Init, PreInit, Start etc.</p> <p>If the service is Init and the parameter <code>EcuMModuleConfigurationRef</code> has been set for that module, the corresponding pointer to the init structure (&lt;Module&gt;_ConfigType) shall be passed as an argument.</p>
<b>Multiplicity</b>	1..1



Type	STRING
Configuration class	<b>VariantPostBuild:</b> VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	<b>EcuMEnableDriver</b>	
Label	Enable Driver	
Description	If enabled, this DriverInitItem will be used by the EcuM_AL_DriverRestart() function.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	<b>VariantPostBuild:</b> VariantPostBuild	
Origin	Elektrobit Automotive GmbH	

#### 5.4.1.11. EcuMSleepMode

Parameters included	
Parameter name	Multiplicity
<a href="#">EcuMSleepModelId</a>	1..1
<a href="#">EcuMSleepModeSuspend</a>	1..1
<a href="#">EcuMSleepModeMcuModeRef</a>	1..1
<a href="#">EcuMWakeupSourceMask</a>	1..n

Parameter Name	<b>EcuMSleepModelId</b>	
Label	Sleep Mode ID	
Description	Unique ID for sleep mode. The ID has to be zero-based and consecutive.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	<b>VariantPostBuild:</b> VariantPostBuild	
Origin	AUTOSAR_ECUC	

Parameter Name	<b>EcuMSleepModeSuspend</b>	
Description	Set this flag to true, if the CPU is suspended, halted, or powered off in the sleep mode. If the CPU keeps running in this sleep mode, then this flag must be set to false.	



<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMSleepModeMcuModeRef</b>
<b>Label</b>	Sleep Mcu Mode
<b>Description</b>	Reference to the corresponding MCU mode for this sleep mode.
<b>Multiplicity</b>	1..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMWakeupSourceMask</b>
<b>Label</b>	Wakeup Source Reference
<b>Description</b>	Reference to a wakeup source that shall be enabled for this sleep mode.
<b>Multiplicity</b>	1..n
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.4.1.12. EcuMWakeupSource

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuValidationTimeout</a>	0..1
<a href="#">EcuMWakeupSourceld</a>	1..1
<a href="#">EcuMWakeupSourcePolling</a>	1..1
<a href="#">EcuMComMChannelRef</a>	0..1
<a href="#">EcuMComMPNCRef</a>	0..1
<a href="#">EcuMResetReasonRef</a>	0..1



<b>Parameter Name</b>	<b>EcuMValidationTimeout</b>	
<b>Label</b>	Validation Timeout (s)	
<b>Description</b>	<p>The validation timeout is the period in seconds for which the ECU State Manager will wait for the validation of a wakeup event.</p> <p>If the timeout parameter is disabled the ECU Manager does not validate the wakeup source. Reported wakeup events of wakeup sources with disabled timeout parameter are considered to be always valid.</p> <p>The validation timeout must always be disabled for all default wakeup sources.</p>	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.0	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>EcuMWakeupSourceId</b>	
<b>Label</b>	Wakeup Source ID	
<b>Description</b>	<p>This parameter is the identifier of a wakeup source.</p> <p>The numerical value ranging from 0 to 31 defines the position corresponding to this wakeup source in all instances of <code>EcuM_WakeupSourceType</code> bitfield.</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>EcuMWakeupSourcePolling</b>	
<b>Label</b>	Wakeup Source Polling	
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	



<b>Parameter Name</b>	<b>EcuMComMChannelRef</b>	
<b>Label</b>	ComM Channel	
<b>Description</b>	Reference to a Network (channel) defined in the Communication Manager. No reference indicates that the wakeup source is not a communication channel.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>EcuMComMPNCRef</b>	
<b>Label</b>	ComM PNC	
<b>Description</b>	This is a reference to a one or more PNC's defined in the Communication Manager. No reference indicates that the wakeup source is not assigned to a partial network.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>EcuMResetReasonRef</b>	
<b>Description</b>	This parameter describes the mapping of reset reasons detected by the MCU driver into wakeup sources.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

#### 5.4.1.13. EcuMDemEventParameterRefs

Parameters included		
Parameter name	Multiplicity	
<a href="#">ECUM_E_ALL_RUN_REQUESTS_KILLED</a>	0..1	
<a href="#">ECUM_E_CONFIGURATION_DATA_INCONSISTENT</a>	0..1	

**Parameters included**[ECUM\\_E\\_RAM\\_CHECK\\_FAILED](#)

0..1

<b>Parameter Name</b>	<b>ECUM_E_ALL_RUN_REQUESTS_KILLED</b>	
<b>Description</b>	<p>Reference to the DemEventParameter which shall be issued when the error ECUM_E_ALL_RUN_REQUESTS_KILLED has occurred.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ EcuMFixedConfiguration is configured to enable EcuM Fixed support.</li> </ul> <p>Further notes:</p> <ul style="list-style-type: none"> <li>▶ Activation: Thrown, if the service EcuM_KillAllRunRequests is issued.</li> <li>▶ Healing: None. The error resides in memory until it is deleted.</li> <li>▶ Trigger debounce: None. The error is reported on first occurrence.</li> <li>▶ Rate of diagnostic checks: None. This error is thrown by every call of the service that reports this error. A list of API functions that report this error can be found in the table of production errors in the <i>Integration notes</i> section of the module references.</li> </ul>	
<b>Multiplicity</b>	0..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ECUM_E_CONFIGURATION_DATA_INCONSISTENT</b>	
<b>Description</b>	<p>Reference to the DemEventParameter which shall be issued when the error "ECUM_E_CONFIGURATION_DATA_INCONSISTENT" has occurred.</p> <p><i>The functionality related to this parameter is not supported by the current implementation.</i></p>	
<b>Multiplicity</b>	0..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>ECUM_E_RAM_CHECK_FAILED</b>	
<b>Description</b>	<p>Reference to the DemEventParameter which shall be issued when the error ECUM_E_RAM_CHECK_FAILED has occurred.</p>	



	<p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ <code>EcuMRamChkFailedReportToDem</code>: Select DEM to enable the reporting of <code>ECUM_E_RAM_CHECK_FAILED</code>.</li> </ul> <p>Further notes:</p> <ul style="list-style-type: none"> <li>▶ Activation: Thrown, if the RAM check during wakeup failed.</li> <li>▶ Healing: Dependent on the implementation of callout function <code>EcuM_CheckRamHash()</code> that must be implemented by user.</li> <li>▶ Trigger debounce: Dependent on the implementation of callout function <code>EcuM_CheckRamHash()</code> that must be implemented by user.</li> <li>▶ Rate of diagnostic checks: Checked on every call of the service that reports this error. A list of API functions that report this error can be found in the table of production errors in the <i>Integration notes</i> section of the module references.</li> </ul>
<b>Multiplicity</b>	0..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.4.1.14. EcuMFixedConfiguration

Containers included		
Container name	Multiplicity	Description
<a href="#">EcuMDriverInitListThree</a>	0..1	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Container for Init Block III. This container holds a list of module IDs that will be initialized. Each module in the list will be called for initialization in the list order. All modules in this list are initialized after the OS is started and so these modules may use OS support. These modules may also rely on the Nvram ReadAll job to have provided all data.</p>
<a href="#">EcuMDriverInitListTwo</a>	0..1	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Container for Init Block II. This container holds a list of module IDs that will be initialized. Each module in the list will be</p>



### Containers included

		called for initialisation in the list order. All modules in this list are initialized after the OS is started and so these modules may use OS support. These modules may not rely on the Nvram ReadAll job to have provided all data.
<a href="#">EcuMFixedUserConfig</a>	1..n	<p>These containers describe the identifiers that are needed to refer to a software component or another appropriate entity in the system which is designated to request the RUN state. Application requestors refer to entities above RTE, system requestors to entities below RTE (e.g. Communication Manager).</p>
<a href="#">EcuMTTII</a>	0..n	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This container contains parameters for the TTII protocol. The list must contain at least one element when <code>ECUM_TTII_ENABLED</code> is set to true.</p> <p>Restriction:</p> <ul style="list-style-type: none"> <li>▶ The <code>EcuMTTII</code> name shall be the same for all EcuM configuration containers</li> </ul>
<a href="#">EcuMWdgM</a>	0..1	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This container holds the configuration parameters for the interaction between the Watchdog Manager (WdgM) and EcuM. The WdgM mode to be selected in a specific Sleep Mode of EcuM is configured in the EcuMSleepMode container.</p>

### Parameters included

Parameter name	Multiplicity
<a href="#">EcuMNvramReadallTimeout</a>	1..1
<a href="#">EcuMNvramWriteallTimeout</a>	1..1
<a href="#">EcuMRunMinimumDuration</a>	1..1
<a href="#">EcuMFixedModuleConfigurationRef</a>	0..n
<a href="#">EcuMComMCommunicationAllowedList</a>	0..n
<a href="#">EcuMNormalMcuModeRef</a>	1..1

Parameter Name	EcuMNvramReadallTimeout



<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>
	Period given in seconds for which the ECU State Manager will wait until it considers a ReadAll job of the NVRAM Manager as failed.
	Dependency:
	► The value shall not be smaller than <code>EcuMMainFunctionPeriod</code> .
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMNvramWriteallTimeout</b>
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>
	Period given in seconds for which the ECU State Manager will wait until it considers a WriteAll job of the NVRAM Manager as failed.
	Dependency:
	► The value shall not be smaller than <code>EcuMMainFunctionPeriod</code> .
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMRunMinimumDuration</b>
<b>Description</b>	Duration given in seconds for which the ECU State Manager will stay in RUN state even when no one requests RUN. This duration should be long at least as long as a SW-Cs needs to request RUN.
	Dependency:
	► The value shall not be smaller than <code>EcuMMainFunctionPeriod</code> .
<b>Multiplicity</b>	1..1



Type	FLOAT
Default value	0.0
Configuration class	<b>VariantPostBuild:</b> VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	<b>EcuMFixedModuleConfigurationRef</b>
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> References to the init structure of the corresponding BSW module.
Multiplicity	0..n
Type	CHOICE-REFERENCE
Configuration class	<b>PostBuild:</b> VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	<b>EcuMComMCommunicationAllowedList</b>
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> References to the ComMChannels for which EcuM has to call ComM_CommunicationAllowed.
Multiplicity	0..n
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	<b>PreCompile:</b> VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	<b>EcuMNormalMcuModeRef</b>
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i>  This parameter is a reference to the normal MCU mode to be restored after a sleep.  Restriction: <ul style="list-style-type: none"><li>▶ This parameter shall have the same value in all configurations</li></ul>
Multiplicity	1..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	<b>VariantPostBuild:</b> VariantPostBuild



<b>Origin</b>	AUTOSAR_ECUC
---------------	--------------

#### 5.4.1.15. EcuMDriverInitListThree

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">EcuMDriverInitItem</a>	1..n	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This container describes one entry in a driver init list.</p> <p>Restriction:</p> <ul style="list-style-type: none"> <li>▶ The <code>EcuMDriverInitItem</code> name in <code>EcuMDriverInitListThree</code> shall be the same for all EcuM configuration containers</li> </ul>

#### 5.4.1.16. EcuMDriverInitItem

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMModuleID</a>	1..1
<a href="#">EcuMModuleService</a>	1..1

<b>Parameter Name</b>	<b>EcuMModuleID</b>
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>
	<p>Specify the module short name (aka module ID) of the module to be initialized; e.g. <code>Adc</code></p> <p>The name must not be empty and follow the naming conventions for valid C identifiers.</p> <p>It is assumed that there exists:</p> <ul style="list-style-type: none"> <li>▶ A header file named after the value of this parameter; e.g. <code>Adc.h</code></li> <li>▶ A initialization function (declared in the mentioned header file) prefixed with the value of this parameter; e.g. <code>Adc_Init()</code></li> </ul> <p>Restriction:</p>



	<ul style="list-style-type: none"> <li>▶ This parameter for a <code>DriverInitItem</code> in <code>EcuMDriverInitListThree</code> shall have the same value in all configurations</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

Parameter Name	<code>EcuMModuleService</code>
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>The service to be called to initialize that module, e.g. Init, PreInit, Start etc.</p> <p>If the service is Init and the parameter <code>EcuMModuleConfigurationRef</code> has been set for that module, the corresponding pointer to the init structure (&lt; Module &gt;_ConfigType) shall be passed as an argument.</p> <p>Restriction:</p> <ul style="list-style-type: none"> <li>▶ This parameter for a <code>DriverInitItem</code> in <code>EcuMDriverInitListThree</code> shall have the same value in all configurations</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.4.1.17. `EcuMDriverInitListTwo`

Containers included		
Container name	Multiplicity	Description
<a href="#"><code>EcuMDriverInitItem</code></a>	1..n	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This container describes one entry in a driver init list.</p> <p>Restriction:</p> <ul style="list-style-type: none"> <li>▶ The <code>EcuMDriverInitItem</code> name in <code>EcuMDriverInitListTwo</code> shall be the same for all EcuM configuration containers</li> </ul>



#### 5.4.1.18. EcuMDriverInitItem

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMModuleID</a>	1..1
<a href="#">EcuMModuleService</a>	1..1

<b>Parameter Name</b>	<b>EcuMModuleID</b>
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Specify the module short name (aka module ID) of the module to be initialized; e.g. <code>Adc</code></p> <p>The name must not be empty and follow the naming conventions for valid C identifiers.</p> <p>It is assumed that there exists:</p> <ul style="list-style-type: none"> <li>▶ A header file named after the value of this parameter; e.g. <code>Adc.h</code></li> <li>▶ A initialization function (declared in the mentioned header file) prefixed with the value of this parameter; e.g. <code>Adc_Init()</code></li> </ul> <p>Restriction:</p> <ul style="list-style-type: none"> <li>▶ This parameter for a <code>DriverInitItem</code> in <code>EcuMDriverInitListTwo</code> shall have the same value in all configurations</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b>   VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMModuleService</b>
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>The service to be called to initialize that module, e.g. Init, PreInit, Start etc.</p> <p>If the service is Init and the parameter <code>EcuMModuleConfigurationRef</code> has been set for that module, the corresponding pointer to the init structure (&lt;Module&gt;_ConfigType) shall be passed as an argument.</p> <p>Restriction:</p>



	▶ This parameter for a <code>DriverInitItem</code> in <code>EcuMDriverInitListTwo</code> shall have the same value in all configurations
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.4.1.19. EcuMFixedUserConfig

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMFixedUser</a>	1..1
<b>Parameter Name</b>	<b>EcuMFixedUser</b>
<b>Description</b>	Parameter used to identify one user. A unique ID needs to be provided for each user. The ID also has to be zero-based and consecutive.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.4.1.20. EcuMTTII

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMDivisor</a>	1..1
<a href="#">EcuMSleepModeRef</a>	1..1
<a href="#">EcuMSuccessorRef</a>	0..1
<b>Parameter Name</b>	<b>EcuMDivisor</b>
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Initial value for the Divisor Counter of this TTII step.</p>



	<p>This Counter is decremented every time the ECU has been woken up by the TTII Wakeup Source. When this counter has elapsed the succeeding TTII step is executed.</p> <p><i>This parameter is not used in this version of the EcuM.</i></p> <p>Restriction:</p> <ul style="list-style-type: none"> <li>▶ This parameter for an EcuM TTII shall have the same value in all configurations</li> <li>▶ The maximal value for <code>EcuMDivisor</code> is 18446744073709551615 (= 0xFFFFFFFFFFFFFF = <math>(2^{64}-1)</math>) according to EcuM configuration but EB tresos Studio limits the maximum value to 9223372036854775807 (= 0x7FFFFFFFFFFFF = <math>(2^{63}-1)</math>).</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Range</b>	<=9223372036854775807 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMSleepModeRef</b>
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This configuration parameter is a reference to a configured sleep mode that is used for TTII.</p> <p>Restriction:</p> <ul style="list-style-type: none"> <li>▶ This parameter for an EcuM TTII shall have the same value in all configurations</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMSuccessorRef</b>
-----------------------	-------------------------



<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This parameter is a reference to the next sleep mode in the TTII protocol.</p> <p>Restriction:</p> <ul style="list-style-type: none"> <li>▶ This parameter for an Ecum TTII shall have the same value in all configurations</li> </ul>
<b>Multiplicity</b>	0..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.4.1.21. EcumWdgM

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcumWdgMPostRunModeRef</a>	1..1
<a href="#">EcumWdgMRunModeRef</a>	1..1
<a href="#">EcumWdgMShutdownModeRef</a>	1..1
<a href="#">EcumWdgMStartupModeRef</a>	1..1
<a href="#">EcumWdgMWakeupModeRef</a>	1..1
<a href="#">EcumMSupervisedEntityRef</a>	1..1

<b>Parameter Name</b>	<b>EcumWdgMPostRunModeRef</b>
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This parameter references the WdgM mode to be set when entering the POST RUN state of Ecum.</p>
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC
<b>Parameter Name</b>	<b>EcumWdgMRunModeRef</b>



<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>
	This parameter references the WdgM mode to be set when entering the RUN state of EcuM.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMWdgMShutdownModeRef</b>
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>
	This parameter references the WdgM mode to be set when leaving the GO OFF I state of EcuM.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMWdgMStartupModeRef</b>
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>
	This parameter references the WdgM mode to be set when entering the START-UP II state of EcuM.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMWdgMWakeupModeRef</b>
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>
	This parameter references the WdgM mode to be set when entering the WAKE-UP I state of EcuM.



<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC
<b>Parameter Name</b>	<b>EcuMSupervisedEntityRef</b>
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This parameter references the Supervised Entity ID that way configured for EcuM in the Watchdog Manager.</p> <p>The EcuM updates this associated alive counter cyclically after it has initialized the WdgM in all states except SLEEP. The cyclic update process is suspended in the SLEEP mode but will be activated after the SLEEP mode is left.</p> <p>Configure the supervised entity of the EcuM in the WdgM so that the WdgM causes a reset if the alive counter is not updated in the WdgM modes WdgMS-tartupMode, WdgMWakeupMode, WdgMRunMode, and WdgMPostRunMode. During the time in which the WdgMShutdownMode is active no alive counter is updated any more because the WdgM is not supervising the counter any more.</p>
<b>Multiplicity</b>	1..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.4.1.22. EcuMFlexConfiguration

Containers included		
Container name	Multiplicity	Description
<a href="#">EcuMAlarmClock</a>	0..n	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>These containers describe the configured alarm clocks. The name of these containers allows giving a symbolic name to one alarm clock.</p>
<a href="#">EcuMFlexUserConfig</a>	1..n	Describes the identifiers that are needed to refer to a particular software component or another appropriate entity in the system which uses the EcuMFlex Interfaces.



<b>Containers included</b>		
<a href="#">EcuMGoDownAllowedUsers</a>	0..1	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This container describes the collection of allowed users which are allowed to call the EcuM_GoDown API.</p>
<a href="#">EcuMResetMode</a>	1..n	<p>These containers describe the configured reset modes. The name of these containers allows one of the following symbolic names to be given to the different reset modes:</p> <ul style="list-style-type: none"> <li>▶ ECUM_RESET MCU</li> <li>▶ ECUM_RESET_WDGM</li> <li>▶ ECUM_RESET_IO</li> </ul>
<a href="#">EcuMSetClockAllowedUsers</a>	0..1	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This container describes the collection of allowed users, which are allowed to call the EcuM_SetClock() API.</p>
<a href="#">EcuMShutdownCause</a>	4..255	<p>These containers describe the configured shut down or reset causes. The name of these containers allows to give one of the following symbolic names to the different shut down causes</p> <ul style="list-style-type: none"> <li>▶ ECUM_CAUSE_ECU_STATE - ECU state machine entered a state for shutdown</li> <li>▶ ECUM_CAUSE_WDGM - WdgM detected failure</li> <li>▶ ECUM_CAUSE_DCM - Dcm requests shutdown</li> <li>▶ and values from configuration</li> </ul>
<a href="#">EcuMShutdownTarget</a>	1..3	<p>These containers describe the configured shut down targets. The name of these containers allows to give symbolic names to the different shutdown targets.</p> <p>Current implementation supports only three predefined shutdown targets - ECUM_STATE_SLEEP, ECUM_STATE_RESET, ECUM_STATE_OFF.</p>

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMFlexModuleConfigurationRef</a>	0..n
<a href="#">EcuMNormalMcuModeRef</a>	1..1



<b>Parameter Name</b>	<b>EcuMFlexModuleConfigurationRef</b>	
<b>Description</b>	<p>This parameter contains a reference to the init structure of the corresponding BSW module.</p> <p>Dependency:</p> <ul style="list-style-type: none"> <li>▶ The reference in the list shall not be repeated.</li> </ul>	
<b>Multiplicity</b>	0..n	
<b>Type</b>	CHOICE-REFERENCE	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>EcuMNormalMcuModeRef</b>	
<b>Label</b>	Normal Mcu Mode	
<b>Description</b>	<p>This parameter is a reference to the normal MCU mode to be restored after a sleep.</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

#### 5.4.1.23. EcuMAAlarmClock

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMAAlarmClockId</a>	1..1
<a href="#">EcuMAAlarmClockTimeOut</a>	1..1
<a href="#">EcuMAAlarmClockUser</a>	1..1

<b>Parameter Name</b>	<b>EcuMAAlarmClockId</b>
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>
	Unique ID for this alarmclock.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER



<b>Default value</b>	0
<b>Range</b>	<=255 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMAAlarmClockTimeOut</b>
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Parameter allows to define a timeout in seconds for this alarm clock.
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMAAlarmClockUser</b>
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Parameter allows an alarm to be assigned to the user.
<b>Multiplicity</b>	1..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.4.1.24. EcuMFlexUserConfig

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMFlexUser</a>	1..1
<a href="#">EcuMFlexEcucPartitionRef</a>	0..1
<b>Parameter Name</b>	<b>EcuMFlexUser</b>



<b>Description</b>	Parameter used to identify one user.
	Restriction:
	▶ User ID shall be in the range 0 to 255 and shall be unique
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	EcuMFlexEcucPartitionRef
<b>Description</b>	Denotes in which "EcucPartition" the user of the EcuM is executed.  This reference parameter has no effect as it is only used by the Alarm Clock functionality which is not supported by the current implementation.
<b>Multiplicity</b>	0..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.4.1.25. EcuMGoDownAllowedUsers

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMGoDownAllowedUserRef</a>	1..n

<b>Parameter Name</b>	EcuMGoDownAllowedUserRef
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  These parameters describe the references to users which are allowed to call the EcuM_GoDown API.
<b>Multiplicity</b>	1..n
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC



#### 5.4.1.26. EcuMResetMode

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMResetModelId</a>	1..1

<b>Parameter Name</b>	<b>EcuMResetModelId</b>	
<b>Description</b>	This ID identifies the reset mode in services like <code>EcuM_SelectShutdownTarget()</code> .	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

#### 5.4.1.27. EcuMSetClockAllowedUsers

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMSetClockAllowedUserRef</a>	1..n

<b>Parameter Name</b>	<b>EcuMSetClockAllowedUserRef</b>	
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>References to the users which are allowed to call the <code>EcuM_SetClock()</code> API.</p>	
<b>Multiplicity</b>	1..n	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

#### 5.4.1.28. EcuMShutdownCause

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>

**Parameters included**

<a href="#">EcuMShutdownCausId</a>	1..1
------------------------------------	------

