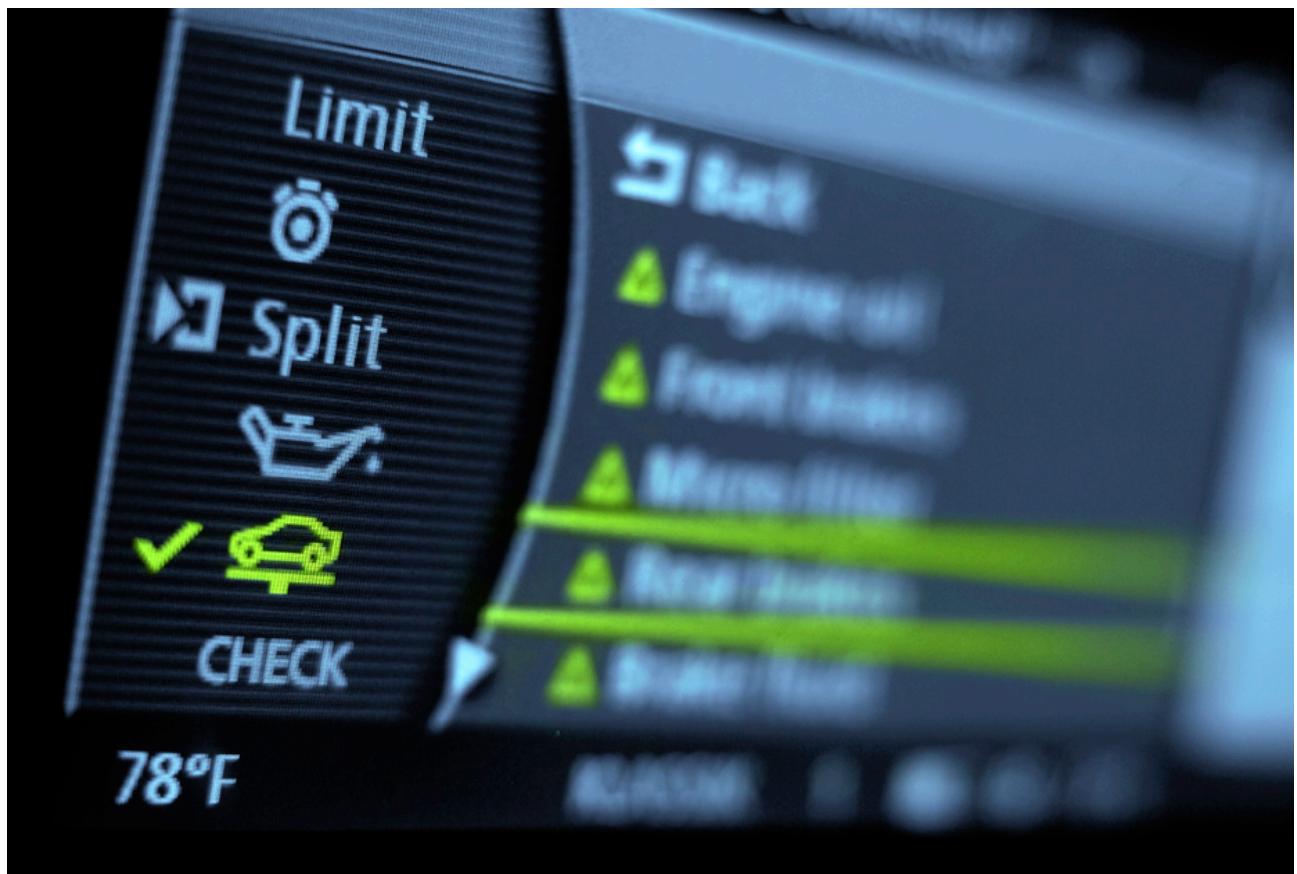




Elektrobit

# EB tresos<sup>®</sup> AutoCore Generic 8 IP Stack 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 IP Stack documentation .....	31
2. Supported features .....	32
3. ACG8 IP Stack release notes .....	39
3.1. Overview .....	39
3.2. Scope of the release .....	39
3.2.1. Configuration tool .....	39
3.2.2. AUTOSAR modules .....	39
3.2.3. EB (Elektrobit) modules .....	40
3.2.4. MCAL modules and EB tresos AutoCore OS .....	40
3.3. Module release notes .....	40
3.3.1. DolP module release notes .....	40
3.3.1.1. Change log .....	41
3.3.1.2. New features .....	49
3.3.1.3. EB-specific enhancements .....	50
3.3.1.4. Deviations .....	54
3.3.1.5. Limitations .....	60
3.3.1.6. Open-source software .....	62
3.3.2. Ethlf module release notes .....	62
3.3.2.1. Change log .....	62
3.3.2.2. New features .....	69
3.3.2.3. EB-specific enhancements .....	70
3.3.2.4. Deviations .....	72
3.3.2.5. Limitations .....	78
3.3.2.6. Open-source software .....	80
3.3.3. EthSM module release notes .....	80
3.3.3.1. Change log .....	80
3.3.3.2. New features .....	85
3.3.3.3. EB-specific enhancements .....	85
3.3.3.4. Deviations .....	86
3.3.3.5. Limitations .....	88
3.3.3.6. Open-source software .....	88
3.3.4. Sd module release notes .....	88
3.3.4.1. Change log .....	88
3.3.4.2. New features .....	95
3.3.4.3. EB-specific enhancements .....	95
3.3.4.4. Deviations .....	96
3.3.4.5. Limitations .....	98
3.3.4.6. Open-source software .....	99
3.3.5. SoAd module release notes .....	100



3.3.5.1. Change log .....	100
3.3.5.2. New features .....	113
3.3.5.3. EB-specific enhancements .....	114
3.3.5.4. Deviations .....	115
3.3.5.5. Limitations .....	122
3.3.5.6. Open-source software .....	123
3.3.6. SomelpTp module release notes .....	123
3.3.6.1. Change log .....	123
3.3.6.2. New features .....	128
3.3.6.3. EB-specific enhancements .....	128
3.3.6.4. Deviations .....	129
3.3.6.5. Limitations .....	133
3.3.6.6. Open-source software .....	133
3.3.7. Tcplp module release notes .....	133
3.3.7.1. Change log .....	133
3.3.7.2. New features .....	145
3.3.7.3. EB-specific enhancements .....	146
3.3.7.4. Deviations .....	148
3.3.7.5. Limitations .....	175
3.3.7.6. Open-source software .....	180
3.3.8. UdpNm module release notes .....	180
3.3.8.1. Change log .....	180
3.3.8.2. New features .....	186
3.3.8.3. EB-specific enhancements .....	186
3.3.8.4. Deviations .....	188
3.3.8.5. Limitations .....	196
3.3.8.6. Open-source software .....	196
4. ACG8 IP Stack user guide .....	197
4.1. Overview .....	197
4.2. Background information .....	197
4.2.1. Modules of the AUTOSAR IP Stack .....	197
4.2.2. IP stack dependencies .....	199
4.2.3. Network management in AUTOSAR IP stack .....	200
4.3. Configuring the ACG8 IP Stack .....	200
4.3.1. Multiple provided service instances .....	200
4.3.1.1. Configuring the SoAd module .....	201
4.3.1.2. Configuring the Sd module .....	201
4.3.2. Debug support .....	202
4.3.2.1. GetAndResetMeasurementData service .....	202
4.4. DolP module user guide .....	202
4.4.1. Overview .....	202
4.4.2. Background Information .....	203



4.4.3. Configuring DolP .....	204
4.5. Sd module user guide .....	204
4.5.1. Overview .....	204
4.5.2. Background information .....	204
4.5.3. Configuring the Sd module .....	205
4.6. SoAd module user guide .....	206
4.6.1. Overview .....	206
4.6.2. Background information .....	206
4.6.2.1. N-PDU buffer pooling .....	206
4.6.2.1.1. Temporary unavailable buffer .....	206
4.6.2.1.2. Debug hooks .....	207
4.6.3. Configuring the SoAd module .....	207
4.6.3.1. Configuring the N-PDU buffer pooling .....	207
4.7. Tcplp module user guide .....	208
4.7.1. Overview .....	208
4.7.2. Background Information .....	208
4.7.2.1. Software hardening support .....	208
4.7.2.1.1. TCP SYN cookies .....	209
4.7.2.1.2. Unpredictable sequence numbers .....	209
4.7.2.2. Storage of the IP address to NVM RAM .....	209
4.7.2.3. IPv4 Duplicate Address Detection (ARP probing) .....	209
4.7.2.4. IPv6 Duplicate Address Detection .....	210
4.7.2.5. Debug support in Tcplp .....	210
4.7.3. Configuring the TcpIp module .....	211
4.7.3.1. Configuring the software hardening features .....	211
4.7.3.2. Configuring the permanent IP address storage to NVM RAM .....	215
4.7.3.3. Configuring the IPv4 Duplicate Address Detection (ARP probing) .....	217
4.7.3.4. Configuring the IPv6 Duplicate Address Detection .....	219
4.8. QoS Support user guide .....	220
4.8.1. Overview .....	220
4.8.2. Background information .....	221
4.8.3. Configuring QoS Support .....	223
4.8.3.1. Configuring the Ethernet Driver Module (Eth) .....	223
4.8.3.2. Configuring the Ethernet Interface Module (EthIf) .....	224
4.8.3.3. Configuring the TCP/IP Module (TcpIp) .....	224
4.8.3.4. Configuring the Socket Adaptor Module (SoAd) .....	225
4.8.3.5. Configuring the Operating System Module (Os) .....	225
4.8.3.6. Configuring the Run-Time Environment Module (Rte) .....	226
5. ACG8 IP Stack module references .....	227
5.1. Overview .....	227
5.1.1. Notation in EB module references .....	227
5.1.1.1. Default value of configuration parameters .....	227



5.1.1.2. Range information of configuration parameters .....	227
5.2. DolP .....	228
5.2.1. Configuration parameters .....	228
5.2.1.1. CommonPublishedInformation .....	229
5.2.1.2. DolPConfigSet .....	232
5.2.1.3. DolPChannel .....	233
5.2.1.4. DolPPduRRxPdu .....	234
5.2.1.5. DolPPduRTxPdu .....	235
5.2.1.6. DolPCustomChannel .....	236
5.2.1.7. DolPPduRRxPdu .....	236
5.2.1.8. DolPPduRTxPdu .....	237
5.2.1.9. DolPConnections .....	238
5.2.1.10. DolPTargetAddress .....	238
5.2.1.11. DolPTcpConnection .....	239
5.2.1.12. DolPSoAdRxPdu .....	240
5.2.1.13. DolPSoAdTxPdu .....	240
5.2.1.14. DolPUdpConnection .....	241
5.2.1.15. DolPSoAdRxPdu .....	242
5.2.1.16. DolPSoAdTxPdu .....	243
5.2.1.17. DolPUdpVehicleAnnouncement .....	243
5.2.1.18. DolPSoAdTxPdu .....	244
5.2.1.19. DolPRoutingActivation .....	245
5.2.1.20. DolPRoutingActivationAuthenticationCallback .....	247
5.2.1.21. DolPRoutingActivationConfirmationCallback .....	248
5.2.1.22. DolPTester .....	249
5.2.1.23. DolPGeneral .....	250
5.2.1.24. DolPGetGid .....	263
5.2.1.25. DolPPowerModeCallback .....	264
5.2.1.26. DolPTTriggerGidSynchronization .....	264
5.2.1.27. DolPDynamicGIDMasterSelectionCallback .....	265
5.2.1.28. DolPVIDRequestReceivedCallback .....	265
5.2.1.29. DolPGetVIN .....	266
5.2.1.30. DolPTcpSoConModeChgCallback .....	266
5.2.1.31. DolPDefensiveProgramming .....	267
5.2.1.32. PublishedInformation .....	270
5.2.2. Application programming interface (API) .....	270
5.2.2.1. Macro constants .....	270
5.2.2.1.1. DOIP_E_INVALID_PARAMETER .....	270
5.2.2.1.2. DOIP_E_INVALID_PDU_SDU_ID .....	270
5.2.2.1.3. DOIP_E_PARAM_POINTER .....	271
5.2.2.1.4. DOIP_E_UNINIT .....	271
5.2.2.1.5. DOIP_INSTANCE_ID .....	271



5.2.2.1.6. DOIP_SID_ACTIVATIONLINESWITCH .....	271
5.2.2.1.7. DOIP_SID_CANCELRECEIVE .....	271
5.2.2.1.8. DOIP_SID_CANCELTRANSMIT .....	272
5.2.2.1.9. DOIP_SID_GETANDRESETMEASUREMENTDATA .....	272
5.2.2.1.10. DOIP_SID_GETVERSIONINFO .....	272
5.2.2.1.11. DOIP_SID_IFTRANSMIT .....	272
5.2.2.1.12. DOIP_SID_INIT .....	272
5.2.2.1.13. DOIP_SID_LOCALIPADDRASSIGNMENTCHG .....	272
5.2.2.1.14. DOIP_SID_MAINFUNCTION .....	273
5.2.2.1.15. DOIP_SID_MAINFUNCTION_TX .....	273
5.2.2.1.16. DOIP_SID_SOADIFRXINDICATION .....	273
5.2.2.1.17. DOIP_SID_SOADIFTXCONFIRMATION .....	273
5.2.2.1.18. DOIP_SID_SOADTPCOPYRXDATA .....	273
5.2.2.1.19. DOIP_SID_SOADTPCOPYTXDATA .....	274
5.2.2.1.20. DOIP_SID_SOADTPRXINDICATION .....	274
5.2.2.1.21. DOIP_SID_SOADTPSTARTOFRECEPTION .....	274
5.2.2.1.22. DOIP_SID_SOADTPTXCONFIRMATION .....	274
5.2.2.1.23. DOIP_SID_SOCONMODECHG .....	274
5.2.2.1.24. DOIP_SID_TPTRANSMIT .....	274
5.2.2.2. Functions .....	275
5.2.2.2.1. DoIP_ActivationLineSwitchActive .....	275
5.2.2.2.2. DoIP_ActivationLineSwitchInactive .....	275
5.2.2.2.3. DoIP_CancelReceive .....	275
5.2.2.2.4. DoIP_CancelTransmit .....	276
5.2.2.2.5. DoIP_GetAndResetMeasurementData .....	276
5.2.2.2.6. DoIP_GetVersionInfo .....	277
5.2.2.2.7. DoIP_Init .....	277
5.2.2.2.8. DoIP_IsValidConfig .....	278
5.2.2.2.9. DoIP_LocalIpAddrAssignmentChg .....	278
5.2.2.2.10. DoIP_MainFunction .....	278
5.2.2.2.11. DoIP_MainFunctionTx .....	279
5.2.2.2.12. DoIP_SoAdIfRxIndication .....	279
5.2.2.2.13. DoIP_SoAdIfTxConfirmation .....	279
5.2.2.2.14. DoIP_SoAdTpCopyRxData .....	279
5.2.2.2.15. DoIP_SoAdTpCopyTxData .....	280
5.2.2.2.16. DoIP_SoAdTpRxIndication .....	281
5.2.2.2.17. DoIP_SoAdTpStartOfReception .....	281
5.2.2.2.18. DoIP_SoAdTpTxConfirmation .....	282
5.2.2.2.19. DoIP_SoConModeChg .....	282
5.2.2.2.20. DoIP_TpTransmit .....	282
5.2.3. Integration notes .....	283
5.2.3.1. Exclusive areas .....	283