<b>Parameter Name</b>	<b>EcuMShutdownCausId</b>
<b>Description</b>	Unique ID for identifying this shut down cause.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

**5.4.1.29. EcuMShutdownTarget****Parameters included**

Parameter name	Multiplicity
<a href="#">EcuMShutdownTargetId</a>	1..1

<b>Parameter Name</b>	<b>EcuMShutdownTargetId</b>
<b>Description</b>	This ID identifies this shut down target in services like <code>EcuM_SelectShutdownTarget()</code> .
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Range</b>	<=255 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

**5.4.1.30. EcuMDefensiveProgramming****Parameters included**

Parameter name	Multiplicity
<a href="#">EcuMDefProgEnabled</a>	1..1

**Parameters included**

<a href="#">EcuMPrecondAssertEnabled</a>	1..1
<a href="#">EcuMPostcondAssertEnabled</a>	1..1
<a href="#">EcuMStaticAssertEnabled</a>	1..1
<a href="#">EcuMUreachAssertEnabled</a>	1..1
<a href="#">EcuMInvariantAssertEnabled</a>	1..1

<b>Parameter Name</b>	<b>EcuMDefProgEnabled</b>	
<b>Label</b>	Enable Defensive Programming	
<b>Description</b>	<p>Enables or disables the defensive programming feature for the module EcuM.</p> <p>Note: This feature is dependent on the use of the development error detection module. To use the defensive programming feature, proceed as follows:</p> <ol style="list-style-type: none"> <li>1. Enable development error detection</li> <li>2. Enable defensive programming</li> <li>3. Enable assertions as required</li> </ol>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<a href="#">VariantPostBuild:</a>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EcuMPrecondAssertEnabled</b>	
<b>Label</b>	Enable Precondition Assertions	
<b>Description</b>	<p>Enables handling of precondition assertion checks reported from the module EcuM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EcuMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EcuMDefProgEnabled</code>): must be enabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EcuMPostcondAssertEnabled</b>	
<b>Label</b>	Enable Postcondition Assertions	
<b>Description</b>	<p>Enables handling of postcondition assertion checks reported from the module EcuM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EcuMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EcuMDefProgEnabled</code>): must be enabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EcuMStaticAssertEnabled</b>	
<b>Label</b>	Enable Static Assertions	
<b>Description</b>	<p>Enables handling of static assertion checks reported from the module EcuM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EcuMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EcuMDefProgEnabled</code>): must be enabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EcuMUnreachAssertEnabled</b>	



<b>Label</b>	Enable Unreachable Code Assertions
<b>Description</b>	<p>Enables handling of unreachable code assertion checks reported from the module EcuM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EcuMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EcuMDefProgEnabled</code>): must be enabled</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EcuMInvariantAssertEnabled</b>	
<b>Label</b>	Enable Invariant Assertions	
<b>Description</b>	<p>Enables handling of invariant assertion checks reported from functions of the module EcuM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EcuMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EcuMDefProgEnabled</code>): must be enabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild	
<b>Origin</b>	Elektrobit Automotive GmbH	

#### 5.4.1.31. EcuMFixedGeneral

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>

**Parameters included**

<a href="#">EcuMIncludeComM</a>	1..1
<a href="#">EcuMIncludeNvM</a>	1..1
<a href="#">EcuMIncludeNvramMgr</a>	1..1
<a href="#">EcuMIncludeWdgM</a>	1..1
<a href="#">EcuMTTIIEnabled</a>	1..1
<a href="#">EcuMTTIIWakeupSourceRef</a>	1..1

**Parameter Name****EcuMIncludeComM****Description**

*The functionality related to this parameter is not supported by the current implementation.*

This configuration parameter defines whether the communication manager is supported by EcuM. This feature is presented for development purpose to compile out the communication manager in the early debugging phase.

**Multiplicity**

1..1

**Type**

BOOLEAN

**Default value**

false

**Configuration class****VariantPostBuild:**

VariantPostBuild

**Origin**

AUTOSAR\_ECUC

**Parameter Name****EcuMIncludeNvM****Description**

*The functionality related to this parameter is not supported by the current implementation.*

This configuration parameter defines whether the non volatile memory manager is supported by EcuM. This feature is presented for development purpose to compile out the volatile memory manager in the early debugging phase.

**Multiplicity**

1..1

**Type**

BOOLEAN

**Default value**

false

**Configuration class****VariantPostBuild:**

VariantPostBuild

**Origin**

AUTOSAR\_ECUC

**Parameter Name****EcuMIncludeNvramMgr****Description**

*The functionality related to this parameter is not supported by the current implementation.*



	<p>If this parameter is enabled, the EcuM will be compiled with NvM (Non-Volatile RAM manager) support; the EcuM will initialize the NvRam-manager and also implement safe start-up and shutdown of the NvRam, regarding the consistency of NvRam caches and data.</p> <p>Dependency:</p> <ul style="list-style-type: none"> <li>▶ NvM must be initialized in init list 2.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMIncludeWdgM</b>
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This configuration parameter defines whether the watchdog manager is supported by EcuM. This feature is presented for development purpose to compile out the watchdog manager in the early debugging phase</p> <p>Dependency:</p> <ul style="list-style-type: none"> <li>▶ WdgM shall be in the init list.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMTTIIEnabled</b>
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Switch to enable or disable Time Triggered Increased Inoperation (TTII).</p>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>EcuMTTIIWakeupSourceRef</b>	
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>References to the initial sleep mode to be used by TTII when TTII is activated after a RUN mode.</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

#### 5.4.1.32. EcuMFlexGeneral

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMAlarmClockPresent</a>	1..1
<a href="#">EcuMEnableDefBehaviour</a>	0..1
<a href="#">EcuMResetLoopDetection</a>	1..1
<a href="#">EcuMAlarmWakeupSource</a>	0..1
<a href="#">EcuMStoredShutdownOperations</a>	1..1

<b>Parameter Name</b>	<b>EcuMAlarmClockPresent</b>	
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Flag indicates whether the optional AlarmClock feature is present.</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	
<b>Parameter Name</b>	<b>EcuMEnableDefBehaviour</b>	



<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i> Switches the defensive behaviour on or off.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	
<b>Parameter Name</b>	<b>EcuMResetLoopDetection</b>	
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i> If this flag is false, no reset loop detection is performed.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	
<b>Parameter Name</b>	<b>EcuMAlarmWakeupSource</b>	
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i> This parameter describes the reference to the EcuMWakeupSource being used for the EcuM AlarmClock.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	
<b>Parameter Name</b>	<b>EcuMStoredShutdownOperations</b>	
<b>Description</b>	<i>This parameter denotes the number of shutdown operations for which information are stored in the EcuM for later retrieval.</i> The parameter is not specified by AUTOSAR but it is an EB specific enhancement to the EcuM	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	



<b>Default value</b>	1
<b>Range</b>	<=255 >=1
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

#### 5.4.1.33. EcuMGeneral

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">EcuMServiceAPI</a>	1..1	<p><b>Label:</b> Service API Parameters</p> <p>Container for configuration of the service API of EcuM.</p> <p>Check "Enable Rte Usage" in order to enable this configuration item.</p>

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMDevErrorDetect</a>	1..1
<a href="#">EcuMIncludeDem</a>	1..1
<a href="#">EcuMIncludeDet</a>	1..1
<a href="#">EcuMMainFunctionPeriod</a>	1..1
<a href="#">EcuMProvideShutdownHook</a>	1..1
<a href="#">EcuMRelocatablePbcfgEnable</a>	1..1
<a href="#">EcuMRteUsage</a>	1..1
<a href="#">EcuMEnableMulticore</a>	1..1
<a href="#">EcuMMasterCoreId</a>	0..1
<a href="#">EcuMInitialCoreId</a>	0..1
<a href="#">EcuMStartInitialCore</a>	0..1
<a href="#">EcuMUseBoottargetSec</a>	1..1
<a href="#">EcuMVersionInfoApi</a>	1..1
<a href="#">EcuMCrossCoreComMWakeup</a>	1..1

<b>Parameter Name</b>	<b>EcuMDevErrorDetect</b>
-----------------------	---------------------------



<b>Label</b>	Enable Development Error Detection	
<b>Description</b>	<p>Enable development error detection. If false, no debug artifacts (e.g. calls to Det) remain in the executable object. Initialization of Det, however is controlled by configuration of optional BSW modules.</p> <p>Dependency:</p> <ul style="list-style-type: none"> <li>▶ EcuMIncludeDet has to be enabled.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	true	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>EcuMIncludeDem</b>	
<b>Label</b>	Include Dem	
<b>Description</b>	<p>This parameter must be enabled to report production errors to Diagnostics Event Manager(DEM). Dem must be initialized before reporting any errors to it. The initialization consist of <code>Dem_PreInit()</code> and <code>Dem_Init()</code>.</p> <p><code>Dem_PreInit()</code> shall be configured in <code>EcuMDriverInitListOne</code> with proper post build parameter. Further initialization of DEM by calling <code>Dem_Init()</code> is controlled by BswM module.</p> <p>Configure <code>Dem_PreInit()</code> in <code>EcuMDriverInitListOne</code>:</p> <ul style="list-style-type: none"> <li>▶ EcuMModuleID : Dem</li> <li>▶ EcuMModuleService : PreInit</li> <li>▶ EcuMModuleInitConfigStr : disable and let <code>EcuMFlexModuleConfigurationRef</code> refer to the Dem configuration container</li> <li>▶ EcuMModuleHeaderFile : leave field blank</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>EcuMIncludeDet</b>
-----------------------	-----------------------



<b>Label</b>	Include Det
<b>Description</b>	<p>According to AUTOSAR this parameter controls the initialization of the Det (Development Error Tracer) module. But in the current EcuM implementation the initialization and the start if the Det is controlled via the initialization lists just like for other AUTOSAR modules.</p> <p>Dependency:</p> <ul style="list-style-type: none"> <li>▶ Det must have an entry for its initialization and start in one of the initialization lists.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	true
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMMainFunctionPeriod</b>
<b>Description</b>	This parameter defines the schedule period of <code>EcuM_MainFunction</code> . Unit: [s]
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.02
<b>Range</b>	<=255 >=0.001
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EcuMProvideShutdownHook</b>
<b>Label</b>	Provide ShutdownHook() implementation
<b>Description</b>	<p>This EB specific parameter enables the provision of a minimalistic implementation of the <code>ShutdownHook()</code> function for the Os.</p> <p>The <code>ShutdownHook()</code> implementation provided by default only calls <code>EcuM_Shutdown()</code>. If you need a more complex implementation of the <code>ShutdownHook()</code> function, disable this parameter and provide an application specific implementation.</p>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN



<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EcuMRelocatablePbcfgEnable</b>	
<b>Label</b>	EcuMRelocatablePbcfgEnable	
<b>Description</b>	<p>Enables/disables support for relocatable postbuild configuration.</p> <ul style="list-style-type: none"> <li>▶ True: Postbuild configuration relocatable in memory.</li> <li>▶ False: Postbuild configuration not relocatable in memory.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	true	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EcuMRteUsage</b>	
<b>Label</b>	Enable Rte Usage	
<b>Description</b>	<p>This parameter enables the usage of the RTE for this module.</p> <p>For an easy integration it is recommended to disable the usage of the RTE at the beginning of the integration work.</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EcuMEnableMulticore</b>	
<b>Label</b>	Enable Multi-core functionality	
<b>Description</b>	<p><i>This parameter enables the multicore functionality.</i></p> <p>If multiple Os configurations are used, this parameter must always be set to true.</p> <p>The parameter is not specified by AUTOSAR but it is an EB specific enhancement to the Ecum</p>	
<b>Multiplicity</b>	1..1	



Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild:
Origin	Elektrobit Automotive GmbH

Parameter Name	EcuMMasterCoreId
Label	EcuM Master Core ID
Description	<i>This parameter maps the EcuM master core instance to a specific Os Core ID.</i>
Multiplicity	0..1
Type	INTEGER
Default value	0
Configuration class	PreCompile: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EcuMInitialCoreId
Label	EcuM Initial Core ID
Description	<p>The ID of the initial core that is started when the system boots up.</p> <p>This parameter can be used to specify the logical core identifier of the core which is initially started when the system boots up.</p> <p><i>Note:</i> If this parameter is not configured, the EcuM assumes that the initial core is the core designated by OS_CORE_ID_MASTER</p>
Multiplicity	0..1
Type	INTEGER
Default value	0
Configuration class	PreCompile: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EcuMStartInitialCore
Label	EcuM Start Initial Core
Description	<p>This parameter enables calling the OS API StartCore(coreID) upon EcuM initialization, also for the core that boots up the system.</p> <p><i>Note:</i> If this parameter is not enabled, the EcuM avoids to invoke StartCore() for the core that boots up the system, since it is already started by the hardware and for this reason some Os solutions are triggering an error.</p>



<b>Multiplicity</b>	0..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	PreCompile:	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EcuMUseBoottargetSec</b>	
<b>Label</b>	Enable Boottarget Section	
<b>Description</b>	This parameter enables the mapping of the variable EcuM_BootTarget to the memory section SEC_ECUM_BOOTTARGET instead of SEC_VAR_CLEARED_UNSPECIFIED. This might be needed if you access EcuM_BootTarget for a bootloader or boot manager. Please see the comments in EcuM_BootTarget.h for implementation details.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	VariantPostBuild:	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EcuMVersionInfoApi</b>	
<b>Label</b>	Enable Version Info API	
<b>Description</b>	Switches the version info API on or off	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	VariantPostBuild:	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>EcuMCrossCoreComMWakeup</b>	
<b>Label</b>	Enable EcuM to ComM Cross Core Support	
<b>Description</b>	Enable EcuM to ComM cross core multicore support. (client server connections in RTE for interfaces ComM_EcuM_WakeUpIndication and ComM_EcuM_PNCWakeUpIndication).  In case EcuM wakeup events are executed on EcuM slave cores or master cores and these cores are not the same as the core on which ComM runs, EcuM	



	<p>calls directly ComM_EcuM_WakeUpIndication and ComM_EcuM_PNCWakeUpIndication which shall be avoided.</p> <p>Enabling this parameter avoid this and triggers the ComM_EcuM_WakeupIndication API in the context of the same core as ComM.</p> <p>Note that also ComMMultiCoreSupport and EcuMEnableMulticore has to be enabled in order to have Rte Connections.</p>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

#### 5.4.1.34. EcuMServiceAPI

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMEnableASR32ServiceAPI</a>	1..1
<a href="#">EcuMEnableASR40ServiceAPI</a>	1..1
<a href="#">EcuMDefaultASRServiceAPI</a>	1..1

<b>Parameter Name</b>	<b>EcuMEnableASR32ServiceAPI</b>
<b>Label</b>	Enable AUTOSAR 3.2 service API
<b>Description</b>	<p>Configures whether the AUTOSAR 3.2 service API shall be provided.</p> <ul style="list-style-type: none"> <li>▶ TRUE = Enables AUTOSAR 3.2 service API.</li> <li>▶ FALSE = Disables AUTOSAR 3.2 service API.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EcuMEnableASR40ServiceAPI</b>
-----------------------	----------------------------------



<b>Label</b>	Enable AUTOSAR 4.0 service API	
<b>Description</b>	Configures whether the AUTOSAR 4.0 service API shall be provided. <ul style="list-style-type: none"> <li>▶ TRUE = Enables AUTOSAR 4.0 service API.</li> <li>▶ FALSE = Disables AUTOSAR 4.0 service API.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EcuMDefaultASRServiceAPI</b>	
<b>Label</b>	Default AUTOSAR service API	
<b>Description</b>	Defines the default AUTOSAR service API. <ul style="list-style-type: none"> <li>▶ AUTOSAR_32 = AUTOSAR 3.2 service API is the default one.</li> <li>▶ AUTOSAR_40 = AUTOSAR 4.0 service API is the default one.</li> <li>▶ NONE = No default AUTOSAR service API is provided.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	AUTOSAR_40	
<b>Range</b>	AUTOSAR_32 AUTOSAR_40 NONE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

#### 5.4.1.35. ReportToDem

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EcuMRamChkFailedReportToDem</a>	1..1
<a href="#">EcuMRamChkFailedReportToDemDetErrorId</a>	1..1
<a href="#">EcuMCfgDataInconsistentReportToDem</a>	1..1

**Parameters included**

<a href="#">EcuMCfgDataInconsistentReportToDemDetErrorId</a>	1..1
--	------

<b>Parameter Name</b>	<b>EcuMRamChkFailedReportToDem</b>	
<b>Label</b>	Ram Check Failure	
<b>Description</b>	<p>Selects the handling of the production error: <i>ECUM_E_RAM_CHECK_FAILED</i></p> <ul style="list-style-type: none"> <li>▶ DEM: All errors are reported to the Diagnostics Event Manager (Dem).</li> <li>▶ DET: All errors are reported to the Development Error Tracer (Det) if enabled.</li> <li>▶ DISABLE: Production errors are not reported at all.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	DEM	
<b>Range</b>	DEM DET DISABLE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EcuMRamChkFailedReportToDemDetErrorId</b>	
<b>Label</b>	EcuM Ram Check Failure DemToDet ErrorId	
<b>Description</b>	<p>This parameter is used to report Dem to Det. It shall be checked that the variable <code>EcuMRamChkFailedReportToDemDetErrorId</code> is set to a value between 30 and 255; the default value shall be 30. A preprocessor define <code>ECUM_E_DEMTODET_RAM_CHECK_FAILED</code> shall be generated holding the value of <code>EcuMRamChkFailedReportToDemDetErrorId</code>.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ The identifier to report Dem to Det must not be zero-based.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	30	
<b>Range</b>	<=255 >=30	



<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EcuMCfgDataInconsistentReportToDem</b>	
<b>Label</b>	Inconsistent Configuration Data	
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Selects the handling of the production error: <code>ECUM_E_CONFIGURATION_DATA_INCONSISTENT</code></p> <ul style="list-style-type: none"> <li>▶ <code>DEM</code>: All errors are reported to the Diagnostics Event Manager (Dem).</li> <li>▶ <code>DET</code>: All errors are reported to the Development Error Tracer (Det) if enabled.</li> <li>▶ <code>DISABLE</code>: Production errors are not reported at all.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	DISABLE	
<b>Range</b>	DEM DET DISABLE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EcuMCfgDataInconsistentReportToDemDetErrorId</b>	
<b>Label</b>	EcuM Configuration Data Inconsistent DemToDet ErrorId	
<b>Description</b>	<p>This parameter is used to report Dem to Det. It shall be checked that the variable <code>EcuMCfgDataInconsistentReportToDemDetErrorId</code> is set to a value between 30 and 255; the default value shall be 32. A preprocessor define <code>ECUM_E_DEMTODET_CONFIGURATION_DATA_INCONSISTENT</code> shall be generated holding the value of <code>EcuMCfgDataInconsistentReportToDemDetErrorId</code>.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ The identifier to report Dem to Det must not be zero-based.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	



<b>Default value</b>	32
<b>Range</b>	<=255 >=30
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

#### 5.4.1.36. PublishedInformation

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">PbcfgMSupport</a>	1..1
<b>Parameter Name</b>	
<b>PbcfgMSupport</b>	
<b>Label</b>	PbcfgM support
<b>Description</b>	Specifies whether or not the EcuM can use the PbcfgM module for post-build support.
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

## 5.4.2. Recommended configurations

### 5.4.2.1. EcuMRecConfiguration

<b>Containers included</b>	
<b>Container name</b>	<b>Container definition</b>
<a href="#">EcuM_Config_0</a>	<a href="#">EcuMConfiguration</a>
<b>Parameters included</b>	
<b>Parameter name</b>	<b>Value</b>



#### 5.4.2.1.1. EcuM\_Config\_0

<b>Containers included</b>	
<b>Container name</b>	<b>Container definition</b>
<a href="#">EcuMFlexConfiguration</a>	<a href="#">EcuMFlexConfiguration</a>
<a href="#">EcuMCommonConfiguration</a>	<a href="#">EcuMCommonConfiguration</a>

#### Parameters included

<b>Parameter name</b>	<b>Value</b>

#### 5.4.2.1.2. EcuMFlexConfiguration

<b>Containers included</b>	
<b>Container name</b>	<b>Container definition</b>
<a href="#">ECUM_CAUSE_UNKNOWN</a>	<a href="#">EcuMShutdownCause</a>
<a href="#">ECUM_CAUSE_ECU_STATE</a>	<a href="#">EcuMShutdownCause</a>
<a href="#">ECUM_CAUSE_WDGM</a>	<a href="#">EcuMShutdownCause</a>
<a href="#">ECUM_CAUSE_DCM</a>	<a href="#">EcuMShutdownCause</a>
<a href="#">ECUM_RESET MCU</a>	<a href="#">EcuMResetMode</a>
<a href="#">ECUM_RESET_WDG</a>	<a href="#">EcuMResetMode</a>
<a href="#">ECUM_RESET_IO</a>	<a href="#">EcuMResetMode</a>
<a href="#">ECUM_STATE_SLEEP</a>	<a href="#">EcuMShutdownTarget</a>
<a href="#">ECUM_STATE_OFF</a>	<a href="#">EcuMShutdownTarget</a>
<a href="#">ECUM_STATE_RESET</a>	<a href="#">EcuMShutdownTarget</a>

#### Parameters included

<b>Parameter name</b>	<b>Value</b>

#### 5.4.2.1.3. ECUM\_CAUSE\_UNKNOWN

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Value</b>
<a href="#">EcuMShutdownCauseld</a>	0



#### 5.4.2.1.4. ECUM\_CAUSE\_ECU\_STATE

Parameters included	
Parameter name	Value
<a href="#">EcuMShutdownCausId</a>	1

#### 5.4.2.1.5. ECUM\_CAUSE\_WDGM

Parameters included	
Parameter name	Value
<a href="#">EcuMShutdownCausId</a>	2

#### 5.4.2.1.6. ECUM\_CAUSE\_DCM

Parameters included	
Parameter name	Value
<a href="#">EcuMShutdownCausId</a>	3

#### 5.4.2.1.7. ECUM\_RESET MCU

Parameters included	
Parameter name	Value
<a href="#">EcuMResetModelId</a>	0

#### 5.4.2.1.8. ECUM\_RESET\_WDG

Parameters included	
Parameter name	Value
<a href="#">EcuMResetModelId</a>	1

#### 5.4.2.1.9. ECUM\_RESET\_IO

Parameters included	
Parameter name	Value



#### Parameters included

<a href="#">EcuMResetModeld</a>	2
---------------------------------	---

#### 5.4.2.1.10. ECUM\_STATE\_SLEEP

#### Parameters included

Parameter name	Value
<a href="#">EcuMShutdownTargetId</a>	80

#### 5.4.2.1.11. ECUM\_STATE\_OFF

#### Parameters included

Parameter name	Value
<a href="#">EcuMShutdownTargetId</a>	128

#### 5.4.2.1.12. ECUM\_STATE\_RESET

#### Parameters included

Parameter name	Value
<a href="#">EcuMShutdownTargetId</a>	144

#### 5.4.2.1.13. EcuMCommonConfiguration

#### Containers included

Container name	Container definition
<a href="#">ECUM_WKSOURCE_POWER</a>	<a href="#">EcuMWakeupSource</a>
<a href="#">ECUM_WKSOURCE_RESET</a>	<a href="#">EcuMWakeupSource</a>
<a href="#">ECUM_WKSOURCE_INTERNAL_RESET</a>	<a href="#">EcuMWakeupSource</a>
<a href="#">ECUM_WKSOURCE_INTERNAL_WDG</a>	<a href="#">EcuMWakeupSource</a>
<a href="#">ECUM_WKSOURCE_EXTERNAL_WDG</a>	<a href="#">EcuMWakeupSource</a>

#### Parameters included

Parameter name	Value
----------------	-------



#### 5.4.2.1.14. ECUM\_WKSOURCE\_POWER

Parameters included	
Parameter name	Value
<a href="#">EcuMWakeupSourceld</a>	0
<a href="#">EcuMValidationTimeout</a>	0.0

#### 5.4.2.1.15. ECUM\_WKSOURCE\_RESET

Parameters included	
Parameter name	Value
<a href="#">EcuMWakeupSourceld</a>	1
<a href="#">EcuMValidationTimeout</a>	0.0

#### 5.4.2.1.16. ECUM\_WKSOURCE\_INTERNAL\_RESET

Parameters included	
Parameter name	Value
<a href="#">EcuMWakeupSourceld</a>	2
<a href="#">EcuMValidationTimeout</a>	0.0

#### 5.4.2.1.17. ECUM\_WKSOURCE\_INTERNAL\_WDG

Parameters included	
Parameter name	Value
<a href="#">EcuMWakeupSourceld</a>	3
<a href="#">EcuMValidationTimeout</a>	0.0

#### 5.4.2.1.18. ECUM\_WKSOURCE\_EXTERNAL\_WDG

Parameters included	
Parameter name	Value



#### Parameters included

<a href="#">EcuMWakeupSourceId</a>	4
<a href="#">EcuMValidationTimeout</a>	0.0

### 5.4.3. Application programming interface (API)

#### 5.4.3.1. Type definitions

##### 5.4.3.1.1. EcuM\_ASR32\_Boot targetType

<b>Purpose</b>	Type for boot target.
<b>Type</b>	uint8
<b>Description</b>	Stores the type of the next boot target.  All valid values for this type are defined as preprocessor macros.

##### 5.4.3.1.2. EcuM\_ASR32\_StateType

<b>Purpose</b>	Type for EcuM configuration.
<b>Type</b>	uint8
<b>Description</b>	Encodes states and sub-states of the ECU State Manager. States are encoded in the hi-nibble and sub-states are encoded in the lo-nibble. The sub-state can be determined by ANDing the state value with ECUM_SUBSTATE_MASK.  All valid values for this type are defined as preprocessor macros.

##### 5.4.3.1.3. EcuM\_ASR32\_UserType

<b>Purpose</b>	Type for a user ID of the EcuM.
<b>Type</b>	uint8
<b>Description</b>	A unique User ID, which must be defined for each user at system generation time.



#### 5.4.3.1.4. EcuM\_AS40\_BootTargetType

<b>Purpose</b>	Type for boot target.
<b>Type</b>	uint8
<b>Description</b>	<p>Stores the type of the next boot target.</p> <p>All valid values for this type are defined as preprocessor macros.</p>

#### 5.4.3.1.5. EcuM\_AS40\_ShutdownCauseType

<b>Purpose</b>	Type for the cause of shutdown.
<b>Type</b>	uint8
<b>Description</b>	<p>Describes the cause for a shutdown by the ECU State Manager. It can be extended by configuration.</p> <p>All valid values for this type are defined as preprocessor macros.</p>

#### 5.4.3.1.6. EcuM\_AS40\_StateType

<b>Purpose</b>	Type for EcuM configuration.
<b>Type</b>	uint8
<b>Description</b>	<p>Encodes states and sub-states of the ECU State Manager. States are encoded in the hi-nibble and sub-states are encoded in the lo-nibble. The sub-state can be determined by ANDing the state value with ECUM_SUBSTATE_MASK.</p> <p>All valid values for this type are defined as preprocessor macros.</p>

#### 5.4.3.1.7. EcuM\_AS40\_UserType

<b>Purpose</b>	Type for a user ID of the EcuM.
<b>Type</b>	uint8
<b>Description</b>	A unique User ID, which must be defined for each user at system generation time.

#### 5.4.3.1.8. EcuM\_BootTargetInternalType

<b>Purpose</b>	Type of internal 64 Bit boot target flag.
----------------	---



Type	uint32[2]
Description	Type for storing the 64 Bit representation of the boot target flag

#### 5.4.3.1.9. EcuM\_BootTargetType

Purpose	Type for boot target.
Type	<a href="#">EcuM_AS40_BootTargetType</a>
Description	<p>Stores the type of the next boot target.</p> <p>All valid values for this type are defined as preprocessor macros.</p>

#### 5.4.3.1.10. EcuM\_DeterminePbConfigurationRetType

Purpose	Return type of callout function <a href="#">EcuM_DeterminePbConfiguration()</a> .
Type	const EcuM_ConfigType *
Description	<p>This type definition is used to prevent problematic nested compiler abstraction macros as in FUNC(P2CONST(), ..) in the declaration/definition of <a href="#">EcuM_DeterminePbConfiguration()</a>.</p>

#### 5.4.3.1.11. EcuM\_ResetType

Purpose	Type for the Reset mechanism.
Type	uint8
Description	<p>Describes the reset mechanisms supported by the ECU State Manager. It can be extended by configuration.</p> <p>All valid values for this type are defined as preprocessor macros.</p>

#### 5.4.3.1.12. EcuM\_ShutdownCauseType

Purpose	EcuM Shutdown Cause Type.
Type	<a href="#">EcuM_AS40_ShutdownCauseType</a>
Description	Provide AUTOSAR 4.0 EcuM_ShutdownCauseType as default to other BSW modules



#### 5.4.3.1.13. EcuM\_SleepModeConfigType

<b>Purpose</b>	Type for the status of a wakeup source.	
<b>Type</b>	struct	
<b>Members</b>	EcuM_WakeupSourceType wake-upSource	Sleep period for which all set wakeup sources will be enabled.
	uint8 mcuMode	Sleep mode of the MCU.  Index for the configuration tables of the MCU driver.
	boolean suspendCPU	Suspend mode flag.  A Flag which is set to true if the CPU is suspended, halted, or powered off during sleep mode. This flag must be set to false if the CPU keeps running while in sleep mode.
<b>Description</b>	This type wraps the Mcu_ModeType. This is done to avoid types of other packages in this API specification. The system designer can define a list containing elements of this type to define the different sleep modes.	

#### 5.4.3.1.14. EcuM\_StateType

<b>Purpose</b>	EcuM States Type.
<b>Type</b>	<a href="#">EcuM_AS40_StateType</a>
<b>Description</b>	Provide AUTOSAR 4.0 EcuM_StateType as default to other BSW modules

#### 5.4.3.1.15. EcuM\_SyncActionType

<b>Purpose</b>	
<b>Type</b>	uint8

#### 5.4.3.1.16. EcuM\_UserType

<b>Purpose</b>	EcuM Usere Type.
<b>Type</b>	<a href="#">EcuM_AS40_UserType</a>



<b>Description</b>	Provide AUTOSAR 4.0 EcuM_UserType as default to other BSW modules
--------------------	---

#### 5.4.3.1.17. EcuM\_WakeupReactionType

<b>Purpose</b>	
<b>Type</b>	uint8

#### 5.4.3.1.18. EcuM\_WakeupSourceType

<b>Purpose</b>	Type for wakeup sources.
<b>Type</b>	uint32
<b>Description</b>	<p>The bit field provides one bit for each wakeup source. In WAKEUP state, all bits cleared indicates that no wakeup source is known. In STARTUP state, all bits cleared indicates that no reason for restart or reset is known (in this case, ECUM_WKSOURCE_RESET shall be assumed.)</p> <p>All valid values for this type are defined as preprocessor macros. Additional values may be added at configuration time.</p>

#### 5.4.3.1.19. EcuM\_WakeupStatusType

<b>Purpose</b>	Type for the status of a wakeup source.
<b>Type</b>	uint8
<b>Description</b>	<p>Describes the possible outcomes of the WAKEUP VALIDATION state. This type may be applied to one wakeup source or a collection of wakeup sources.</p> <p>All valid values for this type are defined as preprocessor macros.</p>

### 5.4.3.2. Macro constants

#### 5.4.3.2.1. ECUM\_ABORT\_SYNC

<b>Purpose</b>	Value for EcuM_SyncActionType.
<b>Value</b>	0x01



Description	EcuM should abort synchronization of its cores
-------------	--

#### 5.4.3.2.2. ECUM\_AR\_RELEASE\_MAJOR\_VERSION

Purpose	AUTOSAR release major version.
Value	4U

#### 5.4.3.2.3. ECUM\_AR\_RELEASE\_MINOR\_VERSION

Purpose	AUTOSAR release minor version.
Value	0U

#### 5.4.3.2.4. ECUM\_AR\_RELEASE\_REVISION\_VERSION

Purpose	AUTOSAR release revision version.
Value	3U

#### 5.4.3.2.5. ECUM\_BOOT\_TARGET\_APP

Purpose	Value for EcuM_BootTargetType.
Value	0U
Description	The ECU will boot into the application.

#### 5.4.3.2.6. ECUM\_BOOT\_TARGET\_BOOTLOADER

Purpose	Value for EcuM_BootTargetType.
Value	1U
Description	The ECU will boot into the bootloader.

#### 5.4.3.2.7. ECUM\_BOOT\_TARGET\_OEM\_BOOTLOADER

Purpose	Value for EcuM_BootTargetType.
---------	--------------------------------



<b>Value</b>	1U
<b>Description</b>	The ECU will boot into the OEM bootloader.

#### 5.4.3.2.8. ECUM\_BOOT\_TARGET\_OEM\_BOOTLOADER\_INTERNAL\_0

<b>Purpose</b>	Value for 1st byte of EcuM_BootTargetInternalType.
<b>Value</b>	0x00FFA55AU
<b>Description</b>	The ECU will boot into the OEM bootloader. This internal value is written to memory to make reading of the values more reliable.

#### 5.4.3.2.9. ECUM\_BOOT\_TARGET\_OEM\_BOOTLOADER\_INTERNAL\_1

<b>Purpose</b>	Value for 2nd byte of EcuM_BootTargetInternalType.
<b>Value</b>	0x2342EB6CU

#### 5.4.3.2.10. ECUM\_BOOT\_TARGET\_SYS\_BOOTLOADER

<b>Purpose</b>	Value for EcuM_BootTargetType.
<b>Value</b>	2U
<b>Description</b>	The ECU will boot into the system supplier bootloader.

#### 5.4.3.2.11. ECUM\_BOOT\_TARGET\_SYS\_BOOTLOADER\_INTERNAL\_0

<b>Purpose</b>	Value for 1st byte of EcuM_BootTargetInternalType.
<b>Value</b>	0xEB15C001U
<b>Description</b>	The ECU will boot into the system supplier bootloader. This internal value is written to memory to make reading of the values more reliable.

#### 5.4.3.2.12. ECUM\_BOOT\_TARGET\_SYS\_BOOTLOADER\_INTERNAL\_1

<b>Purpose</b>	Value for 2nd byte of EcuM_BootTargetInternalType.
<b>Value</b>	0x1BADBABEU



#### 5.4.3.2.13. ECUM\_CAUSE\_DCM

<b>Purpose</b>	Value for EcuM_ShutdownCauseType.
<b>Value</b>	3U
<b>Description</b>	Diagnostic Communication Manager requests a shutdown due to a service request

#### 5.4.3.2.14. ECUM\_CAUSE\_ECU\_STATE

<b>Purpose</b>	Value for EcuM_ShutdownCauseType.
<b>Value</b>	1U
<b>Description</b>	ECU state machine entered a state for shutdown

#### 5.4.3.2.15. ECUM\_CAUSE\_UNKNOWN

<b>Purpose</b>	Value for EcuM_ShutdownCauseType.
<b>Value</b>	0U
<b>Description</b>	No cause was set.

#### 5.4.3.2.16. ECUM\_CAUSE\_WDGM

<b>Purpose</b>	Value for EcuM_ShutdownCauseType.
<b>Value</b>	2U
<b>Description</b>	Watchdog Manager detected a failure

#### 5.4.3.2.17. ECUM\_CONTINUE\_SYNC

<b>Purpose</b>	Value for EcuM_SyncActionType.
<b>Value</b>	0x00
<b>Description</b>	EcuM should continue synchronizing its cores before the sleep/shutdown process

#### 5.4.3.2.18. ECUM\_E\_COMMPCWAKEUPINDICATION

<b>Purpose</b>	Development Error Code.
----------------	-------------------------



<b>Value</b>	0x1DU
<b>Description</b>	The Schm ComM_Ecum_PNCWakeup function returned a value different than SCHM_E_OK.

#### 5.4.3.2.19. ECUM\_E\_COMMWAKEUPINDICATION

<b>Purpose</b>	Development Error Code.
<b>Value</b>	0x1CU
<b>Description</b>	The Schm ComM_Ecum_Wakeup function returned a value different than SCHM_E_-OK.

#### 5.4.3.2.20. ECUM\_E\_INVALID\_PAR

<b>Purpose</b>	Development Error Code.
<b>Value</b>	0x13U
<b>Description</b>	A parameter was invalid (unspecific).

#### 5.4.3.2.21. ECUM\_E\_MISMATCHED\_RUN\_RELEASE

<b>Purpose</b>	Development Error Code.
<b>Value</b>	0x15U
<b>Description</b>	RUN / POSTRUN was released though it was not requested.

#### 5.4.3.2.22. ECUM\_E\_MULTIPLE\_RUN\_REQUESTS

<b>Purpose</b>	Development Error Code.
<b>Value</b>	0x14U
<b>Description</b>	RUN / POSTRUN was requested multiple times by the same user.

#### 5.4.3.2.23. ECUM\_E\_NULL\_POINTER

<b>Purpose</b>	Development Error Code.
----------------	-------------------------



<b>Value</b>	0x12U
<b>Description</b>	A null pointer was passed as an argument.

#### 5.4.3.2.24. ECUM\_E\_PARAM\_POINTER

<b>Purpose</b>	Development Error Code.
<b>Value</b>	0x03U
<b>Description</b>	API service called with a NULL pointer.

#### 5.4.3.2.25. ECUM\_E RTE\_SWITCH\_FAILED

<b>Purpose</b>	Development Error Code.
<b>Value</b>	0x18U
<b>Description</b>	Failed to report EcuM mode switch to RTE.

#### 5.4.3.2.26. ECUM\_E\_SERVICE\_DISABLED

<b>Purpose</b>	Development Error Code.
<b>Value</b>	0x11U
<b>Description</b>	A service was called which was disabled by configuration.

#### 5.4.3.2.27. ECUM\_E\_STATE\_PAR\_OUT\_OF\_RANGE

<b>Purpose</b>	Development Error Code.
<b>Value</b>	0x16U
<b>Description</b>	A state, passed as an argument to a service, was out of range (specific parameter test)

#### 5.4.3.2.28. ECUM\_E\_UNINIT

<b>Purpose</b>	Development Error Code.
----------------	-------------------------



<b>Value</b>	0x10U
<b>Description</b>	A service was called prior to initialization.

#### 5.4.3.2.29. ECUM\_E\_UNKNOWN\_WAKEUP\_SOURCE

<b>Purpose</b>	Development Error Code.
<b>Value</b>	0x17U
<b>Description</b>	An unknown wakeup source was passed as a parameter to an API.

#### 5.4.3.2.30. ECUM\_E\_UNSUCCESSFUL\_CORE\_START

<b>Purpose</b>	Development Error Code.
<b>Value</b>	0x1AU
<b>Description</b>	The Os core has not been successfully started.

#### 5.4.3.2.31. ECUM\_E\_WRONG\_CONFIG\_PARAM

<b>Purpose</b>	Development Error Code.
<b>Value</b>	0x1BU
<b>Description</b>	The provided configuration is inconsistent.

#### 5.4.3.2.32. ECUM\_E\_WRONG\_CONTEXT

<b>Purpose</b>	Development Error Code.
<b>Value</b>	0x19U
<b>Description</b>	The API has been called from an invalid core context.

#### 5.4.3.2.33. ECUM\_MODULE\_ID

<b>Purpose</b>	AUTOSAR module identification.
<b>Value</b>	10U



#### 5.4.3.2.34. ECUM\_RAM\_CHECK\_FAILED

<b>Purpose</b>	Error reason value for <a href="#">EcuM_ErrorHook()</a> .
<b>Value</b>	0x02U
<b>Description</b>	The RAM check routine returned a failure

#### 5.4.3.2.35. ECUM\_SID\_CLEAR\_WK\_EV

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	22U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_ClearWakeupEvent()</a> .

#### 5.4.3.2.36. ECUM\_SID\_DOWN

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	31U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_GoDown()</a>

#### 5.4.3.2.37. ECUM\_SID\_GET\_BOOT\_TAR

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	19U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_GetBootTarget()</a> .

#### 5.4.3.2.38. ECUM\_SID\_GET\_EXP\_WK\_EV

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	25U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_GetExpiredWakeupEvent()</a> .

#### 5.4.3.2.39. ECUM\_SID\_GET\_LAST\_SHUT\_TAR

<b>Purpose</b>	AUTOSAR API service ID.
----------------	-------------------------



<b>Value</b>	8U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_GetLastShutdownTarget()</a> .

#### 5.4.3.2.40. ECUM\_SID\_GET\_NXT\_RCNT\_SHUT\_CAUSE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	30U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_GetNextRecentShutdown()</a> .

#### 5.4.3.2.41. ECUM\_SID\_GET\_PEND\_WK\_EV

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	13U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_GetPendingWakeupEvent()</a> .

#### 5.4.3.2.42. ECUM\_SID\_GET\_RCNT\_SHUT\_CAUSE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	29U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_GetMostRecentShutdown()</a> .

#### 5.4.3.2.43. ECUM\_SID\_GET\_SHUT\_CAUSE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	28U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_GetShutdownCause()</a> .

#### 5.4.3.2.44. ECUM\_SID\_GET\_SHUT\_TAR

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	9U



Description	Definition of service ID for <a href="#">EcuM_GetShutdownTarget()</a> .
-------------	---

#### 5.4.3.2.45. ECUM\_SID\_GET\_STAT\_OF\_WK\_SRC

Purpose	AUTOSAR API service ID.
Value	23U
Description	Definition of service ID for <a href="#">EcuM_GetStatusOfWakeupSource()</a> .

#### 5.4.3.2.46. ECUM\_SID\_GET\_VALID\_WK\_EV

Purpose	AUTOSAR API service ID.
Value	21U
Description	Definition of service ID for <a href="#">EcuM_GetValidatedWakeupEvents()</a> .

#### 5.4.3.2.47. ECUM\_SID\_GET\_VERSION\_INFO

Purpose	AUTOSAR API service ID.
Value	0U
Description	Definition of service ID for <a href="#">EcuM_GetVersionInfo()</a> .

#### 5.4.3.2.48. ECUM\_SID\_GO\_RUN

Purpose	AUTOSAR API service ID.
Value	99U
Description	Definition of service ID for <a href="#">EcuM_BswMInitFinished()</a> .

#### 5.4.3.2.49. ECUM\_SID\_HALT

Purpose	AUTOSAR API service ID.
Value	32U
Description	Definition of service ID for <a href="#">EcuM_GoHalt()</a>



#### 5.4.3.2.50. ECUM\_SID\_INIT

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	1U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_Init()</a>

#### 5.4.3.2.51. ECUM\_SID\_KILL\_ALL\_RUN\_REQUESTS

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	5U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_KillAllRUNRequests()</a>

#### 5.4.3.2.52. ECUM\_SID\_MAIN\_FUNCTION

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	24U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_MainFunction()</a> .

#### 5.4.3.2.53. ECUM\_SID\_POLL

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	33U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_GoPoll()</a>

#### 5.4.3.2.54. ECUM\_SID\_RELEASE\_POST\_RUN

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	11U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_ReleasePOST_RUN()</a>

#### 5.4.3.2.55. ECUM\_SID\_RELEASE\_RUN

<b>Purpose</b>	AUTOSAR API service ID.
----------------	-------------------------



<b>Value</b>	4U
<b>Description</b>	Definition of service ID for EcuM_ReleaseRUN()

#### 5.4.3.2.56. ECUM\_SID\_REQUEST\_POST\_RUN

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	10U
<b>Description</b>	Definition of service ID for EcuM_RequestPOST_RUN()

#### 5.4.3.2.57. ECUM\_SID\_REQUEST\_RUN

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	3U
<b>Description</b>	Definition of service ID for EcuM_RequestRUN()

#### 5.4.3.2.58. ECUM\_SID\_SEL\_BOOT\_TAR

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	18U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_SelectBootTarget()</a> .

#### 5.4.3.2.59. ECUM\_SID\_SEL\_SHUT\_CAUSE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	27U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_SelectShutdownCause()</a> .

#### 5.4.3.2.60. ECUM\_SID\_SEL\_SHUT\_TAR

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	6U



<b>Description</b>	Definition of service ID for EcuM_SelectShutdownTarget().
--------------------	---

#### 5.4.3.2.61. ECUM\_SID\_SET\_WK\_EV

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	12U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_SetWakeupEvent()</a> .

#### 5.4.3.2.62. ECUM\_SID\_SHUTDOWN

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	2U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_Shutdown()</a>

#### 5.4.3.2.63. ECUM\_SID\_STARTUP\_TWO

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	26U
<b>Description</b>	Definition of service ID for <a href="#">EcuM_StartupTwo()</a> .

#### 5.4.3.2.64. ECUM\_SID\_VALIDATE\_WK\_EV

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	20U
<b>Description</b>	Definition of service ID for EcuM_ValidateWakeupEvent.

#### 5.4.3.2.65. ECUM\_STATE\_APP\_POST\_RUN

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x33U
<b>Description</b>	Definition of ECUM_STATE_APP_POST_RUN, substate of EcuM RUN



#### 5.4.3.2.66. ECUM\_STATE\_APP\_RUN

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x32U
<b>Description</b>	Definition of ECUM_STATE_APP_RUN, substate of EcuM RUN.

#### 5.4.3.2.67. ECUM\_STATE\_GO\_OFF\_ONE

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x4dU
<b>Description</b>	Definition of ECUM_STATE_GO_OFF_ONE, substate of EcuM SHUTDOWN

#### 5.4.3.2.68. ECUM\_STATE\_GO\_OFF\_TWO

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x4eU
<b>Description</b>	Definition of ECUM_STATE_GO_OFF_TWO, substate of EcuM SHUTDOWN

#### 5.4.3.2.69. ECUM\_STATE\_GO\_SLEEP

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x49U
<b>Description</b>	Definition of ECUM_STATE_GO_SLEEP, substate of EcuM SHUTDOWN

#### 5.4.3.2.70. ECUM\_STATE\_OFF

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x80U
<b>Description</b>	Definition of ECUM_STATE_OFF, indicates EcuM OFF state.

#### 5.4.3.2.71. ECUM\_STATE\_PREP\_SHUTDOWN

<b>Purpose</b>	Value for EcuM_StateType.
----------------	---------------------------



<b>Value</b>	0x44U
<b>Description</b>	Definition of ECUM_STATE_PREP_SHUTDOWN, substate of EcuM SHUTDOWN

#### 5.4.3.2.72. ECUM\_STATE\_RESET

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x90U
<b>Description</b>	Definition of ECUM_STATE_RESET, indicates EcuM RESET state.