5.2.3.1.1. SCHM_DOIP_EXCLUSIVE_AREA_0 .....	283
5.2.3.2. Production errors .....	283
5.2.3.3. Memory mapping .....	283
5.2.3.4. Integration requirements .....	284
5.2.3.4.1. DoIP.EB_INTREQ_DoIP_0001 .....	284
5.2.3.4.2. DoIP.EB_INTREQ_DoIP_0002 .....	284
5.2.3.4.3. DoIP.EB_INTREQ_DoIP_0003 .....	285
5.2.3.4.4. DoIP.EB_INTREQ_DoIP_0004 .....	285
5.2.3.4.5. DoIP.EB_INTREQ_DoIP_0005 .....	285
5.2.3.4.6. DoIP.EB_INTREQ_DoIP_0006 .....	286
5.2.3.4.7. DoIP.EB_INTREQ_DoIP_0007 .....	286
5.3. EthIf .....	286
5.3.1. Configuration parameters .....	286
5.3.1.1. CommonPublishedInformation .....	287
5.3.1.2. EthIfConfigSet .....	290
5.3.1.3. EthIfController .....	291
5.3.1.4. EthIfFrameOwnerConfig .....	294
5.3.1.5. EthIfPhysController .....	295
5.3.1.6. EthIfPhysCtrlRxMainFunctionPriorityProcessing .....	296
5.3.1.7. EthIfRxIndicationConfig .....	297
5.3.1.8. EthIfSwitch .....	298
5.3.1.9. EthIfSwitchMgmtInfoIndicationConfig .....	298
5.3.1.10. EthIfSwitchPortGroup .....	299
5.3.1.11. EthIfSwitchTimeStampIndicationConfig .....	300
5.3.1.12. EthIfTransceiver .....	300
5.3.1.13. EthIfTrcvLinkStateChgConfig .....	301
5.3.1.14. EthIfTxConfirmationConfig .....	302
5.3.1.15. EthIfEthControllerType .....	302
5.3.1.16. EthIfEthTrcvType .....	303
5.3.1.17. EthIfEthSwtType .....	303
5.3.1.18. EthIfGeneral .....	304
5.3.1.19. ReportToDem .....	322
5.3.1.20. EthIfDefensiveProgramming .....	324
5.3.1.21. PublishedInformation .....	327
5.3.2. Recommended configurations .....	327
5.3.2.1. EthIfRecConfiguration .....	327
5.3.2.1.1. EthIfConfigSet .....	327
5.3.2.1.2. EthIfController_0 .....	328
5.3.2.1.3. EthIfFrameOwnerConfig_IPv4 .....	328
5.3.2.1.4. EthIfFrameOwnerConfig_IPv6 .....	328
5.3.2.1.5. EthIfFrameOwnerConfig_ARP .....	329
5.3.2.1.6. EthIfPhysController_0 .....	329



5.3.2.1.7. EthIfRxIndicationConfig_IPv4 .....	329
5.3.2.1.8. EthIfRxIndicationConfig_IPv6 .....	329
5.3.2.1.9. EthIfRxIndicationConfig_ARP .....	329
5.3.2.1.10. EthIfTrcvLinkStateChgConfig_EthSM .....	330
5.3.2.1.11. EthIfTxConfirmationConfig_0 .....	330
5.3.2.2. EthIfRecConfigurationEthTSyn .....	330
5.3.2.2.1. EthIfConfigSet .....	330
5.3.2.2.2. EthIfController_0 .....	331
5.3.2.2.3. EthIfFrameOwnerConfig_PTP .....	331
5.3.2.2.4. EthIfFrameOwnerConfig_IPv4 .....	331
5.3.2.2.5. EthIfFrameOwnerConfig_IPv6 .....	332
5.3.2.2.6. EthIfFrameOwnerConfig_ARP .....	332
5.3.2.2.7. EthIfPhysController_0 .....	332
5.3.2.2.8. EthIfRxIndicationConfig_PTP .....	332
5.3.2.2.9. EthIfRxIndicationConfig_IPv4 .....	332
5.3.2.2.10. EthIfRxIndicationConfig_IPv6 .....	333
5.3.2.2.11. EthIfRxIndicationConfig_ARP .....	333
5.3.2.2.12. EthIfTrcvLinkStateChgConfig_EthSM .....	333
5.3.2.2.13. EthIfTrcvLinkStateChgConfig_EthTSyn .....	333
5.3.2.2.14. EthIfTxConfirmationConfig_PTP .....	333
5.3.2.2.15. EthIfGeneral .....	334
5.3.3. Application programming interface (API) .....	334
5.3.3.1. Macro constants .....	334
5.3.3.1.1. ETHIF_CBK_RXINDICATION_SVCID .....	334
5.3.3.1.2. ETHIF_CBK_TXCONFIRMATION_SVCID .....	334
5.3.3.1.3. ETHIF_CHECKWAKEUP_SVCID .....	334
5.3.3.1.4. ETHIF_CLEARSWITCHPORTSIGNALQUALITY_SVCID .....	334
5.3.3.1.5. ETHIF_CLEARTRCVSIGNALQUALITY_SVCID .....	334
5.3.3.1.6. ETHIF_CTRLMODEINDICATION_SVCID .....	335
5.3.3.1.7. ETHIF_DISABLERELATEDETHIFCTRLS_SVCID .....	335
5.3.3.1.8. ETHIF_ENABLEEGRESSTIMESTAMP_SVCID .....	335
5.3.3.1.9. ETHIF_ENABLERELATEDETHIFCTRLS_SVCID .....	335
5.3.3.1.10. ETHIF_E_HW_NOT_INITIALIZED .....	335
5.3.3.1.11. ETHIF_E_INIT_FAILED .....	335
5.3.3.1.12. ETHIF_E_INV_CTRL_IDX .....	336
5.3.3.1.13. ETHIF_E_INV_PARAM .....	336
5.3.3.1.14. ETHIF_E_INV_PORT_GROUP_IDX .....	336
5.3.3.1.15. ETHIF_E_INV_SWT_IDX .....	336
5.3.3.1.16. ETHIF_E_INV_TRCV_IDX .....	336
5.3.3.1.17. ETHIF_E_NOT_INITIALIZED .....	336
5.3.3.1.18. ETHIF_E_PARAM_POINTER .....	336
5.3.3.1.19. ETHIF_GETANDRESETMEASUREMENTDATA_SVCID .....	337



5.3.3.1.20. ETHIF_GETARLTABLE_SVCID .....	337
5.3.3.1.21. ETHIF_GETBUFFERLEVEL_SVCID .....	337
5.3.3.1.22. ETHIF_GETCABLEDIAGNOSTICSRESULT_SVCID .....	337
5.3.3.1.23. ETHIF_GETCONTROLLERMODE_SVCID .....	337
5.3.3.1.24. ETHIF_GETCTRLIDXLIST_SVCID .....	337
5.3.3.1.25. ETHIF_GETCURRENTTIME_SVCID .....	338
5.3.3.1.26. ETHIF_GETGRESSTIMESTAMP_SVCID .....	338
5.3.3.1.27. ETHIF_GETINGRESSTIMESTAMP_SVCID .....	338
5.3.3.1.28. ETHIF_GETPHYIDENTIFIER_SVCID .....	338
5.3.3.1.29. ETHIF_GETPHYSADDR_SVCID .....	338
5.3.3.1.30. ETHIF_GETPORTMACADDR_SVCID .....	338
5.3.3.1.31. ETHIF_GETSWITCHPORTSIGNALQUALITY_SVCID .....	338
5.3.3.1.32. ETHIF_GETTRANSCEIVERWAKEUPMODE_SVCID .....	339
5.3.3.1.33. ETHIF_GETTRCVSIGNALQUALITY_SVCID .....	339
5.3.3.1.34. ETHIF_GETVERSIONINFO_SVCID .....	339
5.3.3.1.35. ETHIF_GETVLANID_SVCID .....	339
5.3.3.1.36. ETHIF_INIT_SVCID .....	339
5.3.3.1.37. ETHIF_INSTANCE_ID .....	339
5.3.3.1.38. ETHIF_MAINFUNCTIONRX_SVCID .....	340
5.3.3.1.39. ETHIF_MAINFUNCTIONSTATE_SVCID .....	340
5.3.3.1.40. ETHIF_MAINFUNCTIONTX_SVCID .....	340
5.3.3.1.41. ETHIF_PROVIDETXBUFFER_SVCID .....	340
5.3.3.1.42. ETHIF_READMII_SVCID .....	340
5.3.3.1.43. ETHIF_RESETCONFIGURATION_SVCID .....	340
5.3.3.1.44. ETHIF_RETRANSMIT_SVCID .....	340
5.3.3.1.45. ETHIF_SETCONTROLLERMODE_SVCID .....	341
5.3.3.1.46. ETHIF_SETCORRECTIONTIME_SVCID .....	341
5.3.3.1.47. ETHIF_SETFORWARDINGMODE_SVCID .....	341
5.3.3.1.48. ETHIF_SETGLOBALTIME_SVCID .....	341
5.3.3.1.49. ETHIF_SETPHYLOOPBACKMODE_SVCID .....	341
5.3.3.1.50. ETHIF_SETPHYSADDR_SVCID .....	341
5.3.3.1.51. ETHIF_SETPHYTESTMODE_SVCID .....	342
5.3.3.1.52. ETHIF_SETPHYTXMODE_SVCID .....	342
5.3.3.1.53. ETHIF_SETTRANSCEIVERWAKEUPMODE_SVCID .....	342
5.3.3.1.54. ETHIF_STARTALLPORTS_SVCID .....	342
5.3.3.1.55. ETHIF_STORECONFIGURATION_SVCID .....	342
5.3.3.1.56. ETHIF_SWITCHMGMTINFOINDICATION_SVCID .....	342
5.3.3.1.57. ETHIF_SWITCHPORTGROUPREQUESTMODE_SVCID .....	342
5.3.3.1.58. ETHIF_SWTGETCOUNTERVALUES_SVCID .....	343
5.3.3.1.59. ETHIF_TRANSMIT_SVCID .....	343
5.3.3.1.60. ETHIF_TRCVMODEINDICATION_SVCID .....	343
5.3.3.1.61. ETHIF_UPDATEPHYSADDRFILTER_SVCID .....	343



5.3.3.1.62. ETHIF_VERIFYCONFIG_SVCID .....	343
5.3.3.1.63. ETHIF_WRITEMII_SVCID .....	343
5.3.3.1.64. TS_RELOCATABLE_CFG_ENABLE .....	343
5.3.3.2. Functions .....	344
5.3.3.2.1. EthIf_Cbk_RxIndication .....	344
5.3.3.2.2. EthIf_Cbk_TxConfirmation .....	344
5.3.3.2.3. EthIf_CheckWakeUp .....	345
5.3.3.2.4. EthIf_ClearSwitchPortSignalQuality .....	345
5.3.3.2.5. EthIf_CtrlModeIndication .....	345
5.3.3.2.6. EthIf_DisableRelatedEthIfCtrls .....	346
5.3.3.2.7. EthIf_EnableEgressTimeStamp .....	346
5.3.3.2.8. EthIf_EnableRelatedEthIfCtrls .....	347
5.3.3.2.9. EthIf_GetAndResetMeasurementData .....	347
5.3.3.2.10. EthIf_GetArlTable .....	348
5.3.3.2.11. EthIf_GetBufferLevel .....	348
5.3.3.2.12. EthIf_GetControllerMode .....	349
5.3.3.2.13. EthIf_GetCtrlIdxList .....	349
5.3.3.2.14. EthIf_GetCurrentTime .....	350
5.3.3.2.15. EthIf_GetEgressTimeStamp .....	350
5.3.3.2.16. EthIf_GetIngressTimeStamp .....	351
5.3.3.2.17. EthIf_GetPhysAddr .....	351
5.3.3.2.18. EthIf_GetPortMacAddr .....	352
5.3.3.2.19. EthIf_GetSwitchPortSignalQuality .....	352
5.3.3.2.20. EthIf_GetTransceiverWakeupMode .....	353
5.3.3.2.21. EthIf_GetVersionInfo .....	353
5.3.3.2.22. EthIf_GetVlanId .....	354
5.3.3.2.23. EthIf_Init .....	354
5.3.3.2.24. EthIf_IsValidConfig .....	354
5.3.3.2.25. EthIf_MainFunctionRx .....	355
5.3.3.2.26. EthIf_MainFunctionRxPrio .....	355
5.3.3.2.27. EthIf_MainFunctionState .....	356
5.3.3.2.28. EthIf_MainFunctionTx .....	356
5.3.3.2.29. EthIf_ProvideTxBuffer .....	356
5.3.3.2.30. EthIf_ReadMii .....	357
5.3.3.2.31. EthIf_ResetConfiguration .....	357
5.3.3.2.32. EthIf_Retransmit .....	358
5.3.3.2.33. EthIf_RxIndication .....	359
5.3.3.2.34. EthIf_SetControllerMode .....	359
5.3.3.2.35. EthIf_SetCorrectionTime .....	360
5.3.3.2.36. EthIf_SetForwardingMode .....	360
5.3.3.2.37. EthIf_SetGlobalTime .....	360
5.3.3.2.38. EthIf_SetPhysAddr .....	361



5.3.3.2.39. EthIf_SetTransceiverWakeUpMode .....	361
5.3.3.2.40. EthIf_StartAllPorts .....	362
5.3.3.2.41. EthIf_StoreConfiguration .....	362
5.3.3.2.42. EthIf_SwitchMgmtInfoIndication .....	362
5.3.3.2.43. EthIf_SwitchPortGroupRequestMode .....	363
5.3.3.2.44. EthIf_SwtGetCounterValues .....	363
5.3.3.2.45. EthIf_Transmit .....	364
5.3.3.2.46. EthIf_TrccModelIndication .....	364
5.3.3.2.47. EthIf_TxConfirmation .....	365
5.3.3.2.48. EthIf_UpdatePhysAddrFilter .....	365
5.3.3.2.49. EthIf_VerifyConfig .....	366
5.3.3.2.50. EthIf_WriteMii .....	366
5.3.4. Integration notes .....	367
5.3.4.1. Exclusive areas .....	367
5.3.4.1.1. SCHM_ETHIF_EXCLUSIVE_AREA .....	367
5.3.4.2. Production errors .....	367
5.3.4.3. Memory mapping .....	367
5.3.4.4. Integration requirements .....	368
5.3.4.4.1. lim.EthIf.EB_INTREQ_EthIf_0001 .....	368
5.3.4.4.2. lim.EthIf.EB_INTREQ_EthIf_0002 .....	368
5.3.4.4.3. lim.EthIf.EB_INTREQ_EthIf_0003 .....	369
5.3.4.4.4. lim.EthIf.EB_INTREQ_EthIf_0004 .....	369
5.3.4.4.5. lim.EthIf.EB_INTREQ_EthIf_0005 .....	369
5.3.4.4.6. lim.EthIf.EB_INTREQ_EthIf_0006 .....	369
5.3.4.4.7. lim.EthIf.EB_INTREQ_EthIf_0007 .....	369
5.3.4.4.8. lim.EthIf.EB_INTREQ_EthIf_0008 .....	370
5.3.4.4.9. lim.EthIf.EB_INTREQ_EthIf_0009 .....	370
5.3.4.4.10. lim.EthIf.EB_INTREQ_EthIf_0010 .....	370
5.3.4.4.11. lim.EthIf.EB_INTREQ_EthIf_0011 .....	370
5.3.4.4.12. lim.EthIf.EB_INTREQ_EthIf_0012 .....	371
5.3.4.4.13. lim.EthIf.EB_INTREQ_EthIf_0013 .....	371
5.4. EthSM .....	371
5.4.1. Configuration parameters .....	371
5.4.1.1. CommonPublishedInformation .....	372
5.4.1.2. EthSMConfig .....	375
5.4.1.3. EthSMGeneral .....	375
5.4.1.4. ReportToDem .....	378
5.4.1.5. EthSMDefensiveProgramming .....	380
5.4.1.6. EthSMNetwork .....	383
5.4.1.7. EthSMDemEventParameterRefs .....	384
5.4.1.8. PublishedInformation .....	385
5.4.2. Application programming interface (API) .....	385



5.4.2.1. Macro constants .....	385
5.4.2.1.1. ETHSM_CTRLMODEINDICATION_SVCID .....	385
5.4.2.1.2. ETHSM_DEVAUTHMODEINDICATION_SVCID .....	385
5.4.2.1.3. ETHSM_E_BUSSMMODEIND .....	386
5.4.2.1.4. ETHSM_E_INVALID_NETWORK_HANDLE .....	386
5.4.2.1.5. ETHSM_E_INVALID_NETWORK_MODE .....	386
5.4.2.1.6. ETHSM_E_INVALID_TRCV_LINK_STATE .....	386
5.4.2.1.7. ETHSM_E_INVALID_TcplpMode .....	386
5.4.2.1.8. ETHSM_E_PARAM_CONTROLLER .....	386
5.4.2.1.9. ETHSM_E_PARAM_POINTER .....	386
5.4.2.1.10. ETHSM_E_PARAM_TRANSCEIVER .....	387
5.4.2.1.11. ETHSM_E_UNINIT .....	387
5.4.2.1.12. ETHSM_GETCURRENTCOMMODE_SVCID .....	387
5.4.2.1.13. ETHSM_GETCURRENTINTERNALMODE_SVCID .....	387
5.4.2.1.14. ETHSM_GETVERSIONINFO_SVCID .....	387
5.4.2.1.15. ETHSM_INIT_SVCID .....	387
5.4.2.1.16. ETHSM_INSTANCE_ID .....	387
5.4.2.1.17. ETHSM_MAINFUNCTION_SVCID .....	388
5.4.2.1.18. ETHSM_REQUESTCOMMODE_SVCID .....	388
5.4.2.1.19. ETHSM_TCPIPMODEINDICATION_SVCID .....	388
5.4.2.1.20. ETHSM_TRCVLINKSTATECHG_SVCID .....	388
5.4.2.1.21. ETHSM_TRCVMODEINDICATION_SVCID .....	388
5.4.2.2. Functions .....	388
5.4.2.2.1. EthSM_GetCurrentComMode .....	388
5.4.2.2.2. EthSM_GetVersionInfo .....	389
5.4.2.2.3. EthSM_Init .....	389
5.4.2.2.4. EthSM_IsValidConfig .....	390
5.4.2.2.5. EthSM_MainFunction .....	390
5.4.2.2.6. EthSM_RequestComMode .....	391
5.4.2.2.7. EthSM_TcplpModeIndication .....	391
5.4.3. Integration notes .....	391
5.4.3.1. Exclusive areas .....	391
5.4.3.1.1. SCHM_ETHSM_EXCLUSIVE_AREA_0 .....	392
5.4.3.2. Production errors .....	392
5.4.3.3. Memory mapping .....	392
5.4.3.4. Integration requirements .....	393
5.4.3.4.1. lim.EthSm.EB_INTREQ_EthSm_0001 .....	393
5.5. Sd .....	393
5.5.1. Configuration parameters .....	393
5.5.1.1. CommonPublishedInformation .....	394
5.5.1.2. PublishedInformation .....	397
5.5.1.3. SdConfig .....	398