#### 5.4.3.2.73. ECUM\_STATE\_RUN

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x30U
<b>Description</b>	Definition of ECUM_STATE_RUN, indicates EcuM RUN state.

#### 5.4.3.2.74. ECUM\_STATE\_SHUTDOWN

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x40U
<b>Description</b>	Definition of ECUM_STATE_SHUTDOWN, indicates EcuM SHUTDOWN state.

#### 5.4.3.2.75. ECUM\_STATE\_SLEEP

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x50U
<b>Description</b>	Definition of ECUM_STATE_SLEEP, indicates EcuM SLEEP state.

#### 5.4.3.2.76. ECUM\_STATE\_STARTUP

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x10U



<b>Description</b>	Definition of ECUM_STATE_STARTUP, indicates EcuM STARTUP state.
--------------------	---

#### 5.4.3.2.77. ECUM\_STATE\_STARTUP\_ONE

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x11U
<b>Description</b>	Definition of ECUM_STATE_STARTUP_ONE, substate of EcuM STARTUP.

#### 5.4.3.2.78. ECUM\_STATE\_STARTUP\_TWO

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x12U
<b>Description</b>	Definition of ECUM_STATE_STARTUP_TWO, substate of EcuM STARTUP.

#### 5.4.3.2.79. ECUM\_STATE\_WAKEUP

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x20U
<b>Description</b>	Definition of ECUM_STATE_WAKEUP, indicates EcuM WAKEUP state.

#### 5.4.3.2.80. ECUM\_STATE\_WAKEUP\_ONE

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x21U
<b>Description</b>	Definition of ECUM_STATE_WAKEUP_ONE, substate of EcuM WAKEUP.

#### 5.4.3.2.81. ECUM\_STATE\_WAKEUPREACTION

<b>Purpose</b>	Value for EcuM_StateType.
<b>Value</b>	0x23U
<b>Description</b>	Definition of ECUM_STATE_WAKEUPREACTION, substate of EcuM WAKEUP.



#### 5.4.3.2.82. ECUM\_STATE\_WAKEUP\_TTII

<b>Purpose</b>	Value for Ecum_StateType.
<b>Value</b>	0x26U
<b>Description</b>	Definition of ECUM_STATE_WAKEUP_TTII, substate of Ecum WAKEUP.

#### 5.4.3.2.83. ECUM\_STATE\_WAKEUP\_TWO

<b>Purpose</b>	Value for Ecum_StateType.
<b>Value</b>	0x24U
<b>Description</b>	Definition of ECUM_STATE_WAKEUP_TWO, substate of Ecum WAKEUP.

#### 5.4.3.2.84. ECUM\_STATE\_WAKEUP\_VALIDATION

<b>Purpose</b>	Value for Ecum_StateType.
<b>Value</b>	0x22U
<b>Description</b>	Definition of ECUM_STATE_WAKEUP_VALIDATION, substate of Ecum WAKEUP.

#### 5.4.3.2.85. ECUM\_STATE\_WAKEUP\_WAKESLEEP

<b>Purpose</b>	Value for Ecum_StateType.
<b>Value</b>	0x25U
<b>Description</b>	Definition of ECUM_STATE_WAKEUP_WAKESLEEP, substate of Ecum WAKEUP.

#### 5.4.3.2.86. ECUM\_SUBSTATE\_MASK

<b>Purpose</b>	Bitmask for sub states.
<b>Value</b>	0x0FU

#### 5.4.3.2.87. ECUM\_SUPERSTATE\_MASK

<b>Purpose</b>	Bitmask for super states.
----------------	---------------------------



<b>Value</b>	0xF0U
--------------	-------

#### 5.4.3.2.88. ECUM\_SW\_MAJOR\_VERSION

<b>Purpose</b>	AUTOSAR module major version.
<b>Value</b>	5U

#### 5.4.3.2.89. ECUM\_SW\_MINOR\_VERSION

<b>Purpose</b>	AUTOSAR module minor version.
<b>Value</b>	15U

#### 5.4.3.2.90. ECUM\_SW\_PATCH\_VERSION

<b>Purpose</b>	AUTOSAR module patch version.
<b>Value</b>	8U

#### 5.4.3.2.91. ECUM\_VAR\_BOOTTARGET

<b>Purpose</b>	
<b>Value</b>	ECUM_VAR_CLEARED

#### 5.4.3.2.92. ECUM\_VENDOR\_ID

<b>Purpose</b>	AUTOSAR vendor identification: Elektrobit Automotive GmbH.
<b>Value</b>	1U

#### 5.4.3.2.93. ECUM\_WKACT\_RUN

<b>Purpose</b>	
<b>Value</b>	0x00U



#### 5.4.3.2.94. ECUM\_WKACT\_SHUTDOWN

Purpose	
Value	0x03U

#### 5.4.3.2.95. ECUM\_WKACT\_TTII

Purpose	
Value	0x02U

#### 5.4.3.2.96. ECUM\_WKSOURCE\_ALL\_SOURCES

Purpose	Value for EcuM_WakeupSourceType.
Value	0xFFFFFFFFFU
Description	Set all wakeup sources.

#### 5.4.3.2.97. ECUM\_WKSTATUS\_DISABLED

Purpose	Value for EcuM_WakeupStatusType.
Value	4U
Description	The wakeup source is disabled and does not detect wakeup events.

#### 5.4.3.2.98. ECUM\_WKSTATUS\_ENABLED

Purpose	Value for EcuM_WakeupStatusType.
Value	6U
Description	The wakeup source is enabled and does not detect wakeup events.

#### 5.4.3.2.99. ECUM\_WKSTATUS\_EXPIRED

Purpose	Value for EcuM_WakeupStatusType.
---------	----------------------------------



<b>Value</b>	3U
<b>Description</b>	The wakeup event has not been validated and has therefore expired.

#### 5.4.3.2.100. ECUM\_WKSTATUS\_NONE

<b>Purpose</b>	Value for EcuM_WakeupStatusType.
<b>Value</b>	0U
<b>Description</b>	No pending wakeup event was detected.

#### 5.4.3.2.101. ECUM\_WKSTATUS\_PENDING

<b>Purpose</b>	Value for EcuM_WakeupStatusType.
<b>Value</b>	1U
<b>Description</b>	The wakeup event was detected but not yet validated.

#### 5.4.3.2.102. ECUM\_WKSTATUS\_VALIDATED

<b>Purpose</b>	Value for EcuM_WakeupStatusType.
<b>Value</b>	2U
<b>Description</b>	The wakeup event is valid.

#### 5.4.3.2.103. EcuM\_GetBootTarget

<b>Purpose</b>	Wrapping macro for EcuM_GetBootTarget.
<b>Value</b>	EcuM_ASR40_GetBootTarget
<b>Description</b>	Provides an AUTOSAR 4.0 API as default to other BSW modules.

#### 5.4.3.2.104. EcuM\_GetLastShutdownTarget

<b>Purpose</b>	Wrapping macro for EcuM_GetLastShutdownTarget.
<b>Value</b>	EcuM_ASR40_GetLastShutdownTarget



<b>Description</b>	Provides an AUTOSAR 4.0 API as default to other BSW modules.
--------------------	--

#### 5.4.3.2.105. EcuM\_GetMostRecentShutdown

<b>Purpose</b>	Wrapping macro for EcuM_GetMostRecentShutdown.
<b>Value</b>	EcuM_AS40_GetMostRecentShutdown
<b>Description</b>	Provides an AUTOSAR 4.0 API as default to other BSW modules.

#### 5.4.3.2.106. EcuM\_GetNextRecentShutdown

<b>Purpose</b>	Wrapping macro for EcuM_GetNextRecentShutdown.
<b>Value</b>	EcuM_AS40_GetNextRecentShutdown
<b>Description</b>	Provides an AUTOSAR 4.0 API as default to other BSW modules.

#### 5.4.3.2.107. EcuM\_GetShutdownCause

<b>Purpose</b>	Wrapping macro for EcuM_GetShutdownCause.
<b>Value</b>	EcuM_AS40_GetShutdownCause
<b>Description</b>	Provides an AUTOSAR 4.0 API as default to other BSW modules.

#### 5.4.3.2.108. EcuM\_GetShutdownTarget

<b>Purpose</b>	Wrapping macro for EcuM_GetShutdownTarget.
<b>Value</b>	EcuM_AS40_GetShutdownTarget
<b>Description</b>	Provides an AUTOSAR 4.0 API as default to other BSW modules.

#### 5.4.3.2.109. EcuM\_SelectBootTarget

<b>Purpose</b>	Wrapping macro for EcuM_SelectBootTarget.
<b>Value</b>	EcuM_AS40_SelectBootTarget
<b>Description</b>	Provides an AUTOSAR 4.0 API as default to other BSW modules.



#### 5.4.3.2.110. EcuM\_SelectShutdownCause

<b>Purpose</b>	Wrapping macro for EcuM_SelectShutdownCause.
<b>Value</b>	EcuM_ASR40_SelectShutdownCause
<b>Description</b>	Provides an AUTOSAR 4.0 API as default to other BSW modules.

#### 5.4.3.3. Objects

##### 5.4.3.3.1. EcuM\_BootTarget

<b>Purpose</b>	
Type	<a href="#">EcuM_BootTargetInternalType</a>

#### 5.4.3.4. Functions

##### 5.4.3.4.1. EcuM\_AL\_DriverInitOne

<b>Purpose</b>	Callout in startup sequence one.
<b>Synopsis</b>	<code>void EcuM_AL_DriverInitOne ( const EcuM_ConfigType * ConfigPtr ) ;</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	<p>This callout provides driver initialization and other hardware-related startup activities in case of a power on reset.</p> <p>This callout is invoked by EcuM before Gpt and OS is started.</p>

##### 5.4.3.4.2. EcuM\_AL\_DriverInitZero

<b>Purpose</b>	Callout in startup sequence zero.
<b>Synopsis</b>	<code>void EcuM_AL_DriverInitZero ( void ) ;</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant



<b>Description</b>	This callout provides driver initialization and other hardware-related startup activities in case of a power on reset.  This callout is invoked by EcuM before Mcu is started.
--------------------	--

#### 5.4.3.4.3. EcuM\_AL\_DriverRestart

<b>Purpose</b>	Callout for restarting drivers.
<b>Synopsis</b>	<code>void EcuM_AL_DriverRestart ( const EcuM_ConfigType * ConfigPtr ) ;</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	This callout provides driver initialization and other hardware-related startup activities in the wakeup case.  This callout is invoked by EcuM early in WAKEUP 1 state.

#### 5.4.3.4.4. EcuM\_AL\_Reset

<b>Purpose</b>	Callout for resetting the ECU.
<b>Synopsis</b>	<code>void EcuM_AL_Reset ( EcuM_ResetType reset ) ;</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	This callout shall take the code for resetting the ECU.

#### 5.4.3.4.5. EcuM\_AL\_SetProgrammableInterrupts

<b>Purpose</b>	Callout from the startup phase.
<b>Synopsis</b>	<code>void EcuM_AL_SetProgrammableInterrupts ( void ) ;</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	This callout shall set the interrupts on ECUs with programmable interrupts.  This callout is invoked by EcuM before Mcu is started.



#### 5.4.3.4.6. EcuM\_AL\_SwitchOff

<b>Purpose</b>	Callout for switching off power.
<b>Synopsis</b>	<code>void EcuM_AL_SwitchOff ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	This callout shuts off the power supply of the ECU. If the ECU cannot unpower itself, a reset may be an adequate reaction.

#### 5.4.3.4.7. EcuM\_ASR32\_GetBootTarget

<b>Purpose</b>	Get boot target using AUTOSAR 4.0 API.	
<b>Synopsis</b>	<code>Std_ReturnType EcuM_ASR32_GetBootTarget ( EcuM_ASR32_BootTargetType * target );</code>	
<b>Service ID</b>	19	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (out)</b>	target	The currently selected boot target.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors (Always).
<b>Description</b>	This function gets the currently selected boot target.	

#### 5.4.3.4.8. EcuM\_ASR32\_GetLastShutdownTarget

<b>Purpose</b>	Get target state for shutdown sequence using AUTOSAR 3.2 API.	
<b>Synopsis</b>	<code>Std_ReturnType EcuM_ASR32_GetLastShutdownTarget ( EcuM_ASR32_ShutdownType * shutdownTarget , uint8 * sleepMode );</code>	
<b>Service ID</b>	<a href="#"><u>ECUM_SID_GET_LAST_SHUT_TAR</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (out)</b>	shutdownTarget	Set to one of the following values:



		<ul style="list-style-type: none"> <li>▶ ECUM_STATE_SLEEP</li> <li>▶ ECUM_STATE_RESET</li> <li>▶ ECUM_STATE_OFF</li> </ul>
	sleepMode	If the return parameter is ECUM_STATE_SLEEP, this output parameter tells which of the configured sleep modes was actually chosen (index into Ecum_SleepMode). If the return parameter is ECUM_STATE_RESET, this out parameter tells which of the configured reset modes was actually chosen.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	The new shutdown target was not set.
<b>Description</b>	<p>This function always returns the same value until the next shutdown. The return value describes the ECU state from which the last wakeup or power up occurred.</p> <p>Development Error Detection</p> <ul style="list-style-type: none"> <li>▶ ECUM_E_NULL_POINTER: If a null pointer was passed as argument.</li> </ul>	

#### 5.4.3.4.9. `EcuM_ASR32_GetShutdownTarget`

<b>Purpose</b>	Get target state for shutdown sequence using AUTOSAR 3.2 API.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>EcuM_ASR32_GetShutdownTarget</b> ( EcuM_ASR32_StateType * shutdownTarget , uint8 * sleepMode );</pre>	
<b>Service ID</b>	0x09	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (out)</b>	shutdownTarget	Is set to one of the following values: <ul style="list-style-type: none"> <li>▶ ECUM_STATE_SLEEP</li> <li>▶ ECUM_STATE_RESET</li> <li>▶ ECUM_STATE_OFF</li> </ul>
	sleepMode	If the return parameter is ECUM_STATE_SLEEP, then this output parameter identifies which of the configured sleep modes



		was actually chosen (index into EcuM_-SleepMode). If the return parameter is ECUM_STATE_RESET, this out parameter tells which of the configured reset modes was actually chosen.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors. (Always).
	E_NOT_OK	The service has failed (e.g. due to a Null pointer being passed).
<b>Description</b>	This function always returns the selected shutdown target as set by EcuM_SelectShutdownTarget.  Development Error Detection  ► ECUM_E_NULL_POINTER: If a null pointer was passed as argument.	

#### 5.4.3.4.10. EcuM\_ASR32\_ReleasePOST\_RUN

<b>Purpose</b>	Releases a POST RUN request previously done with a call to EcuM_RequestPOST_RUN using AUTOSAR 3.2 API.	
<b>Synopsis</b>	Std_ReturnType <b>EcuM_ASR32_ReleasePOST_RUN</b> ( EcuM_ASR32_UserType user );	
<b>Service ID</b>	<a href="#"><u>ECUM_SID_RELEASE_POST_RUN</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	user	ID of the entity releasing the POST RUN state.
<b>Return Value</b>	Standard Return Code	
	E_OK	The release request was accepted by EcuM.
	E_NOT_OK	The release request was not accepted by EcuM, a detailed error condition was sent to DET.
<b>Description</b>	After the last POST_RUN request has been released, EcuM indicates the SHUTDOWN mode to Rte. This mode transition can then be caught by BswM to proceed with the shutdown or sleep tasks.  The service is intended for implementing AUTOSAR ports.	



#### 5.4.3.4.11. EcuM\_AS32\_ReleaseRUN

<b>Purpose</b>	Releases a RUN request previously done with a call to EcuM_RequestRUN using AUTOSAR 3.2 API.	
<b>Synopsis</b>	Std_ReturnType <b>EcuM_AS32_ReleaseRUN</b> ( EcuM_AS32_UserType user );	
<b>Service ID</b>	<a href="#"><u>ECUM_SID_RELEASE_RUN</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	user	ID of the entity releasing the RUN state.
<b>Return Value</b>	Standard Return Code	
	E_OK	The release request was accepted by EcuM.
	E_NOT_OK	The release request was not accepted by EcuM, a detailed error condition was sent to DET.
<b>Description</b>	<p>When the last RUN request is released, EcuM leaves the RUN state.</p> <p>The service is intended for implementing AUTOSAR ports.</p>	

#### 5.4.3.4.12. EcuM\_AS32\_RequestPOST\_RUN

<b>Purpose</b>	Places a request for the POST RUN state using AUTOSAR 3.2 API.	
<b>Synopsis</b>	Std_ReturnType <b>EcuM_AS32_RequestPOST_RUN</b> ( EcuM_AS32_UserType user );	
<b>Service ID</b>	<a href="#"><u>ECUM_SID_REQUEST_POST_RUN</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	user	ID of the entity requesting the POST RUN state.
<b>Return Value</b>	Standard Return Code	
	E_OK	The request was accepted by EcuM.
	E_NOT_OK	The request was not accepted by EcuM, a detailed error condition was sent to DET.



<b>Description</b>	<p>EcuM enters POST_RUN state after all RUN requests have been released and stays there for as long as POST_RUN requests are pending.</p> <p>Requests can be placed by every user made known to the state manager at configuration time. Requests for RUN and POST RUN must be tracked independently (in other words: two independent variables). The service is intended for implementing AUTOSAR ports</p>
--------------------	--

#### **5.4.3.4.13. EcuM\_ASR32\_RequestRUN**

<b>Purpose</b>	Places a request for the RUN state using AUTOSAR 3.2 API.	
<b>Synopsis</b>	<pre>Std_ReturnType EcuM_ASR32_RequestRUN ( EcuM_ASR32_UserType user ) ;</pre>	
<b>Service ID</b>	<a href="#"><u>ECUM_SID_REQUEST_RUN</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	user	ID of the entity requesting the RUN state.
<b>Return Value</b>	Standard Return Code	
	E_OK	The request was accepted by EcuM.
	E_NOT_OK	The request was not accepted by EcuM, a detailed error condition was sent to DET.
<b>Description</b>	<p>EcuM enters RUN mode after the <a href="#"><u>EcuM_GoRun()</u></a> callback has been called and stays there for as long as RUN requests requested by this API function are pending.</p> <p>Requests can be placed by every user made known to the state manager at configuration time.</p>	

#### **5.4.3.4.14. EcuM\_ASR32\_SelectBootTarget**

<b>Purpose</b>	Select boot target using AUTOSAR 4.0 API.
<b>Synopsis</b>	Std_ReturnType <b>EcuM_ASR32_SelectBootTarget</b> ( EcuM_ASR32_Boot- TargetType target );
<b>Service ID</b>	18
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Reentrant



<b>Parameters (in)</b>	target	The selected boot target.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	The new boot target was not accepted by EcuM.
<b>Description</b>	This function sets the target which is used after reboot.	

#### 5.4.3.4.15. EcuM\_AS32\_SelectShutdownTarget

<b>Purpose</b>	Select target state for shutdown sequence using AUTOSAR 3.2 API.	
<b>Synopsis</b>	Std_ReturnType <b>EcuM_AS32_SelectShutdownTarget</b> ( EcuM_AS32_-StateType target , uint8 mode );	
<b>Service ID</b>	0x06	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	target	The selected shutdown target. Only the following subset of the EcuM_StateType value range is accepted: <ul style="list-style-type: none"><li>▶ ECUM_STATE_SLEEP</li><li>▶ ECUM_STATE_RESET</li><li>▶ ECUM_STATE_OFF</li></ul>
	mode	An index like value which can be dereferenced to a sleep mode ( <a href="#">EcuM_Sleep-ModeConfigType</a> ). Available sleep modes are defined at configuration time and are stored in the EcuM_SleepMode list. The mode parameter is only used if the target parameter equals to ECUM_STATE_SLEEP. In all other cases, it is ignored.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	The new shutdown target was not set.
<b>Description</b>	This function doesn't initiate any shutdown activities but only stores the target state for later use when performing a shutdown sequence. During SHUTDOWN a change of the shutdown target from and to the value ECUM_STATE_SLEEP is not allowed.	



Development Error Detection

- ▶ ECUM\_E\_STATE\_PAR\_OUT\_OF\_RANGE: On invalid value for parameter target.

#### 5.4.3.4.16. Ecum\_ASR40\_GetBootTarget

<b>Purpose</b>	Get boot target using AUTOSAR 4.0 API.	
<b>Synopsis</b>	Std_ReturnType <b>Ecum_ASR40_GetBootTarget</b> ( Ecum_ASR40_BootTargetType * target );	
<b>Service ID</b>	19	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (out)</b>	target	The currently selected boot target.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors (Always).
<b>Description</b>	This function gets the currently selected boot target.	

#### 5.4.3.4.17. Ecum\_ASR40\_GetLastShutdownTarget

<b>Purpose</b>	Get target state for shutdown sequence using AUTOSAR 4.0 API.	
<b>Synopsis</b>	Std_ReturnType <b>Ecum_ASR40_GetLastShutdownTarget</b> ( Ecum_ASR40_StateType * shutdownTarget , uint8 * sleepMode );	
<b>Service ID</b>	<a href="#">ECUM_SID_GET_LAST_SHUT_TAR</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (out)</b>	shutdownTarget	Set to one of the following values: <ul style="list-style-type: none"> <li>▶ ECUM_STATE_SLEEP</li> <li>▶ ECUM_STATE_RESET</li> <li>▶ ECUM_STATE_OFF</li> </ul>
	sleepMode	If the return parameter is ECUM_STATE_SLEEP, this output parameter tells which of the configured sleep modes was actually chosen (index into Ecum_SleepMode). If the return parameter is ECUM_STATE_-



		RESET, this out parameter tells which of the configured reset modes was actually chosen.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Errors.
	E_NOT_OK	The new shutdown target was not set.
<b>Description</b>	<p>This function always returns the same value until the next shutdown. The return value describes the ECU state from which the last wakeup or power up occurred.</p> <p>Development Error Detection</p> <ul style="list-style-type: none"> <li>▶ ECUM_E_NULL_POINTER: If a null pointer was passed as argument.</li> </ul>	

#### 5.4.3.4.18. EcuM\_AS40\_GetMostRecentShutdown

<b>Purpose</b>	Get most recent shutdown operation using AUTOSAR 4.0 API.	
<b>Synopsis</b>	<pre>Std_ReturnType EcuM_AS40_GetMostRecentShutdown ( EcuM_AS40_- StateType * target , uint8 * mode , EcuM_AS40_ShutdownCause- Type * cause , uint32 * time );</pre>	
<b>Service ID</b>	<a href="#">ECUM_SID_GET_RCNT_SHUT_CAUSE</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (out)</b>	<p>target</p> <p>mode</p> <p>cause</p> <p>time</p>	<p>One of these values is returned:</p> <ul style="list-style-type: none"> <li>▶ ECUM_STATE_SLEEP</li> <li>▶ ECUM_STATE_RESET</li> <li>▶ ECUM_STATE_OFF</li> </ul> <p>This parameter tells which of the configured sleep modes(target is ECUM_STATE_SLEEP) or which of the reset mechanisms(target is ECUM_STATE_RESET) was actually chosen.</p> <p>The selected shutdown cause</p> <p>Absolute time of the shutdown if supported by hardware.</p>
<b>Return Value</b>	<p>Standard Return Code</p> <p>E_OK</p>	The service has succeeded.



	E_NOT_OK	The service has failed, e.g. due to NULL pointer being passed
<b>Description</b>	<p>This function returns information about the most recent shutdown operation.</p> <p>Development Error Detection</p> <ul style="list-style-type: none"> <li>▶ ECUM_E_NULL_POINTER: If a null pointer was passed as argument.</li> </ul>	

#### 5.4.3.4.19. EcuM\_AS40\_GetNextRecentShutdown

<b>Purpose</b>	Get next most recent shutdown operation using AUTOSAR 4.0 API.					
<b>Synopsis</b>	<pre>Std_ReturnType EcuM_AS40_GetNextRecentShutdown ( EcuM_AS40_- StateType * target , uint8 * mode , EcuM_AS40_ShutdownCause- Type * cause , uint32 * time );</pre>					
<b>Service ID</b>	<a href="#">ECUM_SID_GET_NXT_RCNT_SHUT_CAUSE</a>					
<b>Sync/Async</b>	Synchronous					
<b>Reentrancy</b>	Reentrant					
<b>Parameters (out)</b>	<p>target</p> <p>mode</p> <p>cause</p> <p>time</p>	<p>One of these values is returned:</p> <ul style="list-style-type: none"> <li>▶ ECUM_STATE_SLEEP</li> <li>▶ ECUM_STATE_RESET</li> <li>▶ ECUM_STATE_OFF</li> </ul> <p>This parameter tells which of the configured sleep modes(target is ECUM_STATE_SLEEP) or which of the reset mechanisms(target is ECUM_STATE_RESET) was actually chosen.</p> <p>The selected shutdown cause</p> <p>Absolute time of the shutdown if supported by hardware.</p>				
<b>Return Value</b>	<p>Standard Return Code</p> <table border="1"> <tr> <td>E_OK</td> <td>The service has succeeded.</td> </tr> <tr> <td>E_NOT_OK</td> <td>The service has failed, e.g. due to NULL pointer being passed</td> </tr> </table>		E_OK	The service has succeeded.	E_NOT_OK	The service has failed, e.g. due to NULL pointer being passed
E_OK	The service has succeeded.					
E_NOT_OK	The service has failed, e.g. due to NULL pointer being passed					
<b>Description</b>	This function returns information about the next most recent shutdown operation. All stored shutdown information can be read by first calling EcuM_GetMostRecentShutdown and then looping over					



	Development Error Detection
	<ul style="list-style-type: none"><li>▶ ECUM_E_NULL_POINTER: If a null pointer was passed as argument.</li></ul>

#### **5.4.3.4.20. EcuM\_ASR40\_GetShutdownCause**

<b>Purpose</b>	Get the cause of shutdown using AUTOSAR 4.0 API.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>EcuM_AS40_GetShutdownCause</b> ( EcuM_AS40_Shut- downCauseType * shutdownCause );</pre>	
<b>Service ID</b>	<a href="#"><u>ECUM_SID_GET_SHUT_COUSE</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (out)</b>	shutdownCause	The selected cause of the next shutdown.
<b>Return Value</b>	Standard Return Code	
	E_OK	The service has succeeded.
	E_NOT_OK	The service has failed, e.g. due to NULL pointer being passed
<b>Description</b>	<p>This function returns the selected shutdown cause as set by EcuM_SelectShutdownCause.</p> <p>Development Error Detection</p> <ul style="list-style-type: none"> <li>▶ ECUM_E_NULL_POINTER: If a null pointer was passed as argument.</li> </ul>	

#### **5.4.3.4.21. EcuM\_ASR40\_GetShutdownTarget**

<b>Purpose</b>	Get target state for shutdown sequence using AUTOSAR 4.0 API.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>EcuM_ASR40_GetShutdownTarget</b> ( EcuM_ASR40_State- Type * shutdownTarget , uint8 * sleepMode );</pre>	
<b>Service ID</b>	<a href="#"><u>ECUM_SID_GET_SHUT_TAR</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (out)</b>	shutdownTarget	<p>Is set to one of the following values:</p> <ul style="list-style-type: none"> <li>▶ ECUM_STATE_SLEEP</li> <li>▶ ECUM_STATE_RESET</li> </ul>



		► ECUM_STATE_OFF
	sleepMode	If the return parameter is ECUM_STATE_SLEEP, then this output parameter identifies which of the configured sleep modes was actually chosen (index into EcuM_SleepMode). If the return parameter is ECUM_STATE_RESET, this output parameter tells which of the configured reset modes was actually chosen.
<b>Return Value</b>		Standard Return Code
	E_OK	No Errors. (Always).
	E_NOT_OK	The service has failed (e.g. due to a Null pointer being passed).
<b>Description</b>		<p>This function always returns the selected shutdown target as set by EcuM_SelectShutdownTarget.</p> <p>Development Error Detection</p> <p>► ECUM_E_NULL_POINTER: If a null pointer was passed as argument.</p>

#### 5.4.3.4.22. EcuM\_AS40\_ReleasePOST\_RUN

<b>Purpose</b>	Releases a POST RUN request previously done with a call to EcuM_RequestPOST_RUN using AUTOSAR 4.0 API.	
<b>Synopsis</b>	Std_ReturnType <b>EcuM_AS40_ReleasePOST_RUN</b> ( EcuM_AS40_UserType user );	
<b>Service ID</b>	<a href="#"><u>ECUM_SID_RELEASE_POST_RUN</u></a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	user	ID of the entity releasing the POST RUN state.
<b>Return Value</b>		Standard Return Code
	E_OK	The release request was accepted by EcuM.
	E_NOT_OK	The release request was not accepted by EcuM, a detailed error condition was sent to DET.



<b>Description</b>	After the last POST_RUN request has been released, EcuM indicates the SHUTDOWN mode to Rte. This mode transition can then be caught by BswM to proceed with the shutdown or sleep tasks.  The service is intended for implementing AUTOSAR ports.
--------------------	---

#### 5.4.3.4.23. EcuM\_AS40\_ReleaseRUN

<b>Purpose</b>	Releases a RUN request previously done with a call to EcuM_RequestRUN using AUTOSAR 4.0 API.	
<b>Synopsis</b>	Std_ReturnType <b>EcuM_AS40_ReleaseRUN</b> ( EcuM_AS40_UserType user );	
<b>Service ID</b>	<a href="#">ECUM_SID_RELEASE_RUN</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	user	ID of the entity releasing the RUN state.
<b>Return Value</b>	Standard Return Code	
	E_OK	The release request was accepted by EcuM.
	E_NOT_OK	The release request was not accepted by EcuM, a detailed error condition was sent to DET.
<b>Description</b>	When the last RUN request is released, EcuM leaves the RUN state.  The service is intended for implementing AUTOSAR ports.	

#### 5.4.3.4.24. EcuM\_AS40\_RequestPOST\_RUN

<b>Purpose</b>	Places a request for the POST RUN state using AUTOSAR 4.0 API.
<b>Synopsis</b>	Std_ReturnType <b>EcuM_AS40_RequestPOST_RUN</b> ( EcuM_AS40_UserType user );
<b>Service ID</b>	<a href="#">ECUM_SID_REQUEST_POST_RUN</a>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Reentrant



<b>Parameters (in)</b>	user	ID of the entity requesting the POST RUN state.
<b>Return Value</b>	Standard Return Code	
	E_OK	The request was accepted by Ecum.
	E_NOT_OK	The request was not accepted by Ecum, a detailed error condition was sent to DET.
<b>Description</b>	<p>Ecum enters POST_RUN state after all RUN requests have been released and stays there for as long as POST_RUN requests are pending.</p> <p>Requests can be placed by every user made known to the state manager at configuration time. Requests for RUN and POST RUN must be tracked independently (in other words: two independent variables). The service is intended for implementing AUTOSAR ports</p>	

#### 5.4.3.4.25. Ecum\_ASR40\_RequestRUN

<b>Purpose</b>	Places a request for the RUN state using AUTOSAR 4.0 API.	
<b>Synopsis</b>	Std_ReturnType <b>Ecum_ASR40_RequestRUN</b> ( Ecum_ASR40_UserType user );	
<b>Service ID</b>	<a href="#">ECUM_SID_REQUEST_RUN</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	user	ID of the entity requesting the RUN state.
<b>Return Value</b>	Standard Return Code	
	E_OK	The request was accepted by Ecum.
	E_NOT_OK	The request was not accepted by Ecum, a detailed error condition was sent to DET.
<b>Description</b>	<p>Ecum enters RUN mode after the <a href="#">Ecum_GoRun()</a> callback has been called and stays there for as long as RUN requests requested by this API function are pending.</p> <p>Requests can be placed by every user made known to the state manager at configuration time.</p>	

#### 5.4.3.4.26. Ecum\_ASR40\_SelectBootTarget

<b>Purpose</b>	Select boot target using AUTOSAR 4.0 API.
----------------	---



<b>Synopsis</b>	Std_ReturnType <b>EcuM_AS40_SelectBootTarget</b> ( EcuM_AS40_BootTargetType target );	
<b>Service ID</b>	18	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	target	The selected boot target.
<b>Return Value</b>	Standard Return Code E_OK                          No Errors. E_NOT_OK                      The new boot target was not accepted by EcuM.	
<b>Description</b>	This function sets the target which is used after reboot.	

#### 5.4.3.4.27. EcuM\_AS40\_SelectShutdownCause

<b>Purpose</b>	Select cause of shutdown using AUTOSAR 4.0 API.	
<b>Synopsis</b>	Std_ReturnType <b>EcuM_AS40_SelectShutdownCause</b> ( EcuM_AS40_ShutdownCauseType target );	
<b>Service ID</b>	<a href="#">ECUM_SID_SEL_SHUT_CAUSE</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	target	The selected shutdown cause
<b>Return Value</b>	Standard Return Code E_OK                          The new shutdown cause was set. E_NOT_OK                      The new shutdown cause was not set.	
<b>Description</b>	This function elects the cause for a shutdown.  Development Error Detection  ➤ ECUM_E_NULL_POINTER: If a null pointer was passed as argument.	

#### 5.4.3.4.28. EcuM\_CheckRamHash

<b>Purpose</b>	Callout for checking RAM integrity .
----------------	--------------------------------------



<b>Synopsis</b>	<code>uint8 EcuM_CheckRamHash ( void );</code>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Return Value</b>	Result of Check	
	0	RAM integrity test failed.
	<code>else</code>	RAM integrity test passed.
<b>Description</b>	<p>This callout provides a RAM integrity test to ensure that RAM contents is still consistent after a long SLEEP duration.</p> <p>This callout is invoked by EcuM early in WAKEUP 1 state.</p>	

#### 5.4.3.4.29. EcuM\_CheckValidation

<b>Purpose</b>	Callout which checks pending wakeup sources.	
<b>Synopsis</b>	<code>void EcuM_CheckValidation ( EcuM_WakeupSourceType wakeupSource );</code>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	wakeupSource	All pending wakeup sources.
<b>Description</b>	This callout is invoked in wakeup validation state. In this callout the system integrator has to check the wakeup sources for validation of a pending wakeup event.	

#### 5.4.3.4.30. EcuM\_CheckWakeups

<b>Purpose</b>	Callout which checks wakeup sources.	
<b>Synopsis</b>	<code>void EcuM_CheckWakeups ( EcuM_WakeupSourceType wakeupSource );</code>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	wakeupSource	All wakeup sources, activated in the current sleep
<b>Parameters (in,out)</b>	wakeupSource	All wakeup sources, activated in the current sleep
<b>Description</b>	In this callout the system integrator checks the wakeup sources for detected wakeup events. These wakeup events are reported using <a href="#">EcuM_SetWakeupsEvent()</a> .	



This callout is invoked during sleep state or is invoked by an ISR resulting from an wakeup event, in case the CPU is powered off during sleep.

#### 5.4.3.4.31. EcuM\_ClearWakeupEvent

<b>Purpose</b>	Clear wakeup events.	
<b>Synopsis</b>	<code>void EcuM_ClearWakeupEvent ( EcuM_WakeupSourceType sources );</code>	
<b>Service ID</b>	<a href="#">ECUM_SID_CLEAR_WK_EV</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant Non-Interruptible	
<b>Parameters (in)</b>	sources	Events of wakeup sources to be cleared.
<b>Description</b>	This function clears the wakeup events for a set of sources.	

#### 5.4.3.4.32. EcuM\_DefaultInitListOne

<b>Purpose</b>	Default implementation of the EcuMDriverInitListOne.
<b>Synopsis</b>	<code>void EcuM_DefaultInitListOne ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	This function provides the default implementation of the EcuMDriverInitListOne. When this function is called, the initialization of the configured EcuMDriverInitItems within the EcuM configuration container EcuMDriverInitListOne will be carried out.

#### 5.4.3.4.33. EcuM\_DefaultInitListZero

<b>Purpose</b>	Default implementation of the EcuMDriverInitListZero.
<b>Synopsis</b>	<code>void EcuM_DefaultInitListZero ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	This function provides the default implementation of the EcuMDriverInitListZero. When this function is called, the initialization of the configured EcuMDriverInitItems within the EcuM configuration container EcuMDriverInitListZero will be carried out.



#### 5.4.3.4.34. EcuM\_DefaultRestartList

<b>Purpose</b>	Default implementation of the EcuMDriverRestartList.
<b>Synopsis</b>	<code>void EcuM_DefaultRestartList ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	This function provides the default implementation of the EcuMDriverRestartList. When this function is called, the initialization of the configured EcuMDriverInitItems defined within the EcuM configuration container EcuMDriverRestartList will be carried out.

#### 5.4.3.4.35. EcuM\_DeterminePbConfiguration

<b>Purpose</b>	Callout which determines post-build configuration.
<b>Synopsis</b>	<code>EcuM_DeterminePbConfigurationRetType EcuM_DeterminePbConfigura-tion ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Return Value</b>	pointer to post-build configuration
<b>Description</b>	In this callout the system integrator has to return the actual post-build configuration for the EcuM.  This callout is invoked after <a href="#">EcuM_AL_DriverInitZero()</a>

#### 5.4.3.4.36. EcuM\_DisableWakeupSources

<b>Purpose</b>	Callout which disables pending wakeup sources.	
<b>Synopsis</b>	<code>void EcuM_DisableWakeupSources ( EcuM_WakeupSourceType wake-upSource );</code>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	wakeUpSource	All pending wakeup sources.
<b>Parameters (in,out)</b>	wakeUpSource	All pending wakeup sources.
<b>Description</b>	In this callout the system integrator can disable the functionality of pending wakeup source to set a wakeup.	



This callout is invoked after a wakeup event is detected.

#### 5.4.3.4.37. EcuM\_EnableWakeupSources

<b>Purpose</b>	Callout which enables wakeup sources.	
<b>Synopsis</b>	<pre>void <b>EcuM_EnableWakeupSources</b> ( EcuM_WakeupSourceType wake-upSource );</pre>	
<b>Parameters (in)</b>	wakeUpSource	All wakeup sources, activated in the current sleep mode.
<b>Parameters (in,out)</b>	wakeUpSource	All wakeup sources, activated in the current sleep mode.
<b>Description</b>	<p>In this callout the system integrator enables the wakeup functionality of a wakeup source, which shall be activated in the current sleep mode.</p> <p>This callout is invoked before EcuM enter in Sleep.</p>	

#### 5.4.3.4.38. EcuM\_ErrorHook

<b>Purpose</b>	Callout on errors.	
<b>Synopsis</b>	<pre>void <b>EcuM_ErrorHook</b> ( Std_ReturnType reason );</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	reason	Reason for calling the error hook. Currently the EcuM only reports <a href="#">ECUM_RAM_CHECK_FAILED</a> with the error hook function.
<b>Description</b>	<p>This function is used in unrecoverable error situations. The ECU State Manager will call the error hook.</p> <p>It is up to the system integrator to react accordingly (reset, halt, restart, safe state etc.).</p>	

#### 5.4.3.4.39. EcuM\_GenerateRamHash

<b>Purpose</b>	Callout for writing a RAM hash.
----------------	---------------------------------



<b>Synopsis</b>	<code>void EcuM_GenerateRamHash ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	This callout is invoked by EcuM just before putting the ECU physically to sleep.

#### 5.4.3.4.40. `EcuM_GetExpiredWakeupEvents`

<b>Purpose</b>	Get expired wakeup events.
<b>Synopsis</b>	<code>EcuM_WakeupSourceType EcuM_GetExpiredWakeupEvents ( void );</code>
<b>Service ID</b>	<a href="#"><u>ECUM_SID_GET_EXP_WK_EV</u></a>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant Non-Interruptible
<b>Return Value</b>	All events that have been set and for which validation has failed.
<b>Description</b>	This function retrieves wakeup events that have been set and for which validation has failed. Events which do not need validation are never be reported by this function.

#### 5.4.3.4.41. `EcuM_GetPendingWakeupEvents`

<b>Purpose</b>	Gets pending wakeup events.
<b>Synopsis</b>	<code>EcuM_WakeupSourceType EcuM_GetPendingWakeupEvents ( void );</code>
<b>Service ID</b>	<a href="#"><u>ECUM_SID_GET_PEND_WK_EV</u></a>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant Non-Interruptible
<b>Return Value</b>	Wakeup sources which have pending events.
<b>Description</b>	This function gets all wakeup events which have been set but not yet validated.

#### 5.4.3.4.42. `EcuM_GetStatusOfWakeupSource`

<b>Purpose</b>	Get the status of wakeup sources.
<b>Synopsis</b>	<code>EcuM_WakeupStatusType EcuM_GetStatusOfWakeupSource ( EcuM_WakeupSourceType sources );</code>



<b>Service ID</b>	<a href="#">ECUM_SID_GET_STAT_OF_WK_SRC</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	sources	Bit mask of the wakeup sources for which the status is requested.
<b>Return Value</b>	Sum status of all wakeup sources.  ECUM_WKSTATUS_VALIDATED      At least one wakeup source is validated. ECUM_WKSTATUS_PENDING      At least one wakeup source is pending, none is validated ECUM_WKSTATUS_EXPIRED      At least one wakeup source is expired, none is validated or pending ECUM_WKSTATUS_NONE      No wakeup source is validated, pending or expired	
<b>Description</b>	Returns the sum status of the requested wakeup sources	

#### 5.4.3.4.43. EcuM\_GetValidatedWakeupevents

<b>Purpose</b>	Get validated wakeup events.
<b>Synopsis</b>	EcuM_WakeupSourceType <a href="#">EcuM_GetValidatedWakeupevents</a> ( void ) ;
<b>Service ID</b>	<a href="#">ECUM_SID_GET_VALID_WK_EV</a>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant Non-Interruptible
<b>Return Value</b>	Wakeup sources which have been validated events.
<b>Description</b>	This function retrieves wakeup events validated using EcuM_ValidateWakeupevent.

#### 5.4.3.4.44. EcuM\_GetVersionInfo

<b>Purpose</b>	Get version information of the EcuM.
<b>Synopsis</b>	void <a href="#">EcuM_GetVersionInfo</a> ( Std_VersionInfoType * VersionInfoPtr ) ;
<b>Service ID</b>	<a href="#">ECUM_SID_GET_VERSION_INFO</a>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant



<b>Parameters (out)</b>	VersionInfoPtr	Pointer to where to store the version information of this module.
<b>Description</b>	<p>This service returns the version information of this module. The version information includes:</p> <ul style="list-style-type: none"> <li>➤ Module Id</li> <li>➤ Vendor Id</li> <li>➤ Vendor specific version numbers</li> </ul>	

#### 5.4.3.4.45. EcuM\_GoDown

<b>Purpose</b>	Prepare power off or reset.	
<b>Synopsis</b>	Std_ReturnType <b>EcuM_GoDown</b> ( uint16 caller );	
<b>Service ID</b>	<a href="#">ECUM_SID_DOWN</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	caller	Module ID of the calling module. Only special modules are allowed to call this function.
<b>Return Value</b>		
<b>Description</b>	This function instruct the ECU state manager module to perform a power off or a reset depending on the selected shutdown target.	

#### 5.4.3.4.46. EcuM\_GoHalt

<b>Purpose</b>	Prepare halt.	
<b>Synopsis</b>	Std_ReturnType <b>EcuM_GoHalt</b> ( void );	
<b>Service ID</b>	<a href="#">ECUM_SID_HALT</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Return Value</b>		
<b>Description</b>	Instructs the ECU State Manager module to go into a sleep mode where the microcontroller is halted, depending on the selected shutdown target.	



#### 5.4.3.4.47. EcuM\_GoPoll

<b>Purpose</b>	Prepare Polling sleep mode.	
<b>Synopsis</b>	<code>Std_ReturnType EcuM_GoPoll ( void );</code>	
<b>Service ID</b>	<a href="#">ECUM_SID_POLL</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Return Value</b>		
<b>Description</b>	Instructs the ECU State Manager module to go into a polling sleep mode depending on the selected shutdown target.	

#### 5.4.3.4.48. EcuM\_GoRun

<b>Purpose</b>	Enter RUN state.
<b>Synopsis</b>	<code>Std_ReturnType EcuM_GoRun ( void );</code>
<b>Service ID</b>	<a href="#">ECUM_SID_GO_RUN</a>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Reentrant
<b>Return Value</b>	
<b>Description</b>	When EcuMFixed support is enabled, EcuM_Mainfunction() will only perform wake-up validation, and will process EcuMFixed specific functionality only after this callback has been called. This callback is supposed to be called by BswM after it has finished the initialization procedure (after both startup or wakeup), including all deferred initialization tasks.

#### 5.4.3.4.49. EcuM\_Init

<b>Purpose</b>	Initialize EcuM Module.
<b>Synopsis</b>	<code>void EcuM_Init ( void );</code>
<b>Service ID</b>	<a href="#">ECUM_SID_INIT</a>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Description</b>	This function initializes the ECU state manager and carries out the startup procedure. The function will never return (it calls StartOS).



#### 5.4.3.4.50. EcuM\_IsValidConfig

<b>Purpose</b>	Validates the post-build configuration data structure used in the EcuM_Init function.	
<b>Synopsis</b>	Std_ReturnType <b>EcuM_IsValidConfig</b> ( const void * voidConfigPtr ) ;	
<b>Parameters (in)</b>	voidConfigPtr	The pointer to a EcuM post-build data structure.
<b>Parameters (in,out)</b>	voidConfigPtr	The pointer to a EcuM post-build data structure.
<b>Return Value</b>		
E_OK	When the pre-compile, link-time and platform hash values stored within the post-build structure correspond to the hash values of the compiled source files. Otherwise, E_NOT_OK will be returned.	

#### 5.4.3.4.51. EcuM\_KillAllRUNRequests

<b>Purpose</b>	Releases all RUN and POST_RUN requests.
<b>Synopsis</b>	void <b>EcuM_KillAllRUNRequests</b> ( void ) ;
<b>Service ID</b>	<a href="#">ECUM_SID_KILL_ALL_RUN_REQUESTS</a>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Reentrant
<b>Production Errors</b>	► <a href="#">ECUM_E_ALL_RUN_REQUESTS_KILLED</a> : thrown, if the service EcuM_KillAllRUNRequests was issued.
<b>Description</b>	No further RUN or POST_RUN requests are accepted until shutdown has completed.  The benefit of this function over an ECU reset is that the shutdown sequence is executed, which e.g. takes care of writing back NV memory contents.