5.5.1.4. SdInstance .....	398
5.5.1.5. SdClientService .....	400
5.5.1.6. SdClientCapabilityRecord .....	404
5.5.1.7. SdConsumedEventGroup .....	404
5.5.1.8. SdClientCapabilityRecord .....	407
5.5.1.9. SdBlacklistedVersions .....	407
5.5.1.10. SdConsumedMethods .....	408
5.5.1.11. SdClientTimer .....	408
5.5.1.12. SdInstanceDemEventParameterRefs .....	411
5.5.1.13. SdInstanceMulticastRxPdu .....	412
5.5.1.14. SdInstanceTxPdu .....	413
5.5.1.15. SdInstanceUnicastRxPdu .....	413
5.5.1.16. SdServerService .....	414
5.5.1.17. SdEventHandler .....	417
5.5.1.18. SdEventHandlerMulticast .....	419
5.5.1.19. SdEventHandlerTcp .....	419
5.5.1.20. SdEventHandlerUdp .....	420
5.5.1.21. SdServerCapabilityRecord .....	421
5.5.1.22. SdProvidedMethods .....	421
5.5.1.23. SdServerCapabilityRecord .....	422
5.5.1.24. SdServerTimer .....	422
5.5.1.25. SdGeneral .....	425
5.5.1.26. VendorSpecific .....	429
5.5.1.27. ProductionErrorHandling .....	431
5.5.1.28. Sd_SendDataBackServerNotAvailable .....	435
5.5.1.29. Sd_SendDataBackMalformedMsg .....	436
5.5.1.30. Sd_SendDataBackSubscribeNackReceived .....	436
5.5.1.31. SdDefensiveProgramming .....	437
5.5.2. Application programming interface (API) .....	440
5.5.2.1. Macro constants .....	440
5.5.2.1.1. SD_CHECKBUFFERFILLLEVEL_SVCID .....	440
5.5.2.1.2. SD_CLIENTSERVICECEVGEXITAVAILABLE_SVCID .....	440
5.5.2.1.3. SD_CLIENTSERVICEGOTOAVAILABLE_SVCID .....	440
5.5.2.1.4. SD_CLIENTSERVICEMAINFUNCTION_SVCID .....	440
5.5.2.1.5. SD_CLIENTSERVICEMAITIMERS_SVCID .....	441
5.5.2.1.6. SD_CLIENTSERVICEMAITTL_SVCID .....	441
5.5.2.1.7. SD_CLIENTSERVICESETSTATE_SVCID .....	441
5.5.2.1.8. SD_CLIENTSERVICESINITAFTERTHALT_SVCID .....	441
5.5.2.1.9. SD_CLIENTSERVICE_COMSMEVENT_OFFER_SVCID .....	441
5.5.2.1.10. SD_CLIENTSERVICE_COMSMEVENT_START_SVCID .....	441
5.5.2.1.11. SD_CLIENTSERVICE_COMSMEVENT_STOPOFFER_SVCID .....	441
5.5.2.1.12. SD_CLIENTSERVICE_COMSMEVENT_STOP_SVCID .....	442



5.5.2.1.13. SD_CLIENTSERVICE_COMSMEVENT_TIMEOUT_SVCID .....	442
5.5.2.1.14. SD_CLIENTSERVICE_COMSM_EVENT_STOP_SVCID .....	442
5.5.2.1.15. SD_CLIENTSERVICE_COMSM_EVENT_TIMEOUT .....	442
5.5.2.1.16. SD_CLIENTSERVICE_COMSM_SOCONMODECHECK_SVCID .....	442
5.5.2.1.17. SD_CLIENTSERVICE_CONSUMEDEVENTGROUPSETSTATE_SVCID .....	442
5.5.2.1.18. SD_CLIENTSERVICE_EVENT_OFFER_SERVICE_SVCID .....	443
5.5.2.1.19. SD_CLIENTSERVICE_PROCESSOFFERS_SVCID .....	443
5.5.2.1.20. SD_CLIENTSERVICE_TTL_RUN_OUT .....	443
5.5.2.1.21. SD_COMPAREADDRESS_SVCID .....	443
5.5.2.1.22. SD_CONNECTION_SETTING_FAILED .....	443
5.5.2.1.23. SD_CONSUMEDEVENTGROUPSETSTATE_SVCID .....	443
5.5.2.1.24. SD_CONTROL_SOCKET_IMPOSSIBLE .....	444
5.5.2.1.25. SD_CONTROL_SOCKET_NOT_RELEASED .....	444
5.5.2.1.26. SD_E_CLIENT_IMPOSSIBLE_STATE .....	444
5.5.2.1.27. SD_E_COUNT_OF_RETRY_SUBSCRIPTION_EXCEEDED .....	444
5.5.2.1.28. SD_E_DIVISOR_IS_NEGATIVE .....	444
5.5.2.1.29. SD_E_INVALID_ARG .....	444
5.5.2.1.30. SD_E_INV_ID .....	445
5.5.2.1.31. SD_E_INV_MODE .....	445
5.5.2.1.32. SD_E_IPADDR_ASSIGNMENT_NOT_FINISHED .....	445
5.5.2.1.33. SD_E_NOT_INITIALIZED .....	445
5.5.2.1.34. SD_E_OUT_OF_RES .....	445
5.5.2.1.35. SD_E_PARAM_POINTER .....	445
5.5.2.1.36. SD_E_RETRY_INFINITE_TTL_NOT_INFINITE .....	446
5.5.2.1.37. SD_E_RETRY_PROCESS_BIGGER_THAN_TTL .....	446
5.5.2.1.38. SD_E_SHOULD_NOT_HAPPEN .....	446
5.5.2.1.39. SD_GETRANDOM_SVCID .....	446
5.5.2.1.40. SD_GETVERSIONINFO_SVCID .....	446
5.5.2.1.41. SD_HANDLERESPONSETIMERS_SVCID .....	446
5.5.2.1.42. SD_INIT_SVCID .....	446
5.5.2.1.43. SD_INSTANCEMAINFUNCTION_SVCID .....	447
5.5.2.1.44. SD_INSTANCE_ID .....	447
5.5.2.1.45. SD_INTERNAL_API_ID .....	447
5.5.2.1.46. SD_INVARIANT_ASSERT .....	447
5.5.2.1.47. SD_INVARIANT_ASSERT_NO_EVAL .....	447
5.5.2.1.48. SD_LOCALIPADDRASSIGNMENTCHG_SVCID .....	447
5.5.2.1.49. SD_MAINFUNCTION_SVCID .....	448
5.5.2.1.50. SD_POSTCONDITION_ASSERT .....	448
5.5.2.1.51. SD_POSTCONDITION_ASSERT_NO_EVAL .....	448
5.5.2.1.52. SD_PRECONDITION_ASSERT .....	448
5.5.2.1.53. SD_PRECONDITION_ASSERT_NO_EVAL .....	448



5.5.2.1.54. SD_RECEIVEVEGETENTRYOPTIONS_SVCID .....	448
5.5.2.1.55. SD_RESETREMOTENODERXSTATE_SVCID .....	449
5.5.2.1.56. SD_RXINDICATION_SVCID .....	449
5.5.2.1.57. SD_SENDQUEUEOPTIONS_SVCID .....	449
5.5.2.1.58. SD_SEND_FLUSHBUFFER_SVCID .....	449
5.5.2.1.59. SD_SERVERSERVICEADDEVENTGROUPSUBS_SVCID .....	449
5.5.2.1.60. SD_SERVERSERVICEADDMULTEVENTGROUPSUBS_SVCID .....	449
5.5.2.1.61. SD_SERVERSERVICEEVENTSTOPSUBSCRIBE_SVCID .....	450
5.5.2.1.62. SD_SERVERSERVICEHANDLETTL_SVCID .....	450
5.5.2.1.63. SD_SERVERSERVICEMAINFUNCTION_SVCID .....	450
5.5.2.1.64. SD_SERVERSERVICEMAITIMERS_SVCID .....	450
5.5.2.1.65. SD_SERVERSERVICEREMOVEEVENTGROUPSUBS_SVCID .....	450
5.5.2.1.66. SD_SERVERSERVICERESETREMOTECOMMUNICATION_SVCID .....	450
5.5.2.1.67. SD_SERVERSERVICESETSTATE_SVCID .....	451
5.5.2.1.68. SD_SERVERSERVICE_COMSMEVENT_START_SVCID .....	451
5.5.2.1.69. SD_SERVERSERVICE_COMSMEVENT_STOPHALT_SVCID .....	451
5.5.2.1.70. SD_SERVERSERVICE_COMSMEVENT_TIMEOUT_SVCID .....	451
5.5.2.1.71. SD_STATIC_ASSERT .....	451
5.5.2.1.72. SD_UNREACHABLE_CODE_ASSERT .....	451
5.5.2.2. Functions .....	452
5.5.2.2.1. Sd_ClientServiceSetState .....	452
5.5.2.2.2. Sd_ConsumedEventGroupSetState .....	452
5.5.2.2.3. Sd_GetVersionInfo .....	453
5.5.2.2.4. Sd_Init .....	453
5.5.2.2.5. Sd_IsValidConfig .....	453
5.5.2.2.6. Sd_LocallpAddrAssignmentChg .....	454
5.5.2.2.7. Sd_MainFunction .....	454
5.5.2.2.8. Sd_RxIndication .....	455
5.5.2.2.9. Sd_ServerServiceSetState .....	455
5.5.2.2.10. Sd_StartRandom .....	456
5.5.3. Integration notes .....	456
5.5.3.1. Exclusive areas .....	456
5.5.3.1.1. SCHM_SD_EXCLUSIVE_AREA_0 .....	456
5.5.3.2. Production errors .....	456
5.5.3.3. Memory mapping .....	457
5.5.3.4. Integration requirements .....	457
5.5.3.4.1. Sd.EB.IntReq.EB_INTREQ_Sd_0001 .....	457
5.5.3.4.2. Sd.EB.IntReq.EB_INTREQ_Sd_0002 .....	458
5.5.3.4.3. Sd.EB.IntReq.EB_INTREQ_Sd_0003 .....	458
5.6. SoAd .....	458
5.6.1. Configuration parameters .....	458
5.6.1.1. CommonPublishedInformation .....	459



5.6.1.2. PublishedInformation .....	462
5.6.1.3. SoAdDefensiveProgramming .....	463
5.6.1.4. SoAdGeneral .....	466
5.6.1.5. VendorSpecific .....	469
5.6.1.6. SoAdBswModules .....	474
5.6.1.7. SoAdConfig .....	477
5.6.1.8. SoAdPduRoute .....	478
5.6.1.9. SoAdPduRouteDest .....	480
5.6.1.10. SoAdSocketConnectionGroup .....	482
5.6.1.11. SoAdSocketConnection .....	486
5.6.1.12. SoAdSocketRemoteAddress .....	487
5.6.1.13. SoAdSocketProtocol .....	487
5.6.1.14. SoAdSocketUdp .....	488
5.6.1.15. SoAdSocketTcp .....	491
5.6.1.16. SoAdSocketRoute .....	494
5.6.1.17. SoAdSocketRouteDest .....	495
5.6.1.18. SoAdRoutingGroup .....	496
5.6.2. Recommended configurations .....	497
5.6.2.1. SoAdRecConfigurationDflt .....	497
5.6.2.1.1. PduR .....	497
5.6.2.1.2. Sd .....	498
5.6.2.1.3. DolP .....	498
5.6.3. Application programming interface (API) .....	499
5.6.3.1. Macro constants .....	499
5.6.3.1.1. SOAD_CHANGEPARAMETER_SVCID .....	499
5.6.3.1.2. SOAD_CLOSESOCON_SVCID .....	499
5.6.3.1.3. SOAD_COPYTXDATA_SVCID .....	499
5.6.3.1.4. SOAD_DISABLEROUTING_SVCID .....	499
5.6.3.1.5. SOAD_DISABLESPECIFICROUTING_SVCID .....	499
5.6.3.1.6. SOAD_ENABLEROUTING_SVCID .....	500
5.6.3.1.7. SOAD_ENABLESPECIFICROUTING_SVCID .....	500
5.6.3.1.8. SOAD_E_INV_ARG .....	500
5.6.3.1.9. SOAD_E_INV_LENGTH_IN_PDUHEADER .....	500
5.6.3.1.10. SOAD_E_INV_METADATA .....	500
5.6.3.1.11. SOAD_E_INV_PDUHEADER_ID .....	500
5.6.3.1.12. SOAD_E_INV_PDUID .....	501
5.6.3.1.13. SOAD_E_INV_SOCKETID .....	501
5.6.3.1.14. SOAD_E_NOBUFS .....	501
5.6.3.1.15. SOAD_E_NOTINIT .....	501
5.6.3.1.16. SOAD_E_PARAM_POINTER .....	501
5.6.3.1.17. SOAD_E_TRIGGERTXBUF .....	501
5.6.3.1.18. SOAD_E_TWO_PDUS SAME_SOCON .....	502



5.6.3.1.19. SOAD_E_TWO_SOCON_SAME_PDU .....	502
5.6.3.1.20. SOAD_GETANDRESETMEASUREMENTDATA_SVCID .....	502
5.6.3.1.21. SOAD_GETLOCALADDR_SVCID .....	502
5.6.3.1.22. SOAD_GETPHYSADDR_SVCID .....	502
5.6.3.1.23. SOAD_GETREMOTEADDR_SVCID .....	502
5.6.3.1.24. SOAD_GETSOCONID_SVCID .....	503
5.6.3.1.25. SOAD_GETSOCONMODE_SVCID .....	503
5.6.3.1.26. SOAD_GETVERSIONINFO_SVCID .....	503
5.6.3.1.27. SOAD_IFROUTINGGROUPTRANSMIT_SVCID .....	503
5.6.3.1.28. SOAD_IFSPECIFICROUTINGGROUPTRANSMIT_SVCID .....	503
5.6.3.1.29. SOAD_IFTRANSMIT_SVCID .....	503
5.6.3.1.30. SOAD_INIT_SVCID .....	504
5.6.3.1.31. SOAD_INSTANCE_ID .....	504
5.6.3.1.32. SOAD_INTERNAL_SVCID .....	504
5.6.3.1.33. SOAD_INVALID_ULFUNCID .....	504
5.6.3.1.34. SOAD_ISCONNECTIONREADY_SVCID .....	504
5.6.3.1.35. SOAD_LOCALIPADDRASSIGNMENTCHG_SVCID .....	504
5.6.3.1.36. SOAD_MAINFUNCTIONTX_SVCID .....	505
5.6.3.1.37. SOAD_MAINFUNCTION_SVCID .....	505
5.6.3.1.38. SOAD_OPENSOCON_SVCID .....	505
5.6.3.1.39. SOAD_READDHCPHOSTNAMEOPTION_SVCID .....	505
5.6.3.1.40. SOAD_RELEASEIPADDRASSIGNMENT_SVCID .....	505
5.6.3.1.41. SOAD_RELEASEREMOTEADDR_SVCID .....	505
5.6.3.1.42. SOAD_REQUESTIPADDRASSIGNMENT_SVCID .....	506
5.6.3.1.43. SOAD_RXINDICATION_SVCID .....	506
5.6.3.1.44. SOAD_SETREMOTEADDR_SVCID .....	506
5.6.3.1.45. SOAD_SETUNIQUEREMOTEADDR_SVCID .....	506
5.6.3.1.46. SOAD_TCPACCEPTED_SVCID .....	506
5.6.3.1.47. SOAD_TCPCONNECTED_SVCID .....	506
5.6.3.1.48. SOAD_TCPIPEVENT_SVCID .....	507
5.6.3.1.49. SOAD_TPCANCELRECEIVE_SVCID .....	507
5.6.3.1.50. SOAD_TPCANCELTRANSMIT_SVCID .....	507
5.6.3.1.51. SOAD_TPCCHANGEPARAMETER_SVCID .....	507
5.6.3.1.52. SOAD_TPTRANSMIT_SVCID .....	507
5.6.3.1.53. SOAD_TXCONFIRMATION_SVCID .....	507
5.6.3.1.54. SOAD_WRITEDDHCPHOSTNAMEOPTION_SVCID .....	508
5.6.3.1.55. SoAdIf_Transmit .....	508
5.6.3.1.56. SoAdTp_CancelReceive .....	508
5.6.3.1.57. SoAdTp_CancelTransmit .....	508
5.6.3.1.58. SoAdTp_Transmit .....	508
5.6.3.2. Functions .....	508
5.6.3.2.1. SoAd_ChangeParameter .....	508



5.6.3.2.2. SoAd_CloseSoCon .....	509
5.6.3.2.3. SoAd_CopyTxData .....	510
5.6.3.2.4. SoAd_DisableRouting .....	510
5.6.3.2.5. SoAd_DisableSpecificRouting .....	511
5.6.3.2.6. SoAd_EnableRouting .....	511
5.6.3.2.7. SoAd_EnableSpecificRouting .....	512
5.6.3.2.8. SoAd_GetAndResetMeasurementData .....	512
5.6.3.2.9. SoAd_GetLocalAddr .....	513
5.6.3.2.10. SoAd_GetPhysAddr .....	514
5.6.3.2.11. SoAd_GetRemoteAddr .....	515
5.6.3.2.12. SoAd_GetSoConId .....	515
5.6.3.2.13. SoAd_GetSoConMode .....	516
5.6.3.2.14. SoAd_GetVersionInfo .....	516
5.6.3.2.15. SoAd_IfRoutingGroupTransmit .....	517
5.6.3.2.16. SoAd_IfSpecificRoutingGroupTransmit .....	517
5.6.3.2.17. SoAd_IfTransmit .....	518
5.6.3.2.18. SoAd_Init .....	519
5.6.3.2.19. SoAd_IsConnectionReady .....	519
5.6.3.2.20. SoAd_IsValidConfig .....	519
5.6.3.2.21. SoAd_LocallpAddrAssignmentChg .....	520
5.6.3.2.22. SoAd_MainFunction .....	520
5.6.3.2.23. SoAd_MainFunctionTx .....	521
5.6.3.2.24. SoAd_OpenSoCon .....	521
5.6.3.2.25. SoAd_ReadDhcpHostNameOption .....	521
5.6.3.2.26. SoAd_ReleaselpAddrAssignment .....	522
5.6.3.2.27. SoAd_ReleaseRemoteAddr .....	523
5.6.3.2.28. SoAd_RequestlpAddrAssignment .....	523
5.6.3.2.29. SoAd_RxIndication .....	524
5.6.3.2.30. SoAd_SetRemoteAddr .....	524
5.6.3.2.31. SoAd_SetUniqueRemoteAddr .....	525
5.6.3.2.32. SoAd_TcpAccepted .....	526
5.6.3.2.33. SoAd_TcpConnected .....	526
5.6.3.2.34. SoAd_TcplpEvent .....	527
5.6.3.2.35. SoAd_TcpTlsAccepted .....	527
5.6.3.2.36. SoAd_TpCancelReceive .....	528
5.6.3.2.37. SoAd_TpCancelTransmit .....	528
5.6.3.2.38. SoAd_TpChangeParameter .....	528
5.6.3.2.39. SoAd_TpTransmit .....	529
5.6.3.2.40. SoAd_TxConfirmation .....	530
5.6.3.2.41. SoAd_WriteDhcpHostNameOption .....	530
5.6.4. Integration notes .....	531
5.6.4.1. Exclusive areas .....	531



5.6.4.1.1. SOAD_ENTER_CRITSEC() .....	531
5.6.4.2. Production errors .....	531
5.6.4.3. Memory mapping .....	531
5.6.4.4. Integration requirements .....	532
5.6.4.4.1. lim.SoAd.EB_INTREQ_SoAd_0001 .....	532
5.6.4.4.2. lim.SoAd.EB_INTREQ_SoAd_0003 .....	533
5.6.4.4.3. lim.SoAd.EB_INTREQ_SoAd_0004 .....	535
5.6.4.4.4. lim.SoAd.EB_INTREQ_SoAd_0006 .....	535
5.6.4.4.5. lim.SoAd.EB_INTREQ_SoAd_0008 .....	535
5.6.4.4.6. lim.SoAd.EB_INTREQ_SoAd_0009 .....	535
5.6.4.4.7. lim.SoAd.EB_INTREQ_SoAd_0010 .....	535
5.6.4.4.8. lim.SoAd.EB_INTREQ_SoAd_0011 .....	536
5.6.4.4.9. lim.SoAd.EB_INTREQ_SoAd_0012 .....	536
5.6.4.4.10. lim.SoAd.EB_INTREQ_SoAd_0013 .....	536
5.7. SomelpTp .....	536
5.7.1. Configuration parameters .....	536
5.7.1.1. CommonPublishedInformation .....	537
5.7.1.2. PublishedInformation .....	540
5.7.1.3. SomelpTpChannel .....	541
5.7.1.4. SomelpTpRxNSdu .....	542
5.7.1.5. SomelpTpRxNPdu .....	543
5.7.1.6. SomelpTpTxNSdu .....	544
5.7.1.7. SomelpTpTxNPdu .....	545
5.7.1.8. SomelpTpUnSupportedChannel .....	545
5.7.1.9. SomelpTpUnSupportedRxNSdu .....	546
5.7.1.10. SomelpTpUnSupportedRxNPdu .....	546
5.7.1.11. SomelpTpUnSupportedTxNSdu .....	546
5.7.1.12. SomelpTpUnSupportedTxNPdu .....	546
5.7.1.13. SomelpTpGeneral .....	546
5.7.2. Application programming interface (API) .....	548
5.7.2.1. Macro constants .....	548
5.7.2.1.1. SOMEIPTP_DET_REPORT_ERROR .....	548
5.7.2.1.2. SOMEIPTP_DET_REPORT_RUNTIME_ERROR .....	548
5.7.2.1.3. SOMEIPTP_E_ASSEMBLY_INTERRUPT .....	548
5.7.2.1.4. SOMEIPTP_E_ASSEMBLY_INTERRUPT_SOR .....	548
5.7.2.1.5. SOMEIPTP_E_DISASSEMBLY_INTERRUPT .....	549
5.7.2.1.6. SOMEIPTP_E_INCONSISTENT_HEADER .....	549
5.7.2.1.7. SOMEIPTP_E_INCONSISTENT_SEQUENCE .....	549
5.7.2.1.8. SOMEIPTP_E_INVALID_CALL .....	549
5.7.2.1.9. SOMEIPTP_E_MESSAGE_TYPE .....	549
5.7.2.1.10. SOMEIPTP_E_MISSING_SOMEIPTP_HEADER .....	549
5.7.2.1.11. SOMEIPTP_E_NOTINIT .....	549



5.7.2.1.12. SOMEIPTP_E_PARAM .....	550
5.7.2.1.13. SOMEIPTP_E_PARAM_POINTER .....	550
5.7.2.1.14. SOMEIPTP_INSTANCE_ID .....	550
5.7.2.1.15. SOMEIPTP_SID_GETVERSIONINFO .....	550
5.7.2.1.16. SOMEIPTP_SID_INIT .....	550
5.7.2.1.17. SOMEIPTP_SID_MAINFUNCTIONRX .....	550
5.7.2.1.18. SOMEIPTP_SID_MAINFUNCTIONTX .....	551
5.7.2.1.19. SOMEIPTP_SID_PDURIFTRANSMIT .....	551
5.7.2.1.20. SOMEIPTP_SID_PDURTPCOPYRXDATA .....	551
5.7.2.1.21. SOMEIPTP_SID_PDURTPCOPYTXDATA .....	551
5.7.2.1.22. SOMEIPTP_SID_PDURTPRXINDICATION .....	551
5.7.2.1.23. SOMEIPTP_SID_PDURTPSTARTOFRECEPTION .....	551
5.7.2.1.24. SOMEIPTP_SID_PDURTPTXCONFIRMATION .....	551
5.7.2.1.25. SOMEIPTP_SID_RXINDICATION .....	552
5.7.2.1.26. SOMEIPTP_SID_TRANSMIT .....	552
5.7.2.1.27. SOMEIPTP_SID_TRIGGERTRANSMIT .....	552
5.7.2.1.28. SOMEIPTP_SID_TXCONFIRMATION .....	552
5.7.2.2. Functions .....	552
5.7.2.2.1. SomelpTp_GetVersionInfo .....	552
5.7.2.2.2. SomelpTp_Init .....	553
5.7.2.2.3. SomelpTp_MainFunctionRx .....	553
5.7.2.2.4. SomelpTp_MainFunctionTx .....	553
5.7.2.2.5. SomelpTp_PduRIfTransmit .....	553
5.7.2.2.6. SomelpTp_PduRTpCopyRxData .....	554
5.7.2.2.7. SomelpTp_PduRTpCopyTxData .....	554
5.7.2.2.8. SomelpTp_PduRTpRxIndication .....	554
5.7.2.2.9. SomelpTp_PduRTpStartOfReception .....	554
5.7.2.2.10. SomelpTp_PduRTpTxConfirmation .....	555
5.7.2.2.11. SomelpTp_RxIndication .....	555
5.7.2.2.12. SomelpTp_Transmit .....	555
5.7.2.2.13. SomelpTp_TriggerTransmit .....	556
5.7.2.2.14. SomelpTp_TxConfirmation .....	557
5.7.3. Integration notes .....	557
5.7.3.1. Exclusive areas .....	557
5.7.3.1.1. SCHM_SOMEIPTP_EXCLUSIVE_AREA_0 .....	557
5.7.3.1.2. SCHM_SOMEIPTP_EXCLUSIVE_AREA_1 .....	557
5.7.3.2. Production errors .....	558
5.7.3.3. Memory mapping .....	558
5.7.3.4. Integration requirements .....	558
5.8. Tcplp .....	558
5.8.1. Configuration parameters .....	558
5.8.1.1. CommonPublishedInformation .....	559



5.8.1.2. PublishedInformation .....	562
5.8.1.3. TcpIpConfig .....	563
5.8.1.4. TcpIpCtrl .....	564
5.8.1.5. TcpIpOffloadChecksum .....	565
5.8.1.6. TcpIpIpVXCtrl .....	566
5.8.1.7. TcpIpIpV4Ctrl .....	567
5.8.1.8. TcpIpIpV4MtuConfig .....	568
5.8.1.9. TcpIpIpV6Ctrl .....	569
5.8.1.10. TcpIpIpV6MtuConfig .....	570
5.8.1.11. TcpIpDhcpServerConfig .....	571
5.8.1.12. TcpIpDhcpAddressAssignment .....	572
5.8.1.13. TcpIpIpConfig .....	573
5.8.1.14. TcpIpIpFragmentationConfig .....	573
5.8.1.15. TcpIpIpV4Config .....	577
5.8.1.16. TcpIpArpConfig .....	578
5.8.1.17. TcpIpAutoIpConfig .....	580
5.8.1.18. TcpIpDhcpConfig .....	582
5.8.1.19. TcpIpDhcpConfigurableOptions .....	587
5.8.1.20. TcpIpIcmpConfig .....	588
5.8.1.21. TcpIpIcmpMsgHandler .....	589
5.8.1.22. TcpIpIpV4ArpPacketFilter .....	589
5.8.1.23. TcpIpIpFragmentationConfig .....	590
5.8.1.24. TcpIpIpV6Config .....	592
5.8.1.25. TcpIpDhcpV6Config .....	592
5.8.1.26. TcpIpDhcpV6ConfigurableOptions .....	599
5.8.1.27. TcpIpIcmpV6Config .....	600
5.8.1.28. TcpIpIcmpV6MsgHandler .....	602
5.8.1.29. TcpIpIpV6FragmentationConfig .....	603
5.8.1.30. TcpIpNdpConfig .....	605
5.8.1.31. TcpIpNdpArNudConfig .....	605
5.8.1.32. TcpIpNdpPrefixRouterDiscoveryConfig .....	610
5.8.1.33. TcpIpNdpPrefixList .....	614
5.8.1.34. TcpIpNdpPrefixListEntry .....	614
5.8.1.35. TcpIpNdpSlaacConfig .....	615
5.8.1.36. TcpIpIpV6ConfigExtHeaderFilter .....	617
5.8.1.37. TcpIpLocalAddr .....	617
5.8.1.38. TcpIpAddrAssignment .....	619
5.8.1.39. TcpIpStaticIpAddressConfig .....	622
5.8.1.40. TcpIpNvmBlock .....	623
5.8.1.41. TcpIpPhysAddrConfig .....	624
5.8.1.42. TcpIpPhysAddrChgHandler .....	624
5.8.1.43. TcpIpSocketOwnerConfig .....	625



5.8.1.44. TcplpSocketOwner .....	625
5.8.1.45. TcplpTcpConfig .....	628
5.8.1.46. TcplpTcpConfigOptionFilter .....	638
5.8.1.47. TcplpTcpUnpredictableSeqNumbers .....	639
5.8.1.48. TcplpTcpSynCookies .....	640
5.8.1.49. TcplpUdpConfig .....	641
5.8.1.50. TcplpDuplicateAddressDetectionConfig .....	642
5.8.1.51. TcplpRxPolicyCheckIngressHandlerConfig .....	643
5.8.1.52. TcplpRxPolicyCheckRoutingHandlerConfig .....	644
5.8.1.53. TcplpRxPacketPostProcessHandlerConfig .....	644
5.8.1.54. TcplpRxPacketDropHandlerConfig .....	645
5.8.1.55. TcplpMemoryConfig .....	646
5.8.1.56. TcplpMemoryPool .....	646
5.8.1.57. TcplpIpSecConfig .....	647
5.8.1.58. TcplpIpSecGeneral .....	647
5.8.1.59. TcplpIpSecConnections .....	649
5.8.1.60. IpSecRemoteAddrConfig .....	651
5.8.1.61. SingleAddress .....	651
5.8.1.62. AddressRange .....	651
5.8.1.63. TcplpIpSecRemotePhysAddr .....	652
5.8.1.64. TcplpIpSecSecurityPolicies .....	653
5.8.1.65. TcplpIpSecSecurityRule .....	654
5.8.1.66. TcplpIpSecSecurityAssociationCfg .....	656
5.8.1.67. TcplpIpSecSecurityProposal .....	658
5.8.1.68. TcplpIpSecSecurityAssociation .....	659
5.8.1.69. TcplpIpSecReportErrorHandler .....	660
5.8.1.70. TcplpDefensiveProgramming .....	661
5.8.1.71. TcplpGeneral .....	664
5.8.1.72. TcplpIpV4General .....	674
5.8.1.73. TcplpIpV6General .....	677
5.8.2. Recommended configurations .....	681
5.8.2.1. TcplpRecConfiguration .....	682
5.8.2.1.1. TcplpConfig .....	682
5.8.2.1.2. TcplpSocketOwnerConfig .....	682
5.8.2.1.3. TcplpSocketOwner .....	682
5.8.2.1.4. TcplpIpConfig .....	683
5.8.2.1.5. TcplpIpV4Config .....	683
5.8.2.1.6. TcplpArpConfig .....	683
5.8.2.3. Application programming interface (API) .....	684
5.8.3.1. Type definitions .....	684
5.8.3.1.1. Tcplp_ArpCacheEntryType .....	684
5.8.3.1.2. Tcplp_EventType .....	684