#### 5.4.3.4.52. EcuM\_MainFunction

<b>Purpose</b>	EcuM main function.
<b>Synopsis</b>	void <b>EcuM_MainFunction</b> ( void ) ;



<b>Service ID</b>	<a href="#"><u>ECUM_SID_MAIN_FUNCTION</u></a>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Not-Reentrant
<b>Production Errors</b>	► <a href="#"><u>ECUM_E_RAM_CHECK_FAILED</u></a> : thrown, if The RAM check during wakeup failed.
<b>Description</b>	<p>This service implements all activities of the ECU State Manager while the OS is up and running.</p> <p>This function has to be called periodically by a task controlled by the BSW scheduler.</p>

#### 5.4.3.4.53. EcuM\_OnCoreSync

<b>Purpose</b>	Determines if a core shall continue waiting to synchronize.
<b>Synopsis</b>	<code>EcuM_SyncActionType EcuM_OnCoreSync ( void );</code>
<b>Return Value</b>	ECUM_ABORT_SYNC when the active EcuM instance shall abort synchronization with the other cores, otherwise ECUM_CONTINUE_SYNC.
<b>Description</b>	This function is repeatedly called while the EcuM on a core is blocked waiting for another core to synchronize. This callout function shall service a watchdog timer when one exists.

#### 5.4.3.4.54. EcuM\_OnEnterRun

<b>Purpose</b>	Callout on EnterRun.
<b>Synopsis</b>	<code>void EcuM_OnEnterRun ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	<p>This call allows the system designer to notify that the RUN state is about to be entered.</p> <p>This callout is invoked by EcuM on entry of RUN state.</p>

#### 5.4.3.4.55. EcuM\_OnExitPostRun

<b>Purpose</b>	Callout on ExitPostRun.
----------------	-------------------------



<b>Synopsis</b>	<code>void EcuM_OnExitPostRun ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	<p>This call allows the system designer to notify that the POST_RUN state is about to be left.</p> <p>This callout is invoked by EcuM before leaving POST_RUN state.</p>

#### 5.4.3.4.56. EcuM\_OnExitRun

<b>Purpose</b>	Callout on ExitRun.
<b>Synopsis</b>	<code>void EcuM_OnExitRun ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	<p>This call allows the system designer to notify that the RUN state is about to be left.</p> <p>This callout is invoked by EcuM before leaving RUN state.</p>

#### 5.4.3.4.57. EcuM\_OnGoOffOne

<b>Purpose</b>	Callout on GO OFF 1.
<b>Synopsis</b>	<code>void EcuM_OnGoOffOne ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	<p>This call allows the system designer to notify that the GO OFF 1 state is about to be entered.</p> <p>This callout is invoked by EcuM on entry of GO OFF 1 state.</p>

#### 5.4.3.4.58. EcuM\_OnGoOffTwo

<b>Purpose</b>	Callout on GO OFF 2.
<b>Synopsis</b>	<code>void EcuM_OnGoOffTwo ( void );</code>



<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	<p>This call allows the system designer to notify that the GO OFF II state is about to be entered.</p> <p>This callout is invoked by EcuM on entry of GO OFF 2 state.</p>

#### 5.4.3.4.59. EcuM\_OnGoSleep

<b>Purpose</b>	Callout on GoSleep.
<b>Synopsis</b>	<code>void EcuM_OnGoSleep ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	<p>This call allows the system designer to notify that the SLEEP state is about to be entered.</p> <p>This callout is invoked by EcuM on entry of SLEEP state.</p>

#### 5.4.3.4.60. EcuM\_OnPrepShutdown

<b>Purpose</b>	Callout on PrepShutdown.
<b>Synopsis</b>	<code>void EcuM_OnPrepShutdown ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	<p>This call allows the system designer to notify that the PREP SHUTDOWN state is about to be entered.</p> <p>This callout is invoked by EcuM on entry of PREP SHUTDOWN state.</p>

#### 5.4.3.4.61. EcuM\_OnRTEStartup

<b>Purpose</b>	Callout on RTEStartup.
<b>Synopsis</b>	<code>void EcuM_OnRTEStartup ( void );</code>
<b>Sync/Async</b>	Synchronous



<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	<p>This call allows the system designer to notify that the RTE is about to be started.</p> <p>This callout is invoked by EcuM before starting the RTE.</p>

#### 5.4.3.4.62. EcuM\_PostHalt

<b>Purpose</b>	Callout during sleep state.
<b>Synopsis</b>	<code>void EcuM_PostHalt ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	<p>In this callout all the interrupts will be enabled which are disabled by the <a href="#">EcuM_PreHalt()</a>. This will be called after a wakeup event that will return the function Mcu_SetMode()</p> <p>This callout is invoked by EcuM periodically in SLEEP state, but only if the CPU is not suspended (i.e. clock is reduced).</p>

#### 5.4.3.4.63. EcuM\_PreHalt

<b>Purpose</b>	Callout during sleep state.
<b>Synopsis</b>	<code>void EcuM_PreHalt ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	<p>In this callout all the execution of interrupts leading to wakeup will be disabled. This will be called before setting up the ECU in to Halt mode. This prevents the ECU from neglecting the pending wakeup events, which occurred due to the execution of ISR before the call to Mcu_SetMode()</p> <p>This callout is invoked by EcuM periodically in SLEEP state, but only if the CPU is not suspended (i.e. clock is reduced).</p>

#### 5.4.3.4.64. EcuM\_SetWakeupEvent

<b>Purpose</b>	Set wakeup events.
----------------	--------------------



<b>Synopsis</b>	<code>void EcuM_SetWakeupEvent ( EcuM_WakeupSourceType sources );</code>	
<b>Service ID</b>	<a href="#">ECUM_SID_SET_WK_EV</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant Non-Interruptible	
<b>Parameters (in)</b>	sources	Events of wakeup sources to be set.
<b>Description</b>	<p>This function sets the wakeup events for a set of sources and starts the wakeup validation timeout timer.</p> <p>Development Error Detection</p> <ul style="list-style-type: none"> <li>➤ <a href="#">ECUM_E_UNKNOWN_WAKEUP_SOURCE</a>: sources contained unknown wakeup sources.</li> </ul>	

#### 5.4.3.4.65. EcuM\_Shutdown

<b>Purpose</b>	Shutdown ECU.
<b>Synopsis</b>	<code>void EcuM_Shutdown ( void );</code>
<b>Service ID</b>	<a href="#">ECUM_SID_SHUTDOWN</a>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Description</b>	Typically called from the shutdown hook, this function takes over execution control and will carry out GO OFF II activities.

#### 5.4.3.4.66. EcuM\_SleepActivity

<b>Purpose</b>	Callout during sleep state.
<b>Synopsis</b>	<code>void EcuM_SleepActivity ( void );</code>
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	<p>It is explicitly allowed to poll wakeup sources from this callout and to call wakeup notification functions to indicate the end of the sleep state to the ECU State Manager.</p> <p>This callout is invoked by EcuM periodically in SLEEP state, but only if the CPU is not suspended (i.e. clock is reduced).</p>



#### 5.4.3.4.67. EcuM\_StartWakeupSources

<b>Purpose</b>	Callout which starts pending wakeup sources.	
<b>Synopsis</b>	<pre>void <b>EcuM_StartWakeupSources</b> ( EcuM_WakeupSourceType wake-upSource );</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	wakeUpSource	All pending wakeup sources.
<b>Description</b>	<p>In this callout the system integrator has to set up the conditions, so that a wakeup source can validate a wakeup event. For example, set the tranceiver mode of a Can transceiver to STARTED.</p> <p>This callout is invoked in wakeup state.</p>	

#### 5.4.3.4.68. EcuM\_StartupTwo

<b>Purpose</b>	Startup sequence two.	
<b>Synopsis</b>	<pre>void <b>EcuM_StartupTwo</b> ( void );</pre>	
<b>Service ID</b>	<a href="#">ECUM_SID_STARTUP_TWO</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Description</b>	This function implements the STARTUP II state.	

#### 5.4.3.4.69. EcuM\_StopWakeupSources

<b>Purpose</b>	Callout which stops pending and expired wakeup sources.	
<b>Synopsis</b>	<pre>void <b>EcuM_StopWakeupSources</b> ( EcuM_WakeupSourceType wake-upSource );</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	wakeUpSource	All pending/expired wakeup sources.
<b>Description</b>	<p>In this callout the system integrator has to stop the wakeup sources, previously started in callout function <a href="#">EcuM_StartWakeupSources()</a>.</p> <p>This callout is invoked in wakeup validation state after a failed validation of an wakeup event.</p>	



#### 5.4.3.4.70. EcuM\_ValidateWakeupEvent

<b>Purpose</b>	Validate wakeup events.	
<b>Synopsis</b>	<pre>void <b>EcuM_ValidateWakeupEvent</b> ( EcuM_WakeupSourceType sources );</pre>	
<b>Service ID</b>	<a href="#">ECUM_SID_VALIDATE_WK_EV</a>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	sources	Wakeup sources which events shall be validated.
<b>Description</b>	<p>This function validates wakeup events. After wakeup, the ECU State Manager will stop the process during the WAKEUP VALIDATION state to wait for validation of the wakeup event. The validation is carried out with a call to this API service.</p> <p>Development Error Detection</p> <ul style="list-style-type: none"> <li>▶ <a href="#">ECUM_E_UNKNOWN_WAKEUP_SOURCE</a>: sources contained unknown wakeup sources.</li> </ul>	

### 5.4.4. Integration notes

#### 5.4.4.1. Exclusive areas

This section describes the exclusive areas used by the EcuM module.

##### 5.4.4.1.1. SCHM\_ECUM\_EXCLUSIVE\_AREA\_0

<b>Protected data structures</b>	All shared data that shall be protected from mutual access. Only one exclusive area is defined for a Single-Core configuration.
<b>Recommended locking mechanism</b>	This exclusive area must always be protected by a locking mechanism. The options for locking are described in the EB tresos AutoCore Generic documentation. Refer to



the section [Mapping exclusive areas in the basic software modules](#) in the [Integration notes section](#) for details.

#### 5.4.4.1.2. SCHM\_ECUM\_EXCLUSIVE\_AREA\_{coreID}

<b>Protected data structures</b>	All shared data that shall be protected from mutual access in the corresponding Ecum instance. If Multi-Core support is enabled, one exclusive area is defined for each configured Os core.
<b>Recommended locking mechanism</b>	This exclusive area must always be protected by a locking mechanism. The options for locking are described in the <a href="#">EB tresos AutoCore Generic documentation</a> . Refer to the section <a href="#">Mapping exclusive areas in the basic software modules</a> in the <a href="#">Integration notes section</a> for details.

#### 5.4.4.2. Production errors

<a href="#">ECUM_E_ALL_RUN_REQUESTS_KILLED</a>	▶ <a href="#">Ecum_KillAllRUNRequests</a>
<a href="#">ECUM_E_RAM_CHECK_FAILED</a>	▶ <a href="#">Ecum_MainFunction</a>

#### 5.4.4.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section [Memory mapping and compiler abstraction](#) in the [Integration notes section](#) for details.

The following table provides the list of sections that may be mapped for this module:

Memory section
INTERNAL_VAR_POWER_ON_INIT_UNSPECIFIED
VAR_POWER_ON_INIT_8
VAR_CLEARED_8
VAR_CLEARED_16
CONST_8



VAR_INIT_8
CONST_16
VAR_CLEARED_UNSPECIFIED
VAR_INIT_UNSPECIFIED
CONST_32
CONST_UNSPECIFIED
CONFIG_DATA_UNSPECIFIED
CODE
ECUM_BOOTTARGET

#### 5.4.4.4. Integration requirements

**WARNING**



**Integration requirements list is not exhaustive**

The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user's guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

##### 5.4.4.4.1. intgr.EcuM.EB\_INTREQ\_EcuM\_0001

<b>Description</b>	In order to ensure the EcuM_GetMostRecentShutdown() and EcuM_GetNextRecentShutdown() APIs return valid data, the following call chain shall be used in a shutdown or reset sequence: - EcuM_SelectShutdownTarget() - EcuM_SelectShutdownCause() - EcuM_GoDown()
<b>Rationale</b>	EcuM_GetMostRecentShutdown() and EcuM_GetNextRecentShutdown() return the values of the shutdown target, reset mode and shutdown cause. They will not return the expected most recent or next recent shutdown values if they are not previously set using the EcuM_SelectShutdownTarget() and EcuM_SelectShutdownCause() APIs.

## 5.5. Nm

### 5.5.1. Configuration parameters

**Containers included**

<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">CommonPublishedInformation</a>	1..1	<b>Label:</b> Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
<a href="#">NmChannelConfig</a>	1..n	Container for the configuration (parameters) of a Nm channel. The channel parameter shall be harmonized within the whole communication stack.
<a href="#">NmGlobalConfig</a>	1..1	This container contains all global configuration parameters of the Nm Interface.
<a href="#">PublishedInformation</a>	1..1	<b>Label:</b> EB Published Information Additional published parameters not covered by CommonPublishedInformation container.

**Parameters included**

<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">IMPLEMENTATION_CONFIG_VARIANT</a>	1..1

<b>Parameter Name</b>	<b>IMPLEMENTATION_CONFIG_VARIANT</b>	
<b>Label</b>	Config Variant	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	VariantPreCompile	
<b>Range</b>	VariantPreCompile	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile

**5.5.1.1. CommonPublishedInformation****Parameters included**

<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">ArMajorVersion</a>	1..1
<a href="#">ArMinorVersion</a>	1..1
<a href="#">ArPatchVersion</a>	1..1
<a href="#">SwMajorVersion</a>	1..1

**Parameters included**

<a href="#">SwMinorVersion</a>	1..1
<a href="#">SwPatchVersion</a>	1..1
<a href="#">ModuleId</a>	1..1
<a href="#">VendorId</a>	1..1
<a href="#">Release</a>	1..1

**Parameter Name****ArMajorVersion**

<b>Label</b>	AUTOSAR Major Version
<b>Description</b>	Major version number of AUTOSAR specification on which the appropriate implementation is based on.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	3
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

**Parameter Name****ArMinorVersion**

<b>Label</b>	AUTOSAR Minor Version
<b>Description</b>	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	0
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

**Parameter Name****ArPatchVersion**

<b>Label</b>	AUTOSAR Patch Version
<b>Description</b>	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	0



<b>Configuration class</b>	<b>PublishedInformation:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>SwMajorVersion</b>
<b>Label</b>	Software Major Version
<b>Description</b>	Major version number of the vendor specific implementation of the module.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	5
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SwMinorVersion</b>
<b>Label</b>	Software Minor Version
<b>Description</b>	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	12
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SwPatchVersion</b>
<b>Label</b>	Software Patch Version
<b>Description</b>	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	9
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>ModuleId</b>
<b>Label</b>	Numeric Module ID



<b>Description</b>	Module ID of this module from Module List
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	29
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>VendorId</b>
<b>Label</b>	Vendor ID
<b>Description</b>	Vendor ID of the dedicated implementation of this module according to the AU-TOSAR vendor list
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER_LABEL
<b>Default value</b>	1
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>Release</b>
<b>Label</b>	Release Information
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING_LABEL
<b>Default value</b>	
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.5.1.2. NmChannelConfig

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">NmBusType</a>	1..1	<p><b>Label:</b> Nm Bus Type</p> <p>Defines the bus type of Nm channel.</p> <p>► Select <code>NmGenericBusNmPrefix</code> for generic modules which the prefix identifies its Bus &lt;BusNm&gt;</p>

**Containers included**

		► Select <code>NmStandardBusNmConfig</code> for standard AU-TOSAR <BusNm>s
--	--	--

**Parameters included**

Parameter name	Multiplicity
<a href="#">NmActiveCoordinator</a>	0..1
<a href="#">NmChannelId</a>	1..1
<a href="#">NmChannelSleepMaster</a>	1..1
<a href="#">NmCoordClusterIndex</a>	0..1
<a href="#">NmNodeDetectionEnabled</a>	1..1
<a href="#">NmNodeIdEnabled</a>	1..1
<a href="#">NmPassiveModeEnabled</a>	1..1
<a href="#">NmRepeatMsgIndEnabled</a>	1..1
<a href="#">NmShutdownDelayTimer</a>	1..1
<a href="#">NmStateReportEnabled</a>	1..1
<a href="#">NmSynchronizingNetwork</a>	1..1
<a href="#">NmComMChannelRef</a>	1..1
<a href="#">NmStateReportSignalRef</a>	0..1
<a href="#">NmWaitForShutdownTime</a>	0..1

<b>Parameter Name</b>	<b>NmActiveCoordinator</b>
<b>Label</b>	Nm Active Gateway Enable
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Defines whether a NM Coordinator is an active gateway (<code>NmActiveCoordinator = TRUE</code>) or a passive.</p> <p>Scope:</p> <ul style="list-style-type: none"> <li>► Channel</li> </ul> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>► This feature is available only if the parameter <code>NmBusSynchronizationEnabled</code> is set to true.</li> </ul>
<b>Multiplicity</b>	0..1



Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	NmChannelId	
Label	Nm Channel Id	
Description	<p>Defines channel index value. Implementation Type: NetworkHandleType</p> <p>Range:</p> <ul style="list-style-type: none"> <li>▶ 0..254</li> </ul> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ The Channel id has to be unique</li> <li>▶ NmChannelId shall be the same as the ComMChannelId of the ComM-Channel referenced by NmComMChannelRef</li> </ul>	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	NmChannelSleepMaster	
Label	Nm Channel SleepMaster	
Description	<p>Defines if the sleep of this network can be absolutely decided by the local node only and that no other nodes can oppose that decision. If this parameter is set to TRUE, the Nm shall assume that the channel is always ready to go to sleep and that no callouts to Nm_RemoteSleepIndication() or Nm_RemoteSleepCancellation() will be made from the &lt;BusNm&gt; representing this channel. If this parameter is set to FALSE, the Nm shall not assume that the network is ready to sleep until a callout has been made to Nm_RemoteSleepCancellation().</p> <p>Scope:</p> <ul style="list-style-type: none"> <li>▶ Channel</li> </ul> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ If the parameter NmCoordClusterIndex is not defined, this parameter is not valid.</li> </ul>	



<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>NmCoordClusterIndex</b>			
<b>Label</b>	Nm Coordination Cluster Index			
<b>Description</b>	<p>Defines the possibility that the channel to be part of a coordinator cluster</p> <p>The configuration parameter <code>NmCoordClusterIndex</code> is used for specifying to which coordination cluster a channel belongs. If this parameter is undefined for a channel, the corresponding channel does not belong to an NM coordination cluster.</p> <p>The parameter is optional; by enabling it and entering a number "x", the channel is assigned to the Nm coordination cluster "x".</p> <p>Range:</p> <ul style="list-style-type: none"> <li>▶ 0..255</li> </ul> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ There should be at least two channels in a cluster.</li> <li>▶ Cluster indices must span a dense, zero-based number domain.</li> <li>▶ Passive mode shall be disabled if channel belongs to a co-ordinated cluster of networks</li> </ul>			
<b>Multiplicity</b>				
<b>Type</b>				
<b>Default value</b>	0			
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile		
<b>Origin</b>	AUTOSAR_ECUC			

<b>Parameter Name</b>	<b>NmNodeDetectionEnabled</b>	
<b>Label</b>	Nm Node Detection	
<b>Description</b>	<p>Enables the Node Detection feature.</p> <p>Enables the handling <i>Repeat Message Request Bit</i> in the <i>Control Bit Vector</i>.</p>	



	<p>If the <i>Request Message Bit</i> in the NM message set the nodes receiving the message start sending NM messages.</p> <p>For setting the <i>Repeat Message Request Bit</i> in NM messages following API function is provided:</p> <ul style="list-style-type: none"> <li>▶ Nm_RepeatMessageRequest ()</li> </ul> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Support for Node Identifiers must be enabled</li> <li>▶ Passive Mode must be disabled</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>NmNodeIdEnabled</b>
<b>Label</b>	Nm Node Identifier
<b>Description</b>	<p>Enables transmission of the source node identifier in NM messages.</p> <p>Enable support for sending and receiving of Node Ids in NM messages and provide functions for retrieving the node identifier from the most recently received NM PDU or the local node identifier.</p> <p>The following API functions are provided if the feature is enabled:</p> <ul style="list-style-type: none"> <li>▶ Nm_GetNodeIdentifier ()</li> <li>▶ Nm_GetLocalNodeIdentifier ()</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>NmPassiveModeEnabled</b>
<b>Label</b>	Nm Passive Mode
<b>Description</b>	Enables support for 'Passive Mode' of BusNms.