5.8.3.1.3. Tcplp_IPsecStateType .....	684
5.8.3.1.4. Tcplp_IpAddrAssignmentType .....	684
5.8.3.1.5. Tcplp_IpAddrStateType .....	685
5.8.3.1.6. Tcplp_Ip_HandleIdType .....	685
5.8.3.1.7. Tcplp_Ip_RxReturnType .....	685
5.8.3.1.8. Tcplp_NdpCacheEntryType .....	685
5.8.3.1.9. Tcplp_ParamIdType .....	685
5.8.3.1.10. Tcplp_ProtocolType .....	685
5.8.3.1.11. Tcplp_ReturnType .....	686
5.8.3.1.12. Tcplp_SockAddrPtrType .....	686
5.8.3.1.13. Tcplp_SockAddrType .....	686
5.8.3.1.14. Tcplp_SocketIdType .....	686
5.8.3.1.15. Tcplp_StateType .....	686
5.8.3.2. Macro constants .....	687
5.8.3.2.1. TCPIP_AF_INET .....	687
5.8.3.2.2. TCPIP_AF_INET6 .....	687
5.8.3.2.3. TCPIP_AF_UNSPEC .....	687
5.8.3.2.4. TCPIP_DHCPV6_OPTION_FQDN .....	687
5.8.3.2.5. TCPIP_DHCP_OPTION_FQDN .....	687
5.8.3.2.6. TCPIP_E_ADDRINUSE .....	688
5.8.3.2.7. TCPIP_E_ADDRNOTASSIGNED .....	688
5.8.3.2.8. TCPIP_E_ADDRNOTAVAIL .....	688
5.8.3.2.9. TCPIP_E_AFNOSUPPORT .....	688
5.8.3.2.10. TCPIP_E_DESTADDRREQ .....	688
5.8.3.2.11. TCPIP_E_DROP .....	688
5.8.3.2.12. TCPIP_E_INV_ARG .....	689
5.8.3.2.13. TCPIP_E_INV_SOCKADDR .....	689
5.8.3.2.14. TCPIP_E_ISCONN .....	689
5.8.3.2.15. TCPIP_E_MEMORY .....	689
5.8.3.2.16. TCPIP_E_MSGSIZE .....	689
5.8.3.2.17. TCPIP_E_NOBUFS .....	689
5.8.3.2.18. TCPIP_E_NOPROTOOPT .....	689
5.8.3.2.19. TCPIP_E_NOTCONN .....	690
5.8.3.2.20. TCPIP_E_NOTINIT .....	690
5.8.3.2.21. TCPIP_E_NOT_OK .....	690
5.8.3.2.22. TCPIP_E_NOT_PERMITTED .....	690
5.8.3.2.23. TCPIP_E_PARAM_POINTER .....	690
5.8.3.2.24. TCPIP_E_PENDING .....	690
5.8.3.2.25. TCPIP_E_PHYS_ADDR_MISS .....	691
5.8.3.2.26. TCPIP_E_PROTOTYPE .....	691
5.8.3.2.27. TCPIP_IPADDR_ASSIGNMENT_ALL .....	691
5.8.3.2.28. TCPIP_IPADDR_ASSIGNMENT_DHCP .....	691



5.8.3.2.29. TCPIP_IPADDR_ASSIGNMENT_IPV6_ROUTER .....	691
5.8.3.2.30. TCPIP_IPADDR_ASSIGNMENT_LINKLOCAL .....	691
5.8.3.2.31. TCPIP_IPADDR_ASSIGNMENT_LINKLOCAL_DOIP .....	691
5.8.3.2.32. TCPIP_IPADDR_ASSIGNMENT_STATIC .....	692
5.8.3.2.33. TCPIP_IPADDR_STATE_ASSIGNED .....	692
5.8.3.2.34. TCPIP_IPADDR_STATE_ONHOLD .....	692
5.8.3.2.35. TCPIP_IPADDR_STATE_UNASSIGNED .....	692
5.8.3.2.36. TCPIP IPPROTO_TCP .....	692
5.8.3.2.37. TCPIP IPPROTO_UDP .....	692
5.8.3.2.38. TCPIP_IPSEC_ACTIVE .....	693
5.8.3.2.39. TCPIP_IPSEC_INACTIVE .....	693
5.8.3.2.40. TCPIP_IPSEC_INVALID_ICV .....	693
5.8.3.2.41. TCPIP_IP_RX_DEST_UNREACHABLE .....	693
5.8.3.2.42. TCPIP_IP_RX_EXT_HDR_OK .....	693
5.8.3.2.43. TCPIP_IP_RX_FRAGMENT_HEADER_SHORT .....	693
5.8.3.2.44. TCPIP_IP_RX_FRAGMENT_LENGTH .....	693
5.8.3.2.45. TCPIP_IP_RX_FRAGMENT_OFFSET .....	694
5.8.3.2.46. TCPIP_IP_RX_FRAGMENT_OUT_OF_BUFFERS .....	694
5.8.3.2.47. TCPIP_IP_RX_FRAGMENT_OVERLAP .....	694
5.8.3.2.48. TCPIP_IP_RX_FRAGMENT_RESERVED .....	694
5.8.3.2.49. TCPIP_IP_RX_FRAGMENT_TIMEOUT .....	694
5.8.3.2.50. TCPIP_IP_RX_INVALID_HEADER .....	694
5.8.3.2.51. TCPIP_IP_RX_NOT_OK .....	695
5.8.3.2.52. TCPIP_IP_RX_OK .....	695
5.8.3.2.53. TCPIP_IP_RX_PORT_UNREACHABLE .....	695
5.8.3.2.54. TCPIP_IP_RX_PROTOCOL_UNREACHABLE .....	695
5.8.3.2.55. TCPIP_IP_RX_PROTOCOL_UNREACHABLE_FRAG .....	695
5.8.3.2.56. TCPIP_IP_RX_RES_POLICYCHECK .....	695
5.8.3.2.57. TCPIP_IP_RX_SRCADDR_INVALID .....	695
5.8.3.2.58. TCPIP_IP_RX_UNRECOGNIZED_OPTION .....	696
5.8.3.2.59. TCPIP_IP_RX_UNRECOGNIZED_ROUTING_TYPE .....	696
5.8.3.2.60. TCPIP_LOCALADDRID_ANY .....	696
5.8.3.2.61. TCPIP_OK .....	696
5.8.3.2.62. TCPIP_PARAMID_DSCP .....	696
5.8.3.2.63. TCPIP_PARAMID_FLOWLABEL .....	696
5.8.3.2.64. TCPIP_PARAMID_FRAMEPRIO .....	697
5.8.3.2.65. TCPIP_PARAMID_PATHMTU_ENABLE .....	697
5.8.3.2.66. TCPIP_PARAMID_TCP_KEEPALIVE .....	697
5.8.3.2.67. TCPIP_PARAMID_TCP_KEEPALIVE_INTERVAL .....	697
5.8.3.2.68. TCPIP_PARAMID_TCP_KEEPALIVE_PROBES_MAX .....	697
5.8.3.2.69. TCPIP_PARAMID_TCP_KEEPALIVE_TIME .....	697
5.8.3.2.70. TCPIP_PARAMID_TCP_NAGLE .....	698



5.8.3.2.71. TCPIP_PARAMID_TCP_OPTIONFILTER .....	698
5.8.3.2.72. TCPIP_PARAMID_TCP_RXWND_MAX .....	698
5.8.3.2.73. TCPIP_PARAMID_TTL .....	698
5.8.3.2.74. TCPIP_PARAMID_UDP_CHECKSUM .....	698
5.8.3.2.75. TCPIP_PARAMID_UDP_UNSPECIFIED_IP .....	698
5.8.3.2.76. TCPIP_PHYS_ADDR_ENTRY_STATE_DELAY .....	699
5.8.3.2.77. TCPIP_PHYS_ADDR_ENTRY_STATE_FREE .....	699
5.8.3.2.78. TCPIP_PHYS_ADDR_ENTRY_STATE_INCOMPLETE .....	699
5.8.3.2.79. TCPIP_PHYS_ADDR_ENTRY_STATE_PROBE .....	699
5.8.3.2.80. TCPIP_PHYS_ADDR_ENTRY_STATE_REACHABLE .....	699
5.8.3.2.81. TCPIP_PHYS_ADDR_ENTRY_STATE_STALE .....	699
5.8.3.2.82. TCPIP_PHYS_ADDR_ENTRY_STATE_STATIC .....	700
5.8.3.2.83. TCPIP_PORT_ANY .....	700
5.8.3.2.84. TCPIP_SOCKETID_INVALID .....	700
5.8.3.2.85. TCPIP_STATE_OFFLINE .....	700
5.8.3.2.86. TCPIP_STATE_ONHOLD .....	700
5.8.3.2.87. TCPIP_STATE_ONLINE .....	700
5.8.3.2.88. TCPIP_STATE_SHUTDOWN .....	701
5.8.3.2.89. TCPIP_STATE_STARTUP .....	701
5.8.3.2.90. TCPIP_TCP_CLOSED .....	701
5.8.3.2.91. TCPIP_TCP_FIN_RECEIVED .....	701
5.8.3.2.92. TCPIP_TCP_RESET .....	701
5.8.3.2.93. TCPIP_UDP_CLOSED .....	701
5.8.3.3. Functions .....	702
5.8.3.3.1. Tcplp_Bind .....	702
5.8.3.3.2. Tcplp_ChangeParameter .....	703
5.8.3.3.3. Tcplp_Close .....	703
5.8.3.3.4. Tcplp_DhcpGetStatus .....	704
5.8.3.3.5. Tcplp_DhcpReadOption .....	705
5.8.3.3.6. Tcplp_DhcpV6ReadOption .....	705
5.8.3.3.7. Tcplp_DhcpV6WriteOption .....	706
5.8.3.3.8. Tcplp_DhcpWriteOption .....	707
5.8.3.3.9. Tcplp_GetAndResetMeasurementData .....	707
5.8.3.3.10. Tcplp_GetArpCacheEntries .....	708
5.8.3.3.11. Tcplp_GetCtrlIdx .....	709
5.8.3.3.12. Tcplp_GetIpAddr .....	710
5.8.3.3.13. Tcplp_GetNdpCacheEntries .....	710
5.8.3.3.14. Tcplp_GetPhysAddr .....	711
5.8.3.3.15. Tcplp_GetRemotePhysAddr .....	712
5.8.3.3.16. Tcplp_GetVersionInfo .....	713
5.8.3.3.17. Tcplp_IcmpTransmit .....	713
5.8.3.3.18. Tcplp_IcmpTransmitError .....	714



5.8.3.3.19. Tcplp_Init .....	715
5.8.3.3.20. Tcplp_IsConnectionReady .....	715
5.8.3.3.21. Tcplp_IsValidConfig .....	716
5.8.3.3.22. Tcplp_MainFunction .....	716
5.8.3.3.23. Tcplp_MainFunctionTx .....	717
5.8.3.3.24. Tcplp_ReleaselpAddrAssignment .....	717
5.8.3.3.25. Tcplp_RequestComMode .....	717
5.8.3.3.26. Tcplp_RequestIpAddrAssignment .....	718
5.8.3.3.27. Tcplp_RequestIpSecMode .....	719
5.8.3.3.28. Tcplp_RxIndication .....	719
5.8.3.3.29. Tcplp_SetRemotePhysAddr .....	720
5.8.3.3.30. Tcplp_Socket_getSocket .....	721
5.8.3.3.31. Tcplp_TcpConnect .....	722
5.8.3.3.32. Tcplp_TcpListen .....	722
5.8.3.3.33. Tcplp_TcpReceived .....	723
5.8.3.3.34. Tcplp_TcpTransmit .....	724
5.8.3.3.35. Tcplp_UdpTransmit .....	725
5.8.4. Integration notes .....	726
5.8.4.1. Exclusive areas .....	726
5.8.4.1.1. SCHM_TCPIP_EXCLUSIVE_AREA_0 .....	726
5.8.4.1.2. SCHM_TCPIP_EXCLUSIVE_AREA_1 .....	726
5.8.4.1.3. SCHM_TCPIP_EXCLUSIVE_AREA_2 .....	727
5.8.4.1.4. SCHM_TCPIP_EXCLUSIVE_AREA_3 .....	727
5.8.4.2. Production errors .....	727
5.8.4.3. Memory mapping .....	727
5.8.4.4. Integration requirements .....	728
5.8.4.4.1. Tcplp.EB_INTREQ_Tcplp_0001 .....	728
5.8.4.4.2. Tcplp.EB_INTREQ_Tcplp_0002 .....	728
5.8.4.4.3. Tcplp.EB_INTREQ_Tcplp_0003 .....	729
5.8.4.4.4. Tcplp.EB_INTREQ_Tcplp_0004 .....	729
5.8.4.4.5. Tcplp.EB_INTREQ_Tcplp_0005 .....	729
5.8.4.4.6. Tcplp.EB_INTREQ_Tcplp_0006 .....	729
5.8.4.4.7. Tcplp.EB_INTREQ_Tcplp_0007 .....	730
5.8.4.4.8. Tcplp.EB_INTREQ_Tcplp_0008 .....	730
5.8.4.4.9. Tcplp.EB_INTREQ_uisnSecretKeyLen .....	730
5.8.4.4.10. Tcplp.EB_INTREQ_secretHashOutLen .....	730
5.8.4.4.11. Tcplp.EB_INTREQ_storingIpAddresses .....	730
5.8.4.4.12. Tcplp.EB_INTREQ_ASILD_EthIf_ProvideTxBuffer .....	730
5.8.4.4.13. Tcplp.EB_INTREQ_ASILD_EthIf_GetPhysAddr .....	731
5.8.4.4.14. Tcplp.EB_INTREQ_ASILD_Tcplp_GetRemotePhysAddr .....	731
5.9. UdpNm .....	731
5.9.1. Configuration parameters .....	731



5.9.1.1. CommonPublishedInformation .....	732
5.9.1.2. PublishedInformation .....	735
5.9.1.3. UdpNmGeneral .....	735
5.9.1.4. UdpNmGlobalConfig .....	737
5.9.1.5. UdpNmChannelConfig .....	745
5.9.1.6. UdpNmRxPdu .....	758
5.9.1.7. UdpNmTxPdu .....	759
5.9.1.8. UdpNmUserDataTxPdu .....	759
5.9.1.9. UdpNmUserDataRxPdu .....	760
5.9.1.10. UdpNmPnInfo .....	760
5.9.1.11. UdpNmPnFilterMaskByte .....	761
5.9.1.12. UdpNmDefensiveProgramming .....	762
5.9.2. Application programming interface (API) .....	765
5.9.2.1. Macro constants .....	765
5.9.2.1.1. UDPNM_AR_RELEASE_MAJOR_VERSION .....	765
5.9.2.1.2. UDPNM_AR_RELEASE_MINOR_VERSION .....	765
5.9.2.1.3. UDPNM_AR_RELEASE_REVISION_VERSION .....	766
5.9.2.1.4. UDPNM_E_BUSSLEEPMODE .....	766
5.9.2.1.5. UDPNM_E_CARWAKEUPINDICATION .....	766
5.9.2.1.6. UDPNM_E_INIT_FAILED .....	766
5.9.2.1.7. UDPNM_E_INVALID_CHANNEL .....	766
5.9.2.1.8. UDPNM_E_INVALID_PDUID .....	766
5.9.2.1.9. UDPNM_E_NETWORKMODE .....	767
5.9.2.1.10. UDPNM_E_NETWORKSTARTINDICATION .....	767
5.9.2.1.11. UDPNM_E_NO_INIT .....	767
5.9.2.1.12. UDPNM_E_NULL_POINTER .....	767
5.9.2.1.13. UDPNM_E_PDURXINDICATION .....	767
5.9.2.1.14. UDPNM_E_PREPAREBUSSLEEPMODE .....	767
5.9.2.1.15. UDPNM_E_REMOTESENTRYCANCELLATION .....	768
5.9.2.1.16. UDPNM_E_REMOTESENTRYINDICATION .....	768
5.9.2.1.17. UDPNM_E_REPEATMESSAGEINDICATION .....	768
5.9.2.1.18. UDPNM_E_STATECHANGENOTIFICATION .....	768
5.9.2.1.19. UDPNM_INSTANCE_ID .....	768
5.9.2.1.20. UDPNM_INVALID_PDU_INSTANCE_ID .....	768
5.9.2.1.21. UDPNM_MODULE_ID .....	769
5.9.2.1.22. UDPNM_SERVID_CHECKREMOTESLEEPINDICATION .....	769
5.9.2.1.23. UDPNM_SERVID_DISABLECOMMUNICATION .....	769
5.9.2.1.24. UDPNM_SERVID_ENABLECOMMUNICATION .....	769
5.9.2.1.25. UDPNM_SERVID_GETLOCALNODEIDENTIFIER .....	769
5.9.2.1.26. UDPNM_SERVID_GETNODEIDENTIFIER .....	769
5.9.2.1.27. UDPNM_SERVID_GETPDUDATA .....	770
5.9.2.1.28. UDPNM_SERVID_GETSTATE .....	770



5.9.2.1.29. UDPNM_SERVID_GETUSERDATA .....	770
5.9.2.1.30. UDPNM_SERVID_GETVERSIONINFO .....	770
5.9.2.1.31. UDPNM_SERVID_INIT .....	770
5.9.2.1.32. UDPNM_SERVID_MAINFUNCTION_X .....	771
5.9.2.1.33. UDPNM_SERVID_NETWORKGWERAREQUEST .....	771
5.9.2.1.34. UDPNM_SERVID_NETWORKRELEASE .....	771
5.9.2.1.35. UDPNM_SERVID_NETWORKREQUEST .....	771
5.9.2.1.36. UDPNM_SERVID_PASSIVESTARTUP .....	771
5.9.2.1.37. UDPNM_SERVID_REPEATMESSAGEREQUEST .....	772
5.9.2.1.38. UDPNM_SERVID_REQUESTBUSSYNCHRONIZATION .....	772
5.9.2.1.39. UDPNM_SERVID_RXINDICATION .....	772
5.9.2.1.40. UDPNM_SERVID_SETUSERDATA .....	772
5.9.2.1.41. UDPNM_SERVID_TXCONFIRMATION .....	772
5.9.2.1.42. UDPNM_SW_MAJOR_VERSION .....	773
5.9.2.1.43. UDPNM_SW_MINOR_VERSION .....	773
5.9.2.1.44. UDPNM_SW_PATCH_VERSION .....	773
5.9.2.1.45. UDPNM_VENDOR_ID .....	773
5.9.2.2. Functions .....	773
5.9.2.2.1. UdpNm_CheckRemoteSleepIndication .....	773
5.9.2.2.2. UdpNm_DisableCommunication .....	774
5.9.2.2.3. UdpNm_EnableCommunication .....	774
5.9.2.2.4. UdpNm_GetLocalNodeIdentifier .....	775
5.9.2.2.5. UdpNm_GetNodeIdentifier .....	775
5.9.2.2.6. UdpNm_GetPduData .....	776
5.9.2.2.7. UdpNm_GetState .....	777
5.9.2.2.8. UdpNm_GetUserData .....	777
5.9.2.2.9. UdpNm_GetVersionInfo .....	778
5.9.2.2.10. UdpNm_Init .....	778
5.9.2.2.11. UdpNm_IsValidConfig .....	778
5.9.2.2.12. UdpNm_NetworkGwEraRequest .....	779
5.9.2.2.13. UdpNm_NetworkRelease .....	779
5.9.2.2.14. UdpNm_NetworkRequest .....	780
5.9.2.2.15. UdpNm_PassiveStartUp .....	780
5.9.2.2.16. UdpNm_RepeatMessageRequest .....	781
5.9.2.2.17. UdpNm_RequestBusSynchronization .....	781
5.9.2.2.18. UdpNm_SetUserData .....	782
5.9.2.2.19. UdpNm_SoAdlfRxIndication .....	782
5.9.2.2.20. UdpNm_SoAdlfTxConfirmation .....	783
5.9.2.2.21. UdpNm_Transmit .....	783
5.9.3. Integration notes .....	784
5.9.3.1. Exclusive areas .....	784
5.9.3.1.1. SCHM_UDPNM_EXCLUSIVE_AREA_0 .....	784

---

5.9.3.2. Production errors .....	785
5.9.3.3. Memory mapping .....	785
5.9.3.4. Integration requirements .....	785
6. Bibliography .....	786



# 1. Overview of EB tresos AutoCore Generic 8 IP Stack documentation

Welcome to the EB tresos AutoCore Generic 8 IP Stack (ACG8 IP Stack) product documentation.

This document provides:

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



## 2. Supported features

The ACG8 IP Stack product provides the following main features:

- ▶ **AUTOSAR BSW architecture integration:** Ethernet/IP stack fully integrated into the AUTOSAR BSW architecture, enabling
  - ▶ Ethernet communication between application SWCs via Rte
  - ▶ Gateway functionality to CAN, FlexRay, and LIN networks via PduR
  - ▶ Diagnosis communication on Ethernet/IP via Dcm and DoIP
  - ▶ Network management via Nm
  - ▶ Interface-PDU (i.e. non-segmented) and TP-PDU (i.e. segmented) communication via UDP and TCP sockets
  - ▶ Support for multiple upper layers, e.g. to PduR, DoIP, BMW EthDiagMM, SD, V2G, etc.
- ▶ **Versatile control API:** Control API for connection handling, dynamic configuration, and to retrieve stack information, e.g. MAC address of local interface
- ▶ **Ethernet communication startup and shutdown handling:** Activates and deactivates the Ethernet transceiver and the Ethernet controller and triggers IP address assignment via DHCP, AutoIP, or using static configuration.
- ▶ **VLAN support (IEEE 802.1Q VLAN tagging):** Automatic adding (upon sending) and processing and removing (upon receiving) of the IEEE 802.1Q VLAN tag based on static configuration (i.e. VLAN identifier (VID)) and on dedicated parameters provided by the upper layers, i.e. priority code point (PCP).
- ▶ **AUTOSAR 4.x compliant Ethernet MCAL interface:** AUTOSAR 4.0.3, 4.2.2, 4.3.0 compliant interface to the Ethernet driver (Eth) and the Ethernet transceiver driver (EthTrcv) modules.
- ▶ **Prioritized data handling for quality of service (QoS) support:** Prioritized sending and receiving of data (e.g. priority-based queue selection for sending and receiving) via extended APIs (for handling of the priority) according to AUTOSAR 4.3.0 (RfC #64825). This feature requires the ACM8 Eth QoS Support product.
- ▶ **IPv4 protocol family (IPv4, ICMPv4, ARP, DHCPv4):** Implementation of the IP protocol family for Internet Protocol version 4 (IPv4) including:
  - ▶ **Core Internet Protocol version 4 (IPv4) support:** Sending/receiving/processing of IPv4 datagrams according to the format defined in IETF RFC 791 and RFC 894.
  - ▶ **IP fragmentation (Tx) and reassembly (Rx):** IP fragmentation (Tx) and reassembly (Rx) according to AUTOSAR 4.3.0 (RfC #69054) and IETF RFC 2460 including the forbidding of overlapping fragments according to IETF RFC 5722 and the handling of atomic fragments according to IETF RFC 6946.
  - ▶ **Address resolution protocol (ARP) according to IETF RFC 826:** This feature facilitates the mapping of a network address (i.e. IPv4 addresses) to a physical address like an Ethernet/MAC address by sending/receiving/processing ARP request messages and ARP response messages.



- ▶ **Prevention of ARP flooding:** Prevent ARP flooding by having a configurable maximum rate and a maximum number of ARP requests.
- ▶ **Auto-IP according to IETF RFC 3927:** This feature facilitates the automatic dynamic configuration of IPv4 link local addresses (including the address conflict detection) within the address block 169.-254.0.0/16 by exchanging ARP probes.
- ▶ **Internet control message protocol version 4 (ICMPv4) according to IETF RFC 792:** ICMPv4 provides error reporting functionality by sending/receiving/processing error messages (destination (port) unreachable) as well as diagnostic functionality (echo request and echo reply).
- ▶ **Dynamic host configuration protocol version 4 (DHCPv4) client functionality according to IETF RFC 2131:** The DHCPv4 client functionality provides means to dynamically obtain configuration parameters, such as IP addresses, for interfaces and services from a DHCPv4 server.

Hereby the following types of DHCPv4 messages are sent by the client: Discover messages (to discover DHCPv4 servers) and request messages (to request the provision of address and configuration information).

The following DHCPv4 messages are received/processed by the client: Offer messages (sent by the server as a response to discover messages) and acknowledgment messages (sent as answers to request messages).

The DHCPv4 client functionality includes support for the fully qualified domain name (FQDN) option (excluding DNS support) according to IETF RFC 4702.

The DHCPv4 client functionality supports the configuration of custom DHCP options that are readable/writable via `TcpIp_DhcpV4Read/WriteOption()`.

- ▶ **Configurable DSCP and flow label:** Configurable Differentiated Services Codepoint (DSCP) according to IETF RFC 6437 and Flow Label according to IETF RFC 2474.
- ▶ **Configurable UDP checksum calculation:** It can be configured whether the UDP checksum is checked for received frames and added for transmitted frames.
- ▶ **IPv6 protocol family (IPv6, ICMPv6, NDP, DHCPv6):** Implementation of the Internet Protocol version 6 according to AUTOSAR 4.3.0, including:
  - ▶ **Core Internet Protocol version 6 (IPv6) support:** Sending/receiving/processing of IPv6 datagrams according to the format defined in IETF RFC 2464. Stateless address auto configuration (SLAAC) according to IETF RFC 4862 for the creation/assignment of link-local IP addresses. Source address selection according to IETF RFC 6724.
  - ▶ **IP fragmentation (Tx) and reassembly (Rx):** IP fragmentation (Tx) and reassembly (Rx) according to AUTOSAR 4.3.0 (RfC #69054) and IETF RFC 2460 including the forbidding of overlapping fragments according to IETF RFC 5722 and the handling of atomic fragments according to IETF RFC 6946.
  - ▶ **Neighbor discovery protocol (NDP) according to IETF RFC 4861:** This feature facilitates that nodes on the same link discover each other's presence, determine each other's link-layer addresses,



and maintain reachability information about the paths to active neighbors by sending/receiving/processing neighbor solicitation messages and neighbor advertisement messages.

- ▶ **Address Resolution and Neighbor Unreachability Detection (NDP):** Address Resolution is used to learn the MAC address that corresponds to the IPv6 unicast address. Neighbor Unreachability Detection is used to delete hosts from the neighbor cache that are not reachable anymore.
- ▶ **Defensive Neighbor Solicitation/Advertisement Processing:** NDP silently discards all received Neighbor Advertisements that were not requested by a previously transmitted Neighbor Solicitation.
- ▶ **Internet control message protocol version 6 (ICMPv6) according to IETF RFC 4443:** ICMPv6 provides error reporting functionality by sending/receiving/processing error messages, i.e. destination (port) unreachable, parameter problem as well as diagnostic functionality (echo request and echo reply).
- ▶ **Dynamic host configuration protocol version 6 (DHCPv6) client functionality according to IETF RFC 3315:** The DHCPv6 client functionality provides means to dynamically obtain configuration parameters, such as IP addresses, for interfaces and services from a DHCPv6 server.

Hereby the following types of DHCPv6 messages are sent by the client:

- ▶ Solicit messages, to discover DHCPv6 servers
- ▶ Request messages, to request the provision of address and configuration information
- ▶ Renew messages, to extend the lifetime of addresses obtained earlier
- ▶ Rebind messages, to rebind to those addresses if a previous renew message was not answered

The following DHCPv6 messages are received/processed by the client: Advertise messages (sent by the server as a response to discover messages) and reply messages (sent as answers to request, renew, and rebind messages).

The DHCPv6 client functionality includes support for the fully qualified domain name (FQDN) option (excluding DNS support) according to IETF RFC 4704.

- ▶ **Configurable custom DHCP options:** Support the configuration of custom DHCP options that are readable/writable via `TcpIp_DhcpV6Read/WriteOption()`.
- ▶ **IPv6 extension header filter:** Define a white list for the filtering of IPv6 extension headers, i.e. frames containing IPv6 extension headers not listed shall be silently dropped.
- ▶ **IPv6 duplicate address detection:** Detect incorrect (duplicate) IPv6 address configuration.
- ▶ **Router and prefix discovery**
- ▶ **Next hop determination**
- ▶ **User datagram protocol (UDP):** Implementation of the user datagram protocol (UDP) according to IETF RFC 768 including the UDP-related requirements specified by IETF RFC 1122 sections 4.1.3.1 (ports), 4.-1.3.4 (UDP checksums), and 4.1.3.6 (invalid addresses).



- ▶ **Transmission control protocol (TCP):** Implementation of the transmission control protocol (TCP) according to IETF RFC 793 including the TCP-related requirements specified by IETF RFC 1122 sections 4.2.2.-3 (window size), 4.2.2.5 (TCP options), 4.2.2.6 (maximum segment size option), 4.2.2.7 (TCP checksum), 4.2.2.9 (initial sequence number selection), 4.2.2.10 (simultaneous open attempts), 4.2.2.11 (recovery from old duplicate SYN), 4.2.2.13 (closing a connection, excluding *half-duplex close*), 4.2.2.15 (retransmission timeout), 4.2.2.16 (managing the window), 4.2.2.17 (probing zero windows), 4.2.2.18 (passive OPEN calls), 4.2.2.19 (time to live), 4.2.3.2 (when to send an ACK segment (delayed ACK)), 4.2.3.6 (TCP keep-alives), and 4.2.3.10 (remote address validation) and the Nagle algorithm defined in IETF RFC 896.
- ▶ **TCP option filter:** Define a white list of option filter and discard other options silently.
- ▶ **Measurement data support for discarded Ethernet frames:** Implementation of counters for frames/messages that are dropped on different layers of the IP stack. Whenever an Ethernet frames is discarded, a counter is increased. There are independent counters per reason for discarding the frame. The counters can be read and reset via the `<Mod>_GetAndResetMeasurementData()` API defined in AUTOSAR 4.-3.0 that can be accessed via diagnostics. This functionality is supported in Tcplp, SoAd and EthIf.
- ▶ **Buffering of out-of-order segments:** A buffer stores segments that are received out-of-order. In this way, only single lost segments have to be retransmitted instead of all data since the last acknowledged segment. This is a prerequisite to enable the fast retransmission feature on the sender side. This feature also resolves integration issues with TCP/IP communication with Linux.
- ▶ **Release unused socket connections:** A timeout can be configured to release socket connections that are not in use anymore.
- ▶ **Window update transmission in ACK:** Implementation according to IETF RFC 1122 section 4.2.3.3 ("When to Send a Window Update").
- ▶ **Infinite ARP cache timeout:** Infinite timeouts can be configured for ARP table entries to support the use case where no ARPs should be sent at all after initial setup of the network topology and addresses. ARP table entries may time out if only unidirectional communication takes place. ARP table entries that have an infinite timeout are set as fixed, so that no timers run for them, which also saves resources.
- ▶ **AUTOSAR network management coordination algorithm:** The AUTOSAR network management algorithm is realized by the transmission of periodic network management messages as long as the bus communication is requested. The reception of network management messages signals that other nodes request bus communication.

The IP stack network management supports the following features:

- ▶ **Support operational modes:** Support of operational modes Network Mode (with internal states Repeat Message State, Normal Operation State, Ready Sleep State), Prepare Bus-Sleep Mode, and Bus-Sleep Mode according to AUTOSAR specifications.
- ▶ **Support configurable structure of NM messages:** Support for configurable structure of NM messages according to AUTOSAR specification.
- ▶ **Support for communication startup:** Support for interface to the upper layer to initiate transmission of NM messages due to any user(s) requesting communication.