	<p>In passive mode, the BusNm modules will not be able to wake up the bus and will not send NM messages. They will only listen to the NM messages and silently monitor the bus.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Passive mode shall not be enabled if the channel belongs to a coordinated cluster of networks</li> <li>▶ Each channel shall have the same NmPassiveModeEnabled setting as the global NmPassiveModeEnabled.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>NmRepeatMsgIndEnabled</b>
<b>Label</b>	Nm Repeat Message Indication Enabled
<b>Description</b>	Enables the Repeat Message Bit Indication.
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>NmShutdownDelayTimer</b>
<b>Label</b>	Nm Shutdown Delay Timer
<b>Description</b>	<p>Defines the time in seconds which the NM Coordination algorithm shall delay the release of this channel with.</p> <p>This time must be elapsed after starting synchronous shutdown, before an NM channel starts "fall asleep".</p> <p>Range:</p> <ul style="list-style-type: none"> <li>▶ 0..65535</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT



<b>Default value</b>	0.0
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>NmStateReportEnabled</b>
<b>Label</b>	Nm State Reporting
<b>Description</b>	<p>Enables whether the NMS shall be set for the corresponding network.</p> <ul style="list-style-type: none"> <li>▶ False: No NMS shall be set</li> <li>▶ True: The NMS shall be set</li> </ul> <p>Scope:</p> <ul style="list-style-type: none"> <li>▶ Module</li> </ul> <p>Dependency on parameter(s)::</p> <ul style="list-style-type: none"> <li>▶ If State Reporting is enabled parameters 'NmStateChangeIndEnabled' and 'NmComUserDataSupport' shall be enabled.</li> <li>▶ If State Reporting is enabled a Com Signal Reference (NmStateReportSignalRef) shall be configured.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>NmSynchronizingNetwork</b>
<b>Label</b>	Nm Synchronizing Network
<b>Description</b>	<p>Defines if this network (if set to true) is a synchronizing network for the NM co-ordination cluster which it belongs to. The network is expected to call Nm_SynchronizationPoint() at regular intervals</p> <p>Scope:</p> <ul style="list-style-type: none"> <li>▶ Channel</li> </ul> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ If the parameter NmCoordClusterIndex is not defined, this parameter is not valid.</li> </ul>



<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	NmComMChannelRef
<b>Label</b>	ComM Channel Reference
<b>Description</b>	<p>Reference to the corresponding network (ComM channel).</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Bus type of NM channel must match bus type of referenced network channel (ComM).</li> <li>▶ The Passive Mode of the Nm channel must be disabled if the referenced network channel (ComM) expects the Nm variant 'FULL'.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	NmStateReportSignalRef
<b>Label</b>	ComSignal Reference for NMS
<b>Description</b>	<p>Reference to the signal for setting the NMS by calling Com_SendSignal for the respective channel</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Signal must be configured in COM. Only available if NmStateReportEnabled == true</li> </ul>
<b>Multiplicity</b>	0..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>PreCompile:</b> VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	NmWaitForShutdownTime
<b>Label</b>	Nm Wait For Shutdown Time



<b>Description</b>	<p>The time is defined in seconds</p> <p>In case parameter NmWaitForShutdownTime is enabled for a channel and the BusNm stays more is Synchronize State then the configured value (NmWaitForShutdownTime) the channel shall be considered inactive.</p> <p>In case parameter NmWaitForShutdownTime is enabled for a channel and the BusNm is in Buss Sleep State (because of Buss Off) and channel has NmSynchronizingNetwork parameter set to TRUE, the Coordination cluster shall wait for a Synchronization Point the number of malfunctions.</p> <p>The parameter is optional; It shall be only available if NmCoordClusterIndex is specified for the channel.</p> <p>Range:</p> <ul style="list-style-type: none"> <li>▶ 0..65535</li> </ul>
<b>Multiplicity</b>	0..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.0
<b>Configuration class</b>	<b>PreCompile:</b> VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.5.1.3. NmBusType

Containers included		
Container name	Multiplicity	Description
<a href="#">NmGenericBusNmConfig</a>	1..1	
<a href="#">NmStandardBusNmConfig</a>	1..1	

### 5.5.1.4. NmGenericBusNmConfig

Parameters included	
Parameter name	Multiplicity
<a href="#">NmGenericBusNmPrefix</a>	1..1
<b>Parameter Name</b>	<b>NmGenericBusNmPrefix</b>



<b>Description</b>	Defines the prefix which identifies the generic <BusNm>. This will be used to determine the API name to be called by Nm for the provided interfaces of the <BusNm>. This string will be used for the module prefix before the "_" character in the API call name  There should be a header file available for this generic Nm as <code>NmGenericBus-NmPrefix.h</code>  Dependency on parameter(s): <ul style="list-style-type: none"><li>▶ Generic Bus Nm cannot be an empty string.</li><li>▶ Generic Nm prefix must start with a letter. It can contain alphabets, numbers, '_'</li></ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC

### 5.5.1.5. NmStandardBusNmConfig

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">NmStandardBusType</a>	1..1

<b>Parameter Name</b>	<b>NmStandardBusType</b>
<b>Description</b>	Defines the bus type of the channel for standard AUTOSAR <BusNm>s and is used to determine which set of API calls to be called by Nm for the <BusNm>s. Note: The Ethernet bus' NM is UdpNm ! LinNm is not supported as standard bus type  Range: <ul style="list-style-type: none"><li>▶ NM_BUSNM_CANNM</li><li>▶ NM_BUSNM_FRNM</li><li>▶ NM_BUSNM_LINNM</li><li>▶ NM_BUSNM_UDPNM</li></ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION



<b>Range</b>	NM_BUSNM_CANNM NM_BUSNM_FRNM NM_BUSNM_LINNM NM_BUSNM_UDPNM
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC

### 5.5.1.6. NmGlobalConfig

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">NmGlobalConstants</a>	1..1	
<a href="#">NmGlobalFeatures</a>	1..1	
<a href="#">NmGlobalProperties</a>	1..1	

### 5.5.1.7. NmGlobalConstants

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">NmNumberOfChannels</a>	1..1

<b>Parameter Name</b>	<b>NmNumberOfChannels</b>
<b>Label</b>	Nm Number Of Channels
<b>Description</b>	<p>Defines the number of NM channels allowed within one ECU</p> <p>In this Nm implementation this number must be equal to the number of entries in the list of NM channels.</p> <p>Range:</p> <ul style="list-style-type: none"> <li>▶ 1..255</li> </ul> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Number of channels must match to the number of entries in the channel list.</li> </ul>
<b>Multiplicity</b>	1..1



Type	INTEGER
Configuration class	<b>VariantPreCompile:</b> VariantPreCompile
Origin	AUTOSAR_ECUC

### 5.5.1.8. NmGlobalFeatures

Parameters included	
Parameter name	Multiplicity
<a href="#">NmMultiCoreSupport</a>	1..1
<a href="#">NmBusSynchronizationEnabled</a>	1..1
<a href="#">NmCarWakeUpCallback</a>	0..1
<a href="#">NmCarWakeUpRxEnabled</a>	1..1
<a href="#">NmComControlEnabled</a>	1..1
<a href="#">NmComUserDataSupport</a>	1..1
<a href="#">NmCoordinatorSupportEnabled</a>	1..1
<a href="#">NmCoordinatorSyncSupport</a>	0..1
<a href="#">NmGlobalCoordinatorTime</a>	0..1
<a href="#">NmPassiveModeEnabled</a>	0..1
<a href="#">NmPduRxIndicationEnabled</a>	1..1
<a href="#">NmProvideRemoteSleepCallbacks</a>	1..1
<a href="#">NmRemoteSleepIndicationCallback</a>	1..1
<a href="#">NmRemoteSleepCancellationCallback</a>	1..1
<a href="#">NmRemoteSleepIndEnabled</a>	1..1
<a href="#">NmStateChangeIndEnabled</a>	1..1
<a href="#">NmStateChangeNotificationCallout</a>	0..1
<a href="#">NmUserDataEnabled</a>	1..1

Parameter Name	<a href="#">NmMultiCoreSupport</a>
Label	Nm multicore support
Description	Enables MultiCoreSupport.
Multiplicity	1..1
Type	BOOLEAN
Default value	false



<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>NmBusSynchronizationEnabled</b>	
<b>Label</b>	Nm Bus Synchronization Enabled	
<b>Description</b>	<p>Enables buss synchronization support of the &lt;BusNm&gt;s. This feature is required for NM Coordinator nodes only. It triggers the synchronization of a CanNm bus after a remote sleep indication has been notified.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Passive Mode must be disabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>NmCarWakeUpCallback</b>	
<b>Label</b>	Nm Car WakeUp Callback Func	
<b>Description</b>	<p>Name of the callback function to be called if <code>Nm_CarWakeUpIndication()</code> is called.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ This parameter is only available if <code>NmCarWakeUpRxEnabled == TRUE</code></li> </ul>	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FUNCTION-NAME	
<b>Default value</b>		
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>NmCarWakeUpRxEnabled</b>	
<b>Label</b>	Nm CarWakeUp Rx Enable	
<b>Description</b>	<p>Enables or disables CWU detection.</p> <ul style="list-style-type: none"> <li>▶ FALSE - CarWakeUp not supported</li> <li>▶ TRUE - CarWakeUp supported</li> </ul>	



<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>NmComControlEnabled</b>	
<b>Label</b>	Nm Communication Control	
<b>Description</b>	<p>Enables the Communication Control support.</p> <p>The following API functions are provided if the feature is enabled:</p> <ul style="list-style-type: none"> <li>▶ <code>Nm_EnableCommunication()</code></li> <li>▶ <code>Nm_DisableCommunication()</code></li> </ul> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ At least one NM channel must have bus type <code>NM_BUSNM_CANNM</code> or <code>NM_BUSNM_FRNM</code> or <code>NM_BUSNM_UDPNM</code>.</li> <li>▶ Passive Mode must be disabled.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>NmComUserDataSupport</b>	
<b>Label</b>	Nm Com User Data Support	
<b>Description</b>	<p>Enables setting of NMUserData via SW-C. If NmComUserDataSupport is enabled the API <code>Nm_SetUserData</code> shall not be available.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ If buss type is FrNm, FrNmComUserDataSupport should have the same value as NmComUserDataSupport</li> <li>▶ If buss type is CanNm, CanNmComUserDataSupport should have the same value as NmComUserDataSupport</li> <li>▶ If buss type is UdpNm, UdpNmComUserDataSupport should have the same value as NmComUserDataSupport</li> </ul>	



<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>NmCoordinatorSupportEnabled</b>	
<b>Label</b>	Enable Nm coordinator	
<b>Description</b>	<p>Enables NM Coordinator support.</p> <p>Dependency:</p> <p>Only valid if NmRemoteSleepIndEnabled set to TRUE and NmPassiveModeEnabled set to FALSE. Also there must exist more than one Nm channel.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ If NmCoordinatorSupportEnabled is enabled, there need to be at least two channels being assigned to the same cluster.</li> <li>▶ Passive Mode must not be enabled for all channels</li> <li>▶ NmBusSynchronizationEnabled must be enabled</li> <li>▶ NmRemoteSleepIndEnabled must be enabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>NmCoordinatorSyncSupport</b>	
<b>Label</b>	Nm Coordinator Ready To Sleep Indication	
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This parameter control activation of interface used by BusNM to notify reception of "NM Coordinator Sleep Ready" bit. Pre-processor switch for enabling NM coordinator synchronization support. The following API function is provided if the feature is enabled:</p> <ul style="list-style-type: none"> <li>▶ Nm_CoordReadyToSleepIndication()</li> </ul>	



<b>Multiplicity</b>	0..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	PreCompile:	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>NmGlobalCoordinatorTime</b>	
<b>Label</b>	Nm Global Coordinator Time	
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This parameter defines the maximum shutdown time of a connected and coordinated NM-Cluster</p> <p>Note: This includes nested connections.</p> <p>Dependencies:</p> <p>This feature is available only if <code>NmBusSynchronizationEnabled</code> set to 'TRUE'</p> <p>Range:</p> <ul style="list-style-type: none"> <li>▶ 0..Infinity</li> </ul>	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0	
<b>Configuration class</b>	PreCompile:	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>NmPassiveModeEnabled</b>	
<b>Label</b>	Nm Passive Mode Enabled	
<b>Description</b>	Enables Global switch which determines if all channels have the same passive mode value.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	PreCompile:	VariantPreCompile



<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

<b>Parameter Name</b>	<b>NmPduRxIndicationEnabled</b>	
<b>Label</b>	PduR Rx Indication	
<b>Description</b>	<p>Enables the PDU Rx Indication</p> <p>Provide API to BusNm modules for indication of a received NM message:</p> <ul style="list-style-type: none"> <li>▶ <code>Nm_PduRxIndication()</code></li> </ul> <p>If this parameter is enabled, the Nm module will generate a declaration for the function <code>Nm_PduRxIndication()</code> as described in the SWS.</p> <p>It is the task of the integrator/user to provide the definition of the function!</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>NmProvideRemoteSleepCallbacks</b>	
<b>Label</b>	Nm Provide Remote Sleep Callbacks	
<b>Description</b>	If this parameter is set to true, user can provide callback functions for <code>Nm_RemoteSleepCancellation</code> and <code>Nm_RemoteSleepIndication</code>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>NmRemoteSleepIndicationCallback</b>	
<b>Label</b>	Nm Remote Sleep Indication Callback	
<b>Description</b>	Callback function name for <code>Nm_RemoteSleepIndication</code>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FUNCTION-NAME	
<b>Default value</b>		
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile



<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

<b>Parameter Name</b>	<b>NmRemoteSleepCancellationCallback</b>	
<b>Label</b>	Nm Remote Sleep Cancellation Callback	
<b>Description</b>	Callback function name for Nm_RemoteSleepCancellation	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FUNCTION-NAME	
<b>Default value</b>		
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>NmRemoteSleepIndEnabled</b>	
<b>Label</b>	Nm Remote Sleep Indication Enabled	
<b>Description</b>	<p>Enables Remote Sleep Indication support. This feature is required for NM Coordinator nodes only.</p> <p>The Remote Sleep Indication allows the BusNm modules to detect a situation where all nodes in the cluster are ready to sleep apart from one node which still keeps the bus awake.</p> <p>The following API functions are provided if the feature is enabled:</p> <ul style="list-style-type: none"> <li>▶ <code>Nm_CheckRemoteSleepIndication()</code></li> </ul> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Passive Mode must be disabled</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>NmStateChangeIndEnabled</b>	
<b>Label</b>	Nm State Change Indication	
<b>Description</b>	<p>Enables the Network Management state change notification.</p> <p>Provide API to BusNm modules for indication of a state change:</p>	



	<ul style="list-style-type: none"> <li>▶ <code>Nm_StateChangeNotification()</code></li> </ul> <p>If this parameter is enabled, the Nm module will provide the callback function <code>Nm_StateChangeNotification()</code> as described in the SWS.</p>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	AUTOSAR_ECU

<b>Parameter Name</b>	<b>NmStateChangeNotificationCallout</b>
<b>Label</b>	Provide State Change Notification Callout
<b>Description</b>	<p>Enables the callout function <code>Nm_StateChangeNotificationCallout()</code> to be called within the callback function <code>Nm_StateChangeNotification()</code>.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ This parameter is only available if <code>NmStateChangeIndEnabled == TRUE</code></li> </ul>
<b>Multiplicity</b>	0..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>PreCompile:</b> VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>NmUserDataEnabled</b>
<b>Label</b>	Nm User Data
<b>Description</b>	<p>Enables User Data support.</p> <p>The following API functions are provided if the feature is enabled:</p> <ul style="list-style-type: none"> <li>▶ <code>Nm_GetUserData()</code></li> <li>▶ <code>Nm_SetUserData()</code> (<i>Only if Passive Mode is disabled for at least one channel and NmComUserDataSupport is disabled</i>)</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile



<b>Origin</b>	AUTOSAR_ECUC
---------------	--------------

### 5.5.1.9. NmGlobalProperties

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">NmCycletimeMainFunction</a>	1..1
<a href="#">NmDevErrorDetect</a>	1..1
<a href="#">NmVersionInfoApi</a>	1..1

<b>Parameter Name</b>	<b>NmCycletimeMainFunction</b>	
<b>Label</b>	Main Function Cycle Time [s]	
<b>Description</b>	<p>Defines the period between successive calls to the Main Function of the NM Interface in seconds.</p> <p>Calling the main function is only required if Nm coordinator functionality is enabled.</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.02	
<b>Configuration class</b>	<a href="#">VariantPreCompile:</a>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>NmDevErrorDetect</b>	
<b>Label</b>	Enable Development Error Detection	
<b>Description</b>	Enables development error detection and notification.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	true	
<b>Configuration class</b>	<a href="#">VariantPreCompile:</a>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>NmVersionInfoApi</b>	
<b>Label</b>	Enable Version Info API	
<b>Description</b>	Enables version information support	



	▶ Nm_GetVersionInfo()
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC

### 5.5.1.10. PublishedInformation

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
PbcfgMSupport	1..1

<b>Parameter Name</b>	<b>PbcfgMSupport</b>
<b>Label</b>	PbcfgM support
<b>Description</b>	Specifies whether or not the Nm can use the PbcfgM module for post-build support.
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

## 5.5.2. Application programming interface (API)

### 5.5.2.1. Type definitions

#### 5.5.2.1.1. Nm\_BusNmTpType

<b>Purpose</b>	Type holding functions pointers to BusNm functions.
<b>Type</b>	struct



<b>Members</b>	Nm_TS_BusNm_StdFPtrType PassiveStartup	Points to the BusNm_PassiveStartup() function of the lower layer modules.
	Nm_TS_BusNm_GetStateFPtrType GetState	Points to the BusNm_GetState() function of the lower layer modules.
	Nm_TS_BusNm_StdFPtrType NetworkRequest	Points to the BusNm_NetworkRequest() function of the lower layer modules.
	Nm_TS_BusNm_StdFPtrType NetworkRelease	Points to the BusNm_NetworkRelease() function of the lower layer modules.
	Nm_TS_BusNm_GetInfoFPtrType GetUserData	Points to the BusNm_GetUserData() function of the lower layer modules.
	Nm_TS_BusNm_SetUserDataFPtrType SetUserData	Points to the BusNm_SetState() function of the lower layer modules.
	Nm_TS_BusNm_GetInfoFPtrType GetNodeIdentifier	Points to the BusNm_GetNodeIdentifier() function of the lower layer modules.
	Nm_TS_BusNm_GetInfoFPtrType GetLocalNodeIdentifier	Points to the BusNm_GetLocalNodeIdentifier() function of the lower layer modules.
	Nm_TS_BusNm_StdFPtrType RepeatMessageRequest	Points to the BusNm_RepeatMessageRequest() function of the lower layer modules.
	Nm_TS_BusNm_GetInfoFPtrType GetPduData	Points to the BusNm_GetPduData() function of the lower layer modules.
	Nm_TS_BusNm_StdFPtrType RequestBusSynchronization	Points to the BusNm_RequestBusSynchronization() function of the lower layer modules.
	Nm_TS_BusNm_ChkRemoteSleepIndFPtrType CheckRemoteSleepIndication	Points to the BusNm_CheckRemoteSleepIndication() function of the lower layer modules.
	Nm_TS_BusNm_StdFPtrType DisableCommunication	Points to the BusNm_DisableCommunication() function of the lower layer modules.
	Nm_TS_BusNm_StdFPtrType EnableCommunication	Points to the BusNm_EnableCommunication() function of the lower layer modules.
<b>Description</b>	This type is only used internally in the Nm but must be published in the public module API to enable optimisations.	

### 5.5.2.1.2. Nm\_BusNmType

<b>Purpose</b>	BusNm types.
----------------	--------------



Type	uint8
------	-------

#### 5.5.2.1.3. Nm\_ModeType

Purpose	Operational modes of the network management.
Type	uint8

#### 5.5.2.1.4. Nm\_StateType

Purpose	States of the network management state machine.
Type	uint8

#### 5.5.2.1.5. Nm\_TS\_BusNm\_ChkRemoteSleepIndFPtrType

Purpose	Function Pointer for Nm functions with return type Std_ReturnType and 2 parameters as for BusNm_CheckRemoteSleepIndication() .
Type	Std_ReturnType (*) (NetworkHandleType nmChannelHandle, boolean *nmRemoteSleepIndPtr)

#### 5.5.2.1.6. Nm\_TS\_BusNm\_GetInfoFPtrType

Purpose	Function Pointer for Nm functions with return type Std_ReturnType and 2 parameters as for BusNm_GetUserData() .
Type	Std_ReturnType (*) (NetworkHandleType nmChannelHandle, uint8 *nmPduDataPtr)

#### 5.5.2.1.7. Nm\_TS\_BusNm\_GetStateFPtrType

Purpose	Function Pointer for Nm functions with return type Std_ReturnType and 3 parameters as for BusNm_GetState() .
Type	Std_ReturnType (*) (NetworkHandleType nmChannelHandle, Nm_StateType *nmStatePtr, Nm_ModeType *nmModePtr)



#### 5.5.2.1.8. Nm\_TS\_BusNm\_SetUserDataFPtrType

<b>Purpose</b>	Function Pointer for Nm functions with return type Std_ReturnType and 2 parameters as for BusNm_SetUserData() .
<b>Type</b>	Std_ReturnType (*) (NetworkHandleType nmChannelHandle, const uint8 *nmPduDataPtr)

#### 5.5.2.1.9. Nm\_TS\_BusNm\_StdCbkFPtrType

<b>Purpose</b>	Function pointer for Nm callback functions with return type void and parameter of NetworkHandleType.
<b>Type</b>	void (*) (NetworkHandleType nmChannelHandle)

#### 5.5.2.1.10. Nm\_TS\_BusNm\_StdFPtrType

<b>Purpose</b>	Function Pointer for standard Nm functions with return type Std_ReturnType and parameter of NetworkHandleType.
<b>Type</b>	Std_ReturnType (*) (NetworkHandleType nmChannelHandle)

### 5.5.2.2. Macro constants

#### 5.5.2.2.1. NM\_AR\_RELEASE\_MAJOR\_VERSION

<b>Purpose</b>	AUTOSAR release major version.
<b>Value</b>	4U

#### 5.5.2.2.2. NM\_AR\_RELEASE\_MINOR\_VERSION

<b>Purpose</b>	AUTOSAR release minor version.
<b>Value</b>	0U

#### 5.5.2.2.3. NM\_AR\_RELEASE\_REVISION\_VERSION

<b>Purpose</b>	AUTOSAR release revision version.
----------------	-----------------------------------



<b>Value</b>	3U
--------------	----

#### 5.5.2.2.4. NM\_BUSNM\_CANNM

<b>Purpose</b>	CAN NM type.
<b>Value</b>	0U

#### 5.5.2.2.5. NM\_BUSNM\_FRNM

<b>Purpose</b>	FR NM type.
<b>Value</b>	1U

#### 5.5.2.2.6. NM\_BUSNM\_GENERICNM

<b>Purpose</b>	Generic NM type.
<b>Value</b>	4U

#### 5.5.2.2.7. NM\_BUSNM\_LINNM

<b>Purpose</b>	LIN NM type.
<b>Value</b>	2U

#### 5.5.2.2.8. NM\_BUSNM\_UDPNM

<b>Purpose</b>	UDP NM type.
<b>Value</b>	3U

#### 5.5.2.2.9. NM\_BUSNM\_UNDEF

<b>Purpose</b>	NM type undefined.
----------------	--------------------



<b>Value</b>	0xFFU
--------------	-------

#### 5.5.2.2.10. NM\_CC\_STATE\_PREPARE\_SHUTDOWN

<b>Purpose</b>	Timers are initialized for the shutdown.
<b>Value</b>	2U

#### 5.5.2.2.11. NM\_CC\_STATE\_SHUTDOWN\_ABORTED

<b>Purpose</b>	A shudown was aborted and the conditions are rechecked in the next main function call.
<b>Value</b>	4U

#### 5.5.2.2.12. NM\_CC\_STATE\_SHUTDOWN\_INITIATED

<b>Purpose</b>	The shutdown sequence is currently running.
<b>Value</b>	3U

#### 5.5.2.2.13. NM\_CC\_STATE\_WAIT\_FOR\_SHUTDOWN

<b>Purpose</b>	Wait for the shutdown conditions to become true. This is checked in all relevant Nm functions syncronously.
<b>Value</b>	0U

#### 5.5.2.2.14. NM\_CC\_STATE\_WAIT\_SYNCHRONIZATION

<b>Purpose</b>	Wait for synchronization if a synchronizing network is part of the cluster.
<b>Value</b>	1U

#### 5.5.2.2.15. NM\_E\_HANDLE\_UNDEF

<b>Purpose</b>	Development Error Code.
<b>Value</b>	1U



<b>Description</b>	This error is reported if an API function is called with invalid parameters; e.g. an invalid channel ID
--------------------	---

#### 5.5.2.2.16. NM\_E\_NETWORKGWERAREQUEST

<b>Purpose</b>	Error id reported in case SchM call fails for NetworkGwEraRequest.
<b>Value</b>	251U

#### 5.5.2.2.17. NM\_E\_NETWORKRELEASE

<b>Purpose</b>	Error id reported in case SchM call fails for NetworkRelease.
<b>Value</b>	248U

#### 5.5.2.2.18. NM\_E\_NETWORKREQUEST

<b>Purpose</b>	Error id reported in case SchM call fails for NetworkRequest.
<b>Value</b>	249U

#### 5.5.2.2.19. NM\_E\_PARAM\_POINTER

<b>Purpose</b>	Development Error Code.
<b>Value</b>	2U
<b>Description</b>	This error is reported if an API function is called with NULL pointer; e.g. NULL_PTR

#### 5.5.2.2.20. NM\_E\_PASSIVESTARTUP

<b>Purpose</b>	Error id reported in case SchM call fails for PassiveStartup.
<b>Value</b>	250U

#### 5.5.2.2.21. NM\_E\_REQUESTBUSSYNCHRONIZATION

<b>Purpose</b>	Error id reported in case SchM call fails for RequestBusSynchronization.
----------------	--



<b>Value</b>	247U
--------------	------

#### 5.5.2.2.22. NM\_E\_UNINIT

<b>Purpose</b>	Development Error Code.
<b>Value</b>	0U
<b>Description</b>	This error is reported if an API function is called prior to initialization of the Nm module.