- ▶ **Support for communication shutdown:** Support for interface to the upper layer to stop transmission of NM messages due to any user(s) not requesting communication.
- ▶ **Support for communication passive wakeup:** Support for interface to the upper layer to initiate communication capabilities due to a wakeup event network start or network restart indication.
- ▶ **Support for passive mode:** Support for nodes with transmission of network management messages disabled.
- ▶ **Support for detection of remote sleep:** Support for detecting if all other nodes are ready to sleep.
- ▶ **Support for state change notification:** Support for notification function for `Nm` when the `UdpNm` state changes.
- ▶ **Support for car wakeup:** Support of car wakeup bit as part of the network management message and car wakeup callout function.
- ▶ **Support for user data in NM messages:** Support for updating user data in NM messages via `UdpNM` interfaces or the communication stack by collecting the data from a configured I-PDU.
- ▶ **Support for PDU length higher than 8 bytes:** Support payloads higher than 8 bytes of the NM messages if supported by the bus type.
- ▶ **Support for communication control:** Support for interfaces to enable/disable transmission of NM messages.
- ▶ **Support for partial networking:** Support for updating and filtering partial network information as part of the NM messages.
- ▶ **Support for spontaneous transmission:** Support for interface to trigger spontaneous transmission of an NM message with the provided NM user data.
- ▶ **Support for immediate transmission:** Support for transmission of a predefined number of NM messages with a different cycle time when entering the Repeat Message state from Bus Sleep state or Prepare Bus-Sleep state.
- ▶ **Support for immediate restart:** Support for transmission of NM messages when the network was requested in the Prepare Bus-Sleep state.
- ▶ **Support of RepeateMsgInd|NodeDetection|NodIdEnabled channel-based configurable:** Support the per channel configuration of parameters `UdpNmRepeatMsgIndEnabled`, `UdpNmNodeDetectionEnabled`, `UdpNmNodeIdEnabled`.
- ▶ **Support for BSW distribution:** Support for inter-core communication when different bus types are processed on different cores.
- ▶ **Support for post-build:**
  - ▶ Support for handling post-build loadable and selectable configuration in Sd, SoAd, TcpIP, and UdpNm.
  - ▶ Support for post-build loadable configuration in EthIf, EthSm, and DoIP.
- ▶ **DoIP support:** Support for Diagnostic over IP (DoIP) according to AUTOSAR 4.1.3., including:



- ▶ **Vehicle network integration:** Handling of IP address assignment via the SoAd and the Tcplp module.
- ▶ **Vehicle announcement and vehicle discovery:** Sending of messages containing the vehicle identification number (VIN), the IP address, and the logical diagnostic address as a response to vehicle discovery requests of the external testing device (vehicle discovery responses) or upon completion of the IP address assignment (vehicle announcement).
- ▶ **Routing activation:** Activation of routing/relaying of diagnostic messages (including support for authentication and confirmation) from the external testing device to the Diagnostic Communication Manager (Dcm) module of a remote ECU connected to vehicle network and vice versa.
- ▶ **Diagnostic message relaying:** Relaying of diagnostic messages between the external tester and the ECU's local PDU Router (PduR) module. Depending on the logical diagnostic destination address, the final recipient of these diagnostic messages is the ECU's local Dcm module, the Dcm module of a remote ECU connected to vehicle network, or the external testing device.
- ▶ **DolP entity status information:** Provision of status information on the DolP entity, i.e. the ECU itself. This status information includes the type of the DolP entity (DolP node vs. DolP gateway), the maximum number of concurrent DolP connections, the number of currently open DolP connections, and the maximum size of one diagnostic request that can be processed by the DolP entity.
- ▶ **Service component interfaces:** Provision of service component interfaces in addition to callback functions.
- ▶ **Inactivity time-out for TCP connections:** Support of inactivity time-outs to free TCP connections of an inactive tester.
- ▶ **Parallel testers:** Support of parallel testers.
- ▶ **GID synchronization:** Support group identifier (GID) synchronization according to AUTOSAR and EB-specific extensions.
- ▶ **Configuration of EID/GID:** Support static configuration of EID (entity identifier) and GID. Support usage of EID as GID. Support usage of MAC address as EID.
- ▶ **Alive check:** Support of alive check to free inactive tester connection.
- ▶ **DolP end nodes:** DolP acts as a proxy to route diagnostic messages in DolP format and DolP acknowledgements between tester and DolP end nodes.
- ▶ **Support OEM-specific custom payload types:** DolP routes OEM-specific custom payload types to be processed by complex device drivers.
- ▶ **Detect connected tester before ECU switches to sleep mode:** If the ECU is connected to a tester (e.g. only laptop) via Ethernet and no NM is simulated, it is possible to detect the presence of tester already before diagnostic communication is established to keep the ECU awake. Rationale: The ECU might switch to sleep mode faster than it takes to establish diagnostic communication.
- ▶ **Reject insecure connection requests:** DolP shall reject connection requests which are received via an insecure connection. Implementation according to AUTOSAR RFC AR-3349.
- ▶ **Support of Service Discovery:** Support of Service Discovery according to AUTOSAR 4.2.1., including:



- ▶ **Service availability management:** Management of the availability of functional entities called services in the in-vehicle network.
  - ▶ **Announcement/offer of services (server role):** Provide offers of services, withdraw offers of services, and respond to explicit service discovery/find requests.
  - ▶ **Discovery/find of services (client role):** Listen for provided service offers, listen for withdrawal of service offers, send explicit service discovery/find requests.
- ▶ **Event subscription:** Management of the subscription for events created by a service.
  - ▶ **Publishing of events (server role):** (Periodic) provision of a group of events if there is at least one subscriber via the transmission of event messages.
  - ▶ **Subscription of events (client role):** Subscription to a group of events generated by a particular service.
- ▶ **Transmission path handling:** Dynamic management of transmission paths to the AUTOSAR SoAd module to reduce the average bandwidth consumption.
  - ▶ Setup and removal of transmission paths (UDP/TCP based on IPv4/IPv6) for service availability management and event subscription.
  - ▶ Selection of unicast or multicast communication (UDP based on IPv4/IPv6) for the publishing of events based on the number of subscribers.
- ▶ **Seamless service relocation:** Support of seamless relocation of servers providing services (i.e., without needing to update the configuration of the clients) and the seamless adding/relocation of clients consuming services (i.e., without needing to update the configuration of the servers).
- ▶ **Multicast-only event groups:** Support for multicast-only event groups on client and server side.
- ▶ **Efficient startup of SOME/IP-SD:** At startup, Service Discovery queries the underlying IP stack to determine if it is initialized. SOME/IP-SD messages are buffered until Tcplp is ready for transmission, i.e., until the MAC address is known and security associations are established in case of IKEv2. With this mechanism, no messages are lost and no IP packet queue is needed. SOME/IP-SD messages are sent without delay as soon as Tcplp is ready for transmission.
- ▶ **DEM error reporting:** Support for reporting the following DEM events in case of errors:

<b>DEM event</b>	<b>Condition</b>
SD_E_MALFORMED_MSG	Malformed message received (SWS_SD_00707)
SD_E_OUT_OF_RES	Out of resources (SWS_SD_00707)
SD_E_SUBSCR_NACK_RECV	Negative acknowledge received (SWS_SD_00707)
SD_E_INVALID_ID	Invalid server service ID received (extension to AUTOSAR)

DEM events can also be disabled or reported to DET.

## 3. ACG8 IP Stack release notes

### 3.1. Overview

This chapter provides the ACG8 IP Stack 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 IP Stack release.

Module name	AUTOSAR version and revision	SWS version and revision	Module version	Supplier
<a href="#">DoIP</a>	4.1.3 []	4.1.3 [0000]	1.1.22	Elektrobit Automotive GmbH
<a href="#">EthIf</a>	4.3.0 []	4.3.0 [0000]	1.9.19	Elektrobit Automotive GmbH
<a href="#">EthSM</a>	4.3.0 []	4.3.0 [0000]	1.6.14	Elektrobit Automotive GmbH
<a href="#">Sd</a>	4.2.1 []	4.2.1 [0000]	1.4.11	Elektrobit Automotive GmbH
<a href="#">SoAd</a>	4.2.2 []	4.2.2 [0000]	1.8.18	Elektrobit Automotive GmbH



Module name	AUTOSAR version and revision	SWS version and revision	Module version	Supplier
<a href="#">SomelpTp</a>	4.3.0 []	4.3.0 [0000]	1.0.30	Elektrobit Automotive GmbH
<a href="#">Tcplp</a>	4.3.0 []	4.3.0 [0000]	3.5.14	Elektrobit Automotive GmbH
<a href="#">UdpNm</a>	4.1.3 []	3.3.0 [3]	2.9.8	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
No EB modules available		

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. DoIP module release notes

- ▶ AUTOSAR R4.1 Rev 3

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



- ▶ AUTOSAR SWS document version: 4.1.3
- ▶ Module version: 1.1.22.B466224
- ▶ Supplier: Elektrobit Automotive GmbH

### 3.3.1.1. Change log

This chapter lists the changes between different versions.

#### Module version 1.1.22

2021-10-08

- ▶ Implemented tester-friendly TCP connection reset.

#### Module version 1.1.21

2021-06-25

- ▶ ASCDOIP-698 Fixed known issue: Wrong adjacent layer information causes generation error if DoIPCustomPayloadTypeEnabled is set to TRUE.

#### Module version 1.1.20

2021-03-05

- ▶ Implemented post-build selectable variant handling.

#### Module version 1.1.19

2020-10-23

- ▶ Implemented DoIP event callback product enhancement.

#### Module version 1.1.18

2020-08-28

- ▶ Implemented configurable diagnostic messages response handling before connection is in ISO 13400 connection state "Registered[Routing Active]".



- ▶ ASCDOIP-659 Fixed known issue: DoIP custom payload type messages are responded with ACK/NACK messages that may be malformed.

#### **Module version 1.1.17**

2020-06-19

- ▶ ASCDOIP-579 Fixed known issue: Custom Payload feature: reception of segmented custom payload messages handled incorrectly.
- ▶ Implemented optional rejection of routing activation request message for unsecure connections.
- ▶ Improved timeout handling.
- ▶ Implemented counters for frames dropped due to firewall rule or consistency check.
- ▶ Implemented DHCP Host name option with VIN.

#### **Module version 1.1.16**

2020-02-21

- ▶ ASCDOIP-558 Fixed known issue: PduR\_DoIPTpRxIndication() called for diagnostic message initially rejected by PduR.
- ▶ ASCDOIP-562 Fixed known issue: Integration requirement missing that preemptions of IP stack mainfunctions is not allowed.
- ▶ Implemented support of custom payload types.

#### **Module version 1.1.15**

2019-10-11

- ▶ ASCDOIP-514 Fixed known issue: Reception of UDP frame with length in header bigger than received length may cause an out-of-bounds read access.
- ▶ Implemented user callback function to indicate received vehicle identification requests.

#### **Module version 1.1.14**

2019-06-14

- ▶ ASCDOIP-507 Fixed known issue: Invalid routing activation message response code in case of automatic routing activation type.
- ▶ ASCDOIP-502 Fixed known issue: DoIP routes diagnostic messages to the wrong TCP connection.



### Module version 1.1.13

2019-04-17

- ▶ Improved procedure of Alive check response message drop in case Tcp connection is not registered.

### Module version 1.1.12

2019-04-11

- ▶ Implemented check for supported Protocol Versions.
- ▶ ASCDOIP-489 Fixed known issue: Despite an incomplete diagnostic message, a positive notification is sent to the upper layer.
- ▶ Changed behavior of Initial Inactivity Timer and General Inactivity Timer.
- ▶ ASCDOIP-503 Fixed known issue: UDP connection locked to remote address if no response is sent.

### Module version 1.1.11

2019-02-15

- ▶ ASCDOIP-472 Fixed known issue: DoIP incorrectly measures DoIPInitialVehicleAnnouncementTime and DoIPVehicleAnnouncementInterval.
- ▶ ASCDOIP-479 Fixed known issue: Incorrect tester source address is set in routing activation NACK message if an already registered connection is used.

### Module version 1.1.10

2018-12-03

- ▶ ASCDOIP-459 Fixed known issue: Connection is closed when 2 DoIP messages are received in the same TCP frame.

### Module version 1.1.9

2018-10-26

- ▶ Changed call context of SoAd\_ReleaseRemoteAddr() to DoIP\_IfTxConfirmation to enable use of UDP retry.
- ▶ Implemented Service Interface for application SWCs.
- ▶ Changed multiplicity of DoIPTargetAddress and DoIPTargetAddressRef to 65535.



- ▶ ASCDOIP-443 Fixed known issue: Missing critical section in DoIP\_HandleTxDiagMsgConfirmation due to a race condition.
- ▶ ASCDOIP-416 Fixed known issue: Support of PduR\_DoIPIfTxConfirmation() is missing in AdjacentLayer.properties if feature custom payload support is enabled.
- ▶ Implemented Post-build binary support.

#### **Module version 1.1.8**

2018-08-06

- ▶ ASCDOIP-412 Fixed known issue: Reception of diagnostic message with payload type = 0xF001 generates generic header NACK message and closes TCP connection.
- ▶ Added support of DoIPUseMacAddressForIdentification.
- ▶ Added EB specific GID synchronization support via callout functions.

#### **Module version 1.1.7**

2018-06-22

- ▶ ASCDOIP-381 Fixed known issue: Alive check response deactivates DoIPChannels.
- ▶ ASCDOIP-384 Fixed known issue: SoAd\_TpTransmit() is called for a DoIPChannel that did not receive a transmit request.
- ▶ ASCDOIP-387 Fixed known issue: PduR\_DoIPStartOfReception is called multiple times during diagnostic message reception.
- ▶ Added support for uint32 PduLengthType
- ▶ Added DoIP\_MainFunctionTx() to trigger Tx processing in addition to DoIP\_MainFunction().

#### **Module version 1.1.6**

2018-05-29

- ▶ ASCDOIP-379 Fixed known issue: Max data size(MDS) cannot be disabled for Diagnostic entity status response message.

#### **Module version 1.1.5**

2018-05-25

- ▶ Separated the transmit part of DoIP\_MainFunction to DoIP\_MainFunctionTx and added an option to make the latter externally callable.



- ▶ ASCDOIP-368 Fixed known issue: DoIP fails to transmit diagnostic acknowledge message when DoIP-NumByteDiagAckNack > 0.
- ▶ ASCDOIP-369 Fixed known issue: Alive check request resets Rx state machine.
- ▶ Changed to provide both header and payload when DoIP\_SoAdTpCopyTxData() is called.

#### **Module version 1.1.4**

2018-05-14

- ▶ ASCDOIP-355 Fixed known issue: Set RxState to READY after reception of Alive check response message.

#### **Module version 1.1.3**

2018-05-07

- ▶ Added support for Alive check.

#### **Module version 1.1.2**

2018-04-20

- ▶ ASCDOIP-330 Fixed known issue: DoIP\_SoConModeChg() is not called due to incorrect configuration in DoIP\_AdjacentLayer.properties file.
- ▶ ASCDOIP-331 Fixed known issue: Generator sets RoutingActivationIdx to 0 for all routes regardless of configured DoIPRoutingActivationNumber.
- ▶ ASCDOIP-332 Fixed known issue: DoIP RX side blocked forever when response message is transmitted in the context of DoIP\_SoAdTpTxConfirmation() and SoAd\_TpTransmit() returns E\_NOT\_OK.
- ▶ Added limit to DoIPMaxRequestBytes based on PduLengthType.
- ▶ ASCDOIP-337 Fixed known issue: DoIP does not support more than 256 routes per single Tester.
- ▶ ASCDOIP-308 Fixed known issue: TCP connection may get closed if TCP fragmentation buffer reaches its capacity.
- ▶ ASCDOIP-334 Fixed known issue: DoIP RX side loses synchronization with RX data stream when SoAd\_TpTransmit() returns E\_NOT\_OK during transmission of response message.

#### **Module version 1.1.1**

2018-03-16



- ▶ ASCDOIP-294 Fixed known issue: DoIP does not send positive routing activation response message (0x10) after authentication callback returns E\_OK after E\_PENDING.

#### **Module version 1.1.0**

2018-02-16

- ▶ Added support for proper handling of limited broadcast address.
- ▶ Improved DoIP\_BufferIdType size depended on a maximum number of configured buffers.
- ▶ ASCDOIP-280 Fixed known issue: Default PduR expected lower layer If API does not match to DoIP\_-IfTransmit().

#### **Module version 1.0.18**

2018-01-19

- ▶ Added support for manual routing activation of custom channels.

#### **Module version 1.0.17**

2017-12-15

- ▶ Changed NACK feature to ignore incoming NACK and Vehicle announcement messages.
- ▶ Added support for manufacturer-specific DoIP payload types and automatic routing activation.

#### **Module version 1.0.16**

2017-11-17

- ▶ Added support for routing activation authentication and confirmation.
- ▶ Added support to trigger transmission in context of TxConfirmation() of the same PDU.
- ▶ Added discard of diagnostic message in case PduR is not able to receive it.

#### **Module version 1.0.15**

2017-09-22

- ▶ Added support for diagnostic power mode callback.



- ▶ Added support for a transmission of diagnostic message with a maximum size of 64k payload + DoIP headers.
- ▶ Replaced SoAd\_SetRemoteAddr(wildcard) with SoAd\_ReleaseRemoteAddr().
- ▶ Updated Misra metrics to 2012 version.
- ▶ ASCDOIP-181 Fixed known issue: Diagnostic acknowledge message contains superfluous bytes if received diagnostic message is smaller than DoIPNumByteDiagAckNack.

#### **Module version 1.0.14**

2017-08-25

- ▶ Added workaround for Dcm\_GetVin.

#### **Module version 1.0.13**

2017-07-28

- ▶ Added configurable use of MaxDataSize field and Diagnostic entity status information provisioning.

#### **Module version 1.0.12**

2017-06-30

- ▶ Added segmented reception of DoIP headers.

#### **Module version 1.0.11**

2017-06-02

- ▶ Removed configuration check that disabled N:1 PDU routing feature.
- ▶ Updated DoIP documentation according to new build rules.
- ▶ Added configuration parameter DoIPMaxNumByteDiagAckNack to limit the maximum value of DoIPNumByteDiagAckNack.