#### 5.5.2.2.23. NM\_MODE\_BUS\_SLEEP

<b>Purpose</b>	Bus-Sleep Mode.
<b>Value</b>	0U

#### 5.5.2.2.24. NM\_MODE\_NETWORK

<b>Purpose</b>	Network Mode.
<b>Value</b>	3U

#### 5.5.2.2.25. NM\_MODE\_PREPARE\_BUS\_SLEEP

<b>Purpose</b>	Prepare-Bus Sleep Mode.
<b>Value</b>	1U

#### 5.5.2.2.26. NM\_MODE\_SYNCHRONIZE

<b>Purpose</b>	Synchronize Mode.
<b>Value</b>	2U

#### 5.5.2.2.27. NM\_MODULE\_ID

<b>Purpose</b>	AUTOSAR module identification.
<b>Value</b>	29U



#### 5.5.2.2.28. NM\_STATE\_BUS\_SLEEP

Purpose	
Value	1U

#### 5.5.2.2.29. NM\_STATE\_NORMAL\_OPERATION

Purpose	
Value	4U

#### 5.5.2.2.30. NM\_STATE\_OFFLINE

Purpose	
Value	7U

#### 5.5.2.2.31. NM\_STATE\_PREPARE\_BUS\_SLEEP

Purpose	
Value	2U

#### 5.5.2.2.32. NM\_STATE\_READY\_SLEEP

Purpose	
Value	3U

#### 5.5.2.2.33. NM\_STATE\_REPEAT\_MESSAGE

Purpose	
Value	5U

#### 5.5.2.2.34. NM\_STATE\_SYNCHRONIZE

Purpose	
---------	--



<b>Value</b>	6U
--------------	----

#### 5.5.2.2.35. NM\_STATE\_UNINIT

<b>Purpose</b>	
<b>Value</b>	0U

#### 5.5.2.2.36. NM\_SW\_MAJOR\_VERSION

<b>Purpose</b>	AUTOSAR module major version.
<b>Value</b>	5U

#### 5.5.2.2.37. NM\_SW\_MINOR\_VERSION

<b>Purpose</b>	AUTOSAR module minor version.
<b>Value</b>	12U

#### 5.5.2.2.38. NM\_SW\_PATCH\_VERSION

<b>Purpose</b>	AUTOSAR module patch version.
<b>Value</b>	9U

#### 5.5.2.2.39. NM\_VENDOR\_ID

<b>Purpose</b>	AUTOSAR vendor identification: Elektrobit Automotive GmbH.
<b>Value</b>	1U

#### 5.5.2.2.40. Nm\_RemoteSleepCancelation

<b>Purpose</b>	Work around for typos in the AUTOSAR SWS documents.
<b>Value</b>	Nm_RemoteSleepCancellation(nmNetworkHandle)



### 5.5.2.3. Objects

#### 5.5.2.3.1. Nm\_BusNmFp

<b>Purpose</b>	Array holding function pointers to BusNm modules.
<b>Type</b>	const <a href="#">Nm_BusNmFpType</a>
<b>Description</b>	This array is only used internally in the Nm but must be published in the public module API to enable optimisations.

### 5.5.2.4. Functions

#### 5.5.2.4.1. Nm\_BusSleepMode

<b>Purpose</b>	Notifies that Bus Sleep Mode has been entered.	
<b>Synopsis</b>	<code>void Nm_BusSleepMode ( NetworkHandleType nmNetworkHandle );</code>	
<b>Service ID</b>	32	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	nmNetworkHandle	Identification of the NM-channel.
<b>Description</b>	<p>This function provides a notification that the network management has entered Bus-Sleep Mode.</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>▶ The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>	

#### 5.5.2.4.2. Nm\_CheckRemoteSleepIndication

<b>Purpose</b>	Implementation of Nm_CheckRemoteSleepIndication.	
<b>Synopsis</b>	<code>Std_ReturnType Nm_CheckRemoteSleepIndication ( NetworkHandleType nmNetworkHandle , boolean * nmRemoteSleepIndPtr );</code>	
<b>Service ID</b>	13	



<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	nmNetworkHandle	Identification of the NM-channel.
<b>Parameters (out)</b>	nmRemoteSleepIndPtr	Pointer where check result of remote sleep indication shall be copied to.
<b>Return Value</b>	Standard Return Code E_OK                          No error. E_NOT_OK                      Checking of remote sleep indication bits has failed. NetworkHandle does not exist (development only). Module not yet initialized (development only).	
<b>Description</b>	<p>This function checks if remote sleep indication takes place or not. This in turn calls the BusNm_CheckRemoteSleepIndication() for the bus specific NM layer (e.g. CanNm_CheckRemoteSleepIndication() function is called if channel is configured as CAN).</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>► The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>	

#### 5.5.2.4.3. Nm\_CoordReadyToSleepIndication

<b>Purpose</b>	Indicate Ready to Sleep.	
<b>Synopsis</b>	<code>void Nm_CoordReadyToSleepIndication ( NetworkHandleType nmChannelHandle );</code>	
<b>Parameters (in)</b>	nmChannelHandle	Identification of the NM-channel
<b>Description</b>	<p>Sets an indication, when the NM Coordinator Sleep Ready bit in the Control Bit Vector is set</p> <p>The functionality needs to be implemented by the customer, as it is not specified by AUTOSAR.</p>	

#### 5.5.2.4.4. Nm\_DisableCommunication

<b>Purpose</b>	Disable NM-PDU transmissions.
----------------	-------------------------------



<b>Synopsis</b>	Std_ReturnType <b>Nm_DisableCommunication</b> ( NetworkHandleType NetworkHandle );	
<b>Service ID</b>	4	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant for the same NetworkHandle. Reentrant otherwise	
<b>Parameters (in)</b>	NetworkHandle	Identification of the NM-channel.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Disabling of NM PDU transmission ability has failed. NetworkHandle does not exist (development only). The module is not yet initialized (development only).
<b>Description</b>	<p>This function calls the respective function at CanNm module, to disable the NM PDU transmission ability due to a ISO14229 Communication Control (28hex) service.</p> <p>This functionality is not provided by FrNm module.</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>➤ The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>	

#### 5.5.2.4.5. Nm\_EnableCommunication

<b>Purpose</b>	Enable NM-PDU transmissions.	
<b>Synopsis</b>	Std_ReturnType <b>Nm_EnableCommunication</b> ( NetworkHandleType NetworkHandle );	
<b>Service ID</b>	5	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant for the same NetworkHandle. Reentrant otherwise	
<b>Parameters (in)</b>	NetworkHandle	Identification of the NM-channel.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Enabling of NM PDU transmission ability has failed. NetworkHandle does not exist (development only). The module is not yet initialized (development only).



<b>Description</b>	<p>This function calls the respective function at CanNm module to enable the NM PDU transmission ability due to a ISO14229 Communication Control (28hex) service.</p> <p>This functionality is not provided by FrNm module.</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>➤ The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>
--------------------	--

#### 5.5.2.4.6. Nm\_GetCoordinatorState

<b>Purpose</b>	Get the State of the Nm Coordinator.	
<b>Synopsis</b>	Std_ReturnType <b>Nm_GetCoordinatorState</b> ( uint8 nmCoordinatorId , uint8 * nmStatePtr );	
<b>Parameters (in)</b>	nmCoordinatorId	Identification of the coordinator.
	nmStatePtr	Pointer where state of the Coordinator shall be copied to.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Getting of NM state has failed.
<b>Description</b>	<p>This function returns the state of the Nm coordinator.</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>➤ The coordinator id should be valid and the module should have been initialized</li> </ul>	

#### 5.5.2.4.7. Nm\_GetLocalNodeIdentifier

<b>Purpose</b>	Configure the get node identifier for the local node.
<b>Synopsis</b>	Std_ReturnType <b>Nm_GetLocalNodeIdentifier</b> ( NetworkHandleType NetworkHandle , uint8 * nmNodeIdPtr );
<b>Service ID</b>	11
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant for the same NetworkHandle. Reentrant otherwise.



<b>Parameters (in)</b>	NetworkHandle	Identification of the NM-channel.
<b>Parameters (out)</b>	nmNodeIdPtr	Pointer where node identifier of the local node shall be copied to.
<b>Return Value</b>	Standard Return Code	
	E_OK	No error.
	E_NOT_OK	Getting of the node identifier of the local node has failed. NetworkHandle does not exist (development only). Module not yet initialized (development only).
<b>Description</b>	<p>The function configures the Get node identifier for the local node. For that purpose BusNm_GetLocalNodentifier shall be called (e.g. CanNm_GetLocalNodentifier function is called if channel is configured as CAN).</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>► The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>	

#### 5.5.2.4.8. Nm\_GetNodentifier

<b>Purpose</b>	Get the Node Identifier.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>Nm_GetNodeIdentifier</b> ( NetworkHandleType NetworkHandle , uint8 * nmNodeIdPtr );</pre>	
<b>Service ID</b>	10	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant for the same NetworkHandle. Reentrant otherwise	
<b>Parameters (in)</b>	NetworkHandle	Identification of the NM-channel.
<b>Parameters (out)</b>	nmNodeIdPtr	Pointer where node identifier out of the last successfully received NM-message shall be copied to.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Getting of the node identifier out of the last received NM-message has failed. NetworkHandle does not exist (development only). Module not yet initialized (development only).



<b>Description</b> <p>This function gets the node identifier out of the last successfully received NM-message.</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>▶ The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>
--

#### 5.5.2.4.9. Nm\_GetPduData

<b>Purpose</b> <p>Get Pdu Data.</p>		
<b>Synopsis</b> <pre>Std_ReturnType <b>Nm_GetPduData</b> ( NetworkHandleType NetworkHandle , uint8 * nmPduData );</pre>		
<b>Service ID</b> <p>8</p>		
<b>Sync/Async</b> <p>Synchronous</p>		
<b>Reentrancy</b> <p>Non-Reentrant for the same NetworkHandle. Reentrant otherwise</p>		
<b>Parameters (in)</b>	NetworkHandle 	Identification of the NM-channel.
	nmPduData 	Pointer where NM PDU shall be copied to.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Getting of NM PDU data has failed. NetworkHandle does not exist (development only) module not yet initialized (development only).
<b>Description</b> <p>This function gets the whole PDU data out of the most recently received NM message.</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>▶ The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>		

#### 5.5.2.4.10. Nm\_GetState

<b>Purpose</b> <p>Get the State of the network management module.</p>		
<b>Synopsis</b> <pre>Std_ReturnType <b>Nm_GetState</b> ( NetworkHandleType nmNetworkHandle , Nm_StateType * nmStatePtr , Nm_ModeType * nmModePtr );</pre>		



<b>Service ID</b>	14	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant for the same NetworkHandle. Reentrant otherwise	
<b>Parameters (in)</b>	nmNetworkHandle	Identification of the NM-channel.
	nmStatePtr	Pointer where state of the network management shall be copied to.
	nmModePtr	Pointer to the location where the mode of the network management shall be copied to.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Getting of NM state has failed. NetworkHandle does not exist (development only). Module not yet initialized (development only).
<b>Description</b>	<p>This function returns the state of the network management. This function in turn calls the BusNm_GetState function (e.g. CanNm_GetState function is called if channel is configured as CAN).</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>▶ The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>	

#### 5.5.2.4.11. Nm\_GetUserData

<b>Purpose</b>	Get user data out of the last received NM message.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>Nm_GetUserData</b> ( NetworkHandleType NetworkHandle , uint8 * nmUserDataPtr );</pre>	
<b>Service ID</b>	7	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant for the same NetworkHandle. Reentrant otherwise	
<b>Parameters (in)</b>	NetworkHandle	Channel Identification of the NM-channel.
<b>Parameters (out)</b>	nmUserDataPtr	Pointer to a memory area where the extracted data shall be written to.
<b>Return Value</b>	Standard Return Code	



	E_OK	No Error.
	E_NOT_OK	Getting of user data has failed. NetworkHandle does not exist (development only). Module not yet initialized (development only).
<b>Description</b>	<p>This function gets user data out of the last successfully received NM message. For that purpose BusNm_GetUserData shall be called (e.g. CanNm_GetUserData function is called if channel is configured as CAN).</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>▶ The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>	

#### 5.5.2.4.12. Nm\_Init

<b>Purpose</b>	Initializes the NM Interface.
<b>Synopsis</b>	<code>void Nm_Init ( void );</code>
<b>Service ID</b>	0
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	This function initializes the Nm Interface (Nm).

#### 5.5.2.4.13. Nm\_MainFunction

<b>Purpose</b>	This function is supplied for the NM coordinator functionality.
<b>Synopsis</b>	<code>void Nm_MainFunction ( void );</code>
<b>Service ID</b>	16
<b>Description</b>	<p>This function implements the processes of the NM Interface, which need a fix cyclic scheduling. This function is supplied for the NM coordinator functionality (Nm020). However, specific implementation may not need it (Nm093).</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>▶ NM Interface must be initialized before</li> </ul>



#### 5.5.2.4.14. Nm\_NetworkMode

<b>Purpose</b>	Notifies that the network management has entered Network Mode.	
<b>Synopsis</b>	<code>void Nm_NetworkMode ( NetworkHandleType nmNetworkHandle );</code>	
<b>Service ID</b>	30	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	nmNetworkHandle	Identification of the NM-channel.
<b>Description</b>	<p>This function gives a notification that the network management has entered Network Mode. The callback function enables transmission of application messages.</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>➤ The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>	

#### 5.5.2.4.15. Nm\_NetworkRelease

<b>Purpose</b>	Releases the Bus communication.					
<b>Synopsis</b>	<code>Std_ReturnType Nm_NetworkRelease ( NetworkHandleType NetworkHandle );</code>					
<b>Service ID</b>	3					
<b>Sync/Async</b>	Synchronous					
<b>Reentrancy</b>	Non-Reentrant for the same NetworkHandle. Reentrant otherwise					
<b>Parameters (in)</b>	NetworkHandle	Identification of the NM-channel.				
<b>Return Value</b>	<p>Standard Return Code</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">E_OK</td> <td style="padding: 2px;">No Error.</td> </tr> <tr> <td style="padding: 2px;">E_NOT_OK</td> <td style="padding: 2px;">Releasing of bus communication has failed. NetworkHandle does not exist (development only). The module is not yet initialized (development only).</td> </tr> </table>		E_OK	No Error.	E_NOT_OK	Releasing of bus communication has failed. NetworkHandle does not exist (development only). The module is not yet initialized (development only).
E_OK	No Error.					
E_NOT_OK	Releasing of bus communication has failed. NetworkHandle does not exist (development only). The module is not yet initialized (development only).					
<b>Description</b>	<p>This function calls the BusNm_NetworkRelease() bus specific function (e.g. CAN-Nm_NetworkRelease() function is called if channel is configured as CAN).</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>➤ The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>					



#### 5.5.2.4.16. Nm\_NetworkRequest

<b>Purpose</b>	Requests bus communication.	
<b>Synopsis</b>	Std_ReturnType <b>Nm_NetworkRequest</b> ( NetworkHandleType Net-workHandle );	
<b>Service ID</b>	2	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant for the same NetworkHandle. Reentrant otherwise	
<b>Parameters (in)</b>	NetworkHandle	NM channel for which the network shall be requested.
<b>Return Value</b>	Standard Return Code E_OK                          No Error. E_NOT_OK                      Requesting of bus communication has failed. NetworkHandle does not exist (development only). The module is not yet initialized (development only).	
<b>Description</b>	<p>This function calls the function BusNm_NetworkRequest() (e.g. CanNm_NetworkRequest() function is called if channel is configured as CAN).</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>► The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>	

#### 5.5.2.4.17. Nm\_NetworkStartIndication

<b>Purpose</b>	Notifies that Network Mode has been entered.	
<b>Synopsis</b>	void <b>Nm_NetworkStartIndication</b> ( NetworkHandleType nmNet-workHandle );	
<b>Service ID</b>	33	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	nmNetworkHandle	Identification of the NM-channel.
<b>Description</b>	This function provides a notification that a NM-message has been received in the Bus-Sleep Mode, which indicates that some nodes in the network have already en-	



tered the Network Mode. The callback function starts the network management state machine.

Preconditions:

- The channel handle should be valid and the module should have been initialized for this channel (checked).

#### 5.5.2.4.18. Nm\_PassiveStartUp

<b>Purpose</b>	Passive start of Network Management.	
<b>Synopsis</b>	Std_ReturnType <b>Nm_PassiveStartUp</b> ( NetworkHandleType Net-workHandle );	
<b>Service ID</b>	1	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant for the same NetworkHandle. Reentrant otherwise	
<b>Parameters (in)</b>	NetworkHandle	Identification of the NM-channel.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Passive start of network management has failed. NetworkHandle does not exist (development only) module not yet initialized (development only).
<b>Description</b>	<p>This function calls the BusNm_PassiveStartUp function (e.g. CanNm_PassiveStartUp function is called if channel is configured as CAN).</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>➤ The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>	

#### 5.5.2.4.19. Nm\_PreparesBusSleepMode

<b>Purpose</b>	Notification that the network management has entered Prepare Bus-Sleep Mode.
<b>Synopsis</b>	void <b>Nm_PreparesBusSleepMode</b> ( NetworkHandleType nmNetworkHandle );
<b>Service ID</b>	31



<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	nmNetworkHandle	Identification of the NM-channel.
<b>Description</b>	<p>This function provides a notification that the network management has entered Prepare Bus-Sleep Mode. The callback function disables transmission of application messages.</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>▶ The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>	

#### 5.5.2.4.20. Nm\_RemoteSleepCancellation

<b>Purpose</b>	Notification that not all other nodes are ready to sleep.	
<b>Synopsis</b>	<pre>void <b>Nm_RemoteSleepCancellation</b> ( NetworkHandleType nmNetworkHandle );</pre>	
<b>Parameters (in)</b>	nmNetworkHandle	Identification of the NM-channel
<b>Description</b>	<p>Notification that the network management has detected that not all other nodes on the network are longer ready to enter Bus-Sleep Mode.</p> <p>If the Nm coordinator support is disabled this callback function is implemented as function-like macro.</p> <p>This function is not specified in AUTOSAR R3.x Nm SWS but in the R4.0 draft SWS.</p>	

#### 5.5.2.4.21. Nm\_RemoteSleepIndication

<b>Purpose</b>	Notification that all other nodes are ready to sleep.	
<b>Synopsis</b>	<pre>void <b>Nm_RemoteSleepIndication</b> ( NetworkHandleType nmNetworkHandle );</pre>	
<b>Parameters (in)</b>	nmNetworkHandle	Identification of the NM-channel
<b>Description</b>	<p>This function provides a notification that the network management has detected that all other nodes are ready to sleep. The NM gateway checks if the Bus is still required.</p> <p>If the Nm coordinator support is disabled this API function is implemented as empty function-like macro, as it provides no functionality.</p>	



#### 5.5.2.4.22. Nm\_RepeatMessageRequest

<b>Purpose</b>	Set Repeat Message Request bit.	
<b>Synopsis</b>	Std_ReturnType <b>Nm_RepeatMessageRequest</b> ( NetworkHandleType NetworkHandle );	
<b>Service ID</b>	9	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant for the same NetworkHandle. Reentrant otherwise	
<b>Parameters (in)</b>	NetworkHandle	Identification of the NM-channel.
<b>Return Value</b>	Standard Return Code	
	E_OK	No error.
	E_NOT_OK	Setting of Repeat Message Request Bit has failed. NetworkHandle does not exist (development only). Module not yet initialized (development only).
<b>Description</b>	<p>This function sets the Repeat Message Request Bit for NM messages transmitted next on the bus. For that purpose BusNm_RepeatMessageRequest() shall be called (e.g. CanNm_RepeatMessageRequest() function is called if channel is configured as CAN)</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>➤ The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>	

#### 5.5.2.4.23. Nm\_SetUserData

<b>Purpose</b>	Set user data.	
<b>Synopsis</b>	Std_ReturnType <b>Nm_SetUserData</b> ( NetworkHandleType NetworkHandle , const uint8 * nmUserDataPtr );	
<b>Service ID</b>	6	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant for the same NetworkHandle. Reentrant otherwise	
<b>Parameters (in)</b>	NetworkHandle	Identification of the NM-channel.



	nmUserDataPtr	User data for the next transmitted NM message.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Setting of user data has failed. NetworkHandle does not exist (development only). The module is not yet initialized (development only).
<b>Description</b>	<p>This function sets the user data for NM messages transmitted next on the bus. For that purpose BusNm_SetUserData shall be called (e.g. CanNm_SetUserData function is called if channel is configured as CAN).</p> <p>Preconditions:</p> <ul style="list-style-type: none"> <li>➤ The channel handle should be valid and the module should have been initialized for this channel (checked).</li> </ul>	

#### 5.5.2.4.24. Nm\_SynchronizationPoint

<b>Purpose</b>	Notification that this is a suitable point in time to initiate the coordination algorithm on.	
<b>Synopsis</b>	<pre>void <b>Nm_SynchronizationPoint</b> ( NetworkHandleType nmNetworkHandle );</pre>	
<b>Parameters (in)</b>	nmNetworkHandle	Identification of the NM-channel
<b>Description</b>	<p>Notification NM Coordinator functionality that this is a suitable point in time to initiate the coordination algorithm on</p> <p>If the Nm coordinator support is disabled this callback function is implemented as function-like macro.</p> <p>This function is not specified in AUTOSAR R3.x Nm SWS but in the R4.0 draft SWS.</p>	

#### 5.5.2.4.25. Nm\_TxTimeoutException

<b>Purpose</b>	Notification service to indicate that an attempt to send a NM message failed.	
<b>Synopsis</b>	<pre>void <b>Nm_TxTimeoutException</b> ( NetworkHandleType nmNetworkHandle );</pre>	
<b>Parameters (in)</b>	nmNetworkHandle	Identification of the NM-channel



<b>Description</b>	This function provides an indication that an attempt to send an NM message failed.
--------------------	--

## 5.5.3. Integration notes

### 5.5.3.1. Exclusive areas

This section describes the exclusive areas used by the `Nm` module.

#### 5.5.3.1.1. SCHM\_NM\_EXCLUSIVE\_AREA\_0

<b>Protected data structures</b>	All shared data that shall be protected from mutual access.
<b>Recommended locking mechanism</b>	This exclusive area must always be protected by a locking mechanism. The options for locking are described in the EB tresos AutoCore Generic documentation. Refer to the section <a href="#">Mapping exclusive areas in the basic software modules</a> in the <a href="#">Integration notes</a> section for details.

### 5.5.3.2. Production errors

Production errors are not reported by the `Nm` module.

### 5.5.3.3. Memory mapping

General information about memory mapping is provided in the EB tresos AutoCore Generic documentation. Refer to the section [Memory mapping and compiler abstraction](#) in the [Integration notes](#) section for details.

The following table provides the list of sections that may be mapped for this module:

<b>Memory section</b>
CODE
CONST_UNSPECIFIED
CONST_8



VAR\_CLEARED\_UNSPECIFIED

VAR\_CLEARED\_16

VAR\_INIT\_8

VAR\_CLEARED\_8

#### 5.5.3.4. Integration requirements

---

**WARNING**



**Integration requirements list is not exhaustive**

The following list of integration requirements helps you to integrate your product. However, this list is not exhaustive. You also require information from the user guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

---

Integration requirements are not listed for the Nm module.