#### **Module version 1.0.10**

2017-05-05

- ▶ Added support of DoIP NACK feature.



### Module version 1.0.9

2017-03-31

- ▶ Added default configuration values.
- ▶ Updated DoIP\_SoAdTpCopyTxData() syntax to be backward compatible with AUTOSAR 4.0.3.
- ▶ Added handle ID wizard support for DoIPUdpVehicleAnnouncement\DoIPSoAdTxPduId.

### Module version 1.0.8

2017-03-03

- ▶ Added handle ID wizard support for DoIPPduR[Rx|Tx]PduId.

### Module version 1.0.7

2017-02-03

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

### Module version 1.0.6

2016-12-02

- ▶ ASCDOIP-85 Fixed known issue: Diagnostic message acknowledgment sent to wrong target address.

### Module version 1.0.5

2016-11-04

- ▶ Updated local IP address assignment according to AUTOSAR 4.2.2.
- ▶ Changed the XPath check for the parameter `DoIPSoAdTxPduId` in the container `DoIPUdpVehicleAnnouncement`.
- ▶ ASCDOIP-78 Fixed known issue: Incorrect error message if no `DoIPUdpVehicleAnnouncement` container is present.

### Module version 1.0.4

2016-05-26

- ▶ ASCDOIP-71 Fixed known issue: Routing activation response code is wrong.



### Module version 1.0.3

2015-11-06

- ▶ Added transmission serialization.
- ▶ ASCDOIP-63 Fixed known issue: Different tester using the same route activation number and same target leads to invalid configuration structure.

### Module version 1.0.2

2015-06-19

- ▶ Added task auto assign of `DoIP_MainFunction()` for RTE.

### Module version 1.0.1

2015-02-20

- ▶ ASCDOIP-34 Fixed known issue: Compilation of DoIP module fails.
- ▶ ASCDOIP-37 Fixed known issue: Permanent sending of same diagnostic acknowledgments.
- ▶ ASCDOIP-35 Fixed known issue: DoIP is not able to receive or transmit data via TCP.
- ▶ ASCDOIP-39 Fixed known issue: The "reserved by ISO" bytes in the routing activation response frame are not zero.
- ▶ ASCDOIP-41 Fixed known issue: DoIP does not send vehicle identification announcements.
- ▶ ASCDOIP-42 Fixed known issue: The call of `DoIP_SoAdIfTxConfirmation()` for vehicle ID announcement reports `DOIP_E_INVALID_PDU_SDU_ID` to Det.
- ▶ ASCDOIP-46 Fixed known issue: Closing of a TCP connection causes invalid behavior
- ▶ ASCDOIP-47 Fixed known issue: Routing activation request containing a not configured source address causes an invalid memory access.

### Module version 1.0.0

2014-10-02

- ▶ Initial version.

#### 3.3.1.2. New features

- ▶ Add a feature to optionally close Tcp connection with FIN.



### 3.3.1.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- ▶ DoIP provides `DoIPMaxNumByteDiagAckNack` to limit the maximum value of `DoIPNumByteDiagAckNack`

Description:

If the value of `DoIPTester/DoIPNumByteDiagAckNack` is greater than `DoIPMaxNumByteDiagAckNack`, the number of bytes of the original Diagnostic message that is copied into the response message is limited to `DoIPMaxNumByteDiagAckNack`.

- ▶ DoIP provides a possibility to configure a callback function `User_DoIPGetVin()` to retrieve VIN

Description:

When you configure a callback function in `General/DoIPGetVIN/DoIPGetVin`, then this function is called to copy the vehicle identification number (VIN) to a specified memory location. This is a workaround for `Dcm_GetVin()`.

A VIN retrieved this way is used in the following cases: during verification of a Vehicle identification request message with VIN, when vehicle identification response/vehicle announcement messages are generated and optionally when `SoAd_WriteDhcpHostNameOption` is called.

- ▶ DoIP provides configuration parameter `DoIPChannel/DoIPRoutingActivationType` which define how routing activation is performed for a specified channel.

Description:

When `DoIPRoutingActivationType` is set to `DOIP_ROUTING_ACTIVATION_AUTOMATIC`, then it is not required to send routing activation message prior exchanging diagnostic messages.

When `DoIPRoutingActivationType` is set to `DOIP_ROUTING_ACTIVATION_MSG`, then the reception of routing activation message is a precondition for exchanging diagnostic messages.

`DolPTester/DolPRoutingActivationRef` lower multiplicity is changed from 1 to 0 for testers that are referenced by channels with automatic routing activation.

- ▶ DoIP provides configuration parameter `DoIPCustomPayloadTypeEnabled` to enable support for custom payload types.

Description:

This config parameter enables Custom payload type feature - handling of diagnostic messages with payload types in range `0xF000..0xFFFF` which is reserved for manufacturer specific use. If a DoIP frame with payload type in the reserved range is received, then the generic header shall be stripped away, `SoCnId` and `PayloadType` shall be set for Rx PDU in corresponding custom payload type container belonging to the TCP connection and the message payload shall be passed to the PduR via TP API. If a Tx PDU



with a custom payload type is requested for sending via TP API, PayloadType is retrieved from Tx Pdu meta data and added to the generic header. Then this PDU is sent or queued equal to any other PDU from a standard DoIP channel.

Custom diagnostic messages are routed via custom channels defined in `DoIPCustomChannel` container, with the following parameters:

`DoIPTcpConnectionRef` - Reference to Tcp connection used for transmission of custom diagnostic messages

`DoIPPduRRxPduId` - Values should be consecutive after `DoIPPduRRxPdulds` from `DoIPChannel`

`DoIPPduRRxPduRef` - Reference to Rx PDUs to provide Meta data items `SoConId` and `PayloadType`

`DoIPPduRTxPduId` - Values should be consecutive after `DoIPPduRTxPdulds` from `DoIPChannel`

`DoIPPduRTxPduRef` - Reference to Tx PDUs to retrieve Meta data items `SoConId` and `PayloadType`

- ▶ `DoIP` provides a `DoIPGetAndResetMeasurementDataApi` which enables `DoIP_GetAndResetMeasurementData` API i.e. reading and resetting of measurement data for diagnostic purposes. Used measurement indexes: - `DOIP_MEAS_DROP_GENHDR` - for number of dropped messages due to generic header errors - `DOIP_MEAS_DROP_DIAGMSG` - for number of dropped diagnostic messages
- ▶ `DoIP` implements proper handling of limited broadcast address according to [https://www.autosar.org/bugzilla/show\\_bug.cgi?id=74847](https://www.autosar.org/bugzilla/show_bug.cgi?id=74847).
- ▶ `DoIP` provides a `DoIPEnableMainFunctionTx` which enables the transmit section of `DoIP_MainFunction` - `DolPEnableMainFunctionTx` to be externally callable.
- ▶ `DoIP` provides an extension to Autosar handling of `EID` of a vehicle identification response/vehicle announcement message by using `DoIPEid` if this config param is set, or `DoIPGIDInvalidityPattern` if not, instead of the MAC address received via `SoAd_GetPhysAddr`, in case when `SoAd_GetPhysAddr` returns `NOT_OK`.
- ▶ `DoIP` provides method for Dynamic GID master/slave selection and related GID synchronization between DoIP entities.

Description:

The feature is enabled by configuring user callback functions `User_DoIPGetGID()` and `User_DoIPDynamicGIDMasterSelection()`.

The feature replaces AUTOSAR GID Synchronization process due to problems identified in [https://bugzilla.autosar.org/show\\_bug.cgi?id=80770](https://bugzilla.autosar.org/show_bug.cgi?id=80770).

The process consists of two parts:



1) Calling `User_DoIPDynamicGIDMasterSelection()` during `DoIP_Init()` to obtain information whether DoIP entity is GID master or slave. In case DoIP entity is GID master, the process of Dynamic GID Synchronization is completed.

2) In case DoIP entity is GID slave, `User_DoIPGetGID()` is called in the context of `DoIP_MainFunction()` to obtain GID. This is done until `User_DoIPGetGID()` returns `E_OK`. Only then GID Synchronization process is completed for that DoIP entity.

- ▶ DoIP supports up to 65535 DoIPTargetAddress, i.e. `DoIPTargetAddress` multiplicity has been changed from 1..255 to 1..65535 according to RfC 79727.

DoIP supports up to 65535 `DoIPTargetAddressRef` in `DoIPRoutingActivation`, i.e. `DoIPTargetAddressRef` multiplicity has been changed from 1..255 to 1..65535 according to RfC 79727.

- ▶ In order to define in more detail the behavior of Initial and General inactivity timers, the following clarification are given:

1) Initial inactivity timer shall be stopped only after a valid Routing activation request message is received (SA is registered to a Tcp connection without taking into account authentication/confirmation phase).

2) Reception of Diagnostic message (or any DoIP message other then RA) would not affect Initial inactivity timer.

3) General inactivity timer would be started only after the step given in 1).

4) After routing activation has been performed General inactivity timer is reset under any of the following conditions:

- Reception of a new valid routing activation message
- Reception of a valid diagnostic message, i.e. message that generates positive acknowledge
- Reception of a valid custom payload type message
- Reception of a valid diagnostic message that was rejected by UL due to insufficient buffers, which resulted in diagnostic NACK message (0x05)
- Reception of a valid Alive check response (received Source address matches registered)
- Call of `SoAd_TpTransmit()` as a result of `DoIP_TpTransmit()` - transmission initiation
- Transmission of a each diagnostic message segment (`DoIP_SoAdTpCopyTxData`)
- Transmission of a each custom diagnostic message segment (`DoIP_SoAdTpCopyTxData`)

- ▶ On reception of UDP DoIP requests the remote address and port of the corresponding UDP connection gets released by calling `SoAd_ReleaseRemoteAddr()` 1) in context of `DoIP_SoAdIfTxConfirmation()` if a response was transmitted. This is necessary to support UDP retry functionality provided by



SoAd. 2) immediately if no response was transmitted or transmission failed. 3) after a deadline in case of `DoIP_SoAdIfTxConfirmation()` was not called.

- ▶ If the DoIP module receives a valid diagnostic message and the according "Source address" is registered at another Tcp connection the DoIP module will send a diagnostic message negative acknowledge message with the diagnostic message negative acknowledge code set to 0x06 (Route inactive). Additionally the message will be discarded.

This is valid for both AUTOMATIC and MANUAL routing activation.

Note: Valid diagnostic message here means that DoIP channel exist with configured Source and Target address that match the received values from Diagnostic message header.

- ▶ DoIP provides an optional user defined callback function to indicate received vehicle identification requests. The functionality can be configured with config parameter `DoIPGeneral/DoIPVIDRequestReceivedCallback/DoIPVIDRequestReceived` and a user defined header file can be configured with `DoIPHeaderFileInclusion`.
- ▶ DoIP provides a `DoIPDhcpHostNamePrefix` which enables the addition of the vendor specific name if required. If parameter `DoIPDhcpHostNamePrefix` is not an empty string it will be added to the DHCP host name after DoIP- and prior to VIN when `DoIPDhcpOptionVinUse` parameter is enabled. According to SWS this parameter shall be filled with "VIN", but it can also be filled with vendor specific value.
- ▶ DoIP provides configurable diagnostic messages response handling before connection is in ISO 13400 connection state "Registered[Routing Active]". The functionality can be configured with config parameter `DoIPGeneral/ResponseBeforeRoutingActivation`.
- ▶ DoIP provides optional feature to call a list of user provided callback functions after routing activation has been successfully performed.

Callback functions have the following syntax: (void) User\_RoutingActivationCallback (SoAd\_SoConIdType SoConId, uint16 SourceAddr, uint8 ActivationType, const uint8\* OemSpecificPtr);

- ▶ DoIP supports post-build variant handling for the following parameters: DoIPEid, DoIPLogicalAddress and DoIPTargetAddressValue.
- ▶ DoIP can close and reset TCP connections in a two ways.

As a "Soft" closure where both sender and receiver agree on closing the session (closure with FIN flag) by calling `SoAd_CloseSoCon` with Abort = FALSE. This mechanism is used whenever DoIP NACK needs to be sent along with TCP closure. "Soft" closure is optionally enabled by using `DoIPEnableTcpClosureWithFIN` configuration parameter.

As an abrupt TCP connection reset (closure with RST flag). This is accomplished by calling `SoAd_CloseSoCon` with Abort = TRUE. This mechanism is used in ALL cases accept those described in the previous paragraph, e.g. Alive check, Inactivity timeout, `DoIP_ActivationLineSwitchInactive()` etc. When `DoIPEnableTcpClosureWithFIN` is disabled, Tcp connection reset is performed in all cases.



### 3.3.1.4. Deviations

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

- ▶ Different syntax of `DoIP_SoAdStartOfReception()`

Description:

The syntax of `DoIP_SoAdStartOfReception()` of AUTOSAR 4.1 rev 1 is used. Function parameter `const PduInfoType* info` is omitted.

Rationale:

This is a compatibility aspect to stay compatible with actual implementation of `SoAd`.

Requirements:

`SWS_DoIP_00037`

- ▶ Socket connection status not stored

Description:

The module does not store socket connection state of its connections, but it does react on state changes reported with `DoIP_SoConModeChg()` in the following manner: UDP announcement connections: (OFFLINE, RECONNECT) -> ONLINE - Enable sending vehicle announcement messages according to [DoIP.-EB.SWS\_DoIP\_00205]. ONLINE -> (OFFLINE, RECONNECT) - Ignore notification (no functional impact). TCP connections: (OFFLINE, RECONNECT) -> ONLINE - Ignore notification (no functional impact). Connection is registered, and inactivity timer started only with `DoIP_SoAdTpStartOfReception()` implying connection is ONLINE. ONLINE -> (OFFLINE, RECONNECT) - Reset connection according to [DoIP.ASR41.-SWS\_DoIP\_00243]. UDP connections: Ignore all notifications. Udp connection is RX driven. If a message is received, it implies the connection is ONLINE, and response is generated.

Rationale:

Skipping the socket connection status maintenance reduces runtime and RAM consumption.

Requirements:

`SWS_DoIP_00001, SWS_DoIP_00002, SWS_DoIP_00241, SWS_DoIP_00012`

- ▶ `DoIP_ActivationLineSwitch()` replaced and functionally modified

Description:

The function `DoIP_ActivationLineSwitch()` is replaced by the functions `DoIP_ActivationLineSwitchActive()` and `DoIP_ActivationLineSwitchInactive()`. The call of `Rte_Mode_-DoIPActivationLineSwitchNotification_CurrentDoIPActivationLineStatus()` is not



performed. SWS requirements regarding SVC interface SWS\_DoIP\_00242 and SWS\_DoIP\_00265 are replaced with respective EB requirements. Correct handling of limited broadcast address added.

#### Rationale:

See AUTOSAR bugzillas: [https://www.autosar.org/bugzilla/show\\_bug.cgi?id=66944](https://www.autosar.org/bugzilla/show_bug.cgi?id=66944). [https://www.autosar.org/bugzilla/show\\_bug.cgi?id=74847](https://www.autosar.org/bugzilla/show_bug.cgi?id=74847).

#### Requirements:

SWS\_DoIP\_00003, SWS\_DoIP\_00071, SWS\_DoIP\_00203, SWS\_DoIP\_00204, SWS\_DoIP\_00205, SWS\_DoIP\_00234, SWS\_DoIP\_00235, SWS\_DoIP\_00242, SWS\_DoIP\_00251, SWS\_DoIP\_00252, SWS\_DoIP\_00265, SWS\_DoIP\_00271

- ▶ Update of DHCP host name is configurable per connection

#### Description:

When API `DoIP_LocalIpAddrAssignmentChg()` is called with the State set to `TCPIP_IPADDR_STATE_ASSIGNED`, the DoIP module shall check the configuration parameter `DoIPUpdateDhcpHostNameOption` specified for that connection. Only if this parameter is set to TRUE, DoIP shall update DHCP host name option.

#### Requirements:

SWS\_DoIP\_00154

- ▶ Different parameter type for `DoIP_SoAdTpTxConfirmation()` and `DoIP_SoAdTpRxIndication()`

#### Description:

The type of the function parameter `result` is `NotifResultType` instead of `Std_ReturnType` for `DoIP_SoAdTpTxConfirmation()` and `DoIP_SoAdTpRxIndication()`. No parameter check is performed for this function parameter.

#### Rationale:

This is a compatibility aspect to stay compatible with actual implementation of SoAd. The function parameter `result` is not processed.

#### Requirements:

SWS\_DoIP\_00032, SWS\_DoIP\_00038, SWS\_DoIP\_00182, SWS\_DoIP\_00192

- ▶ Function parameter `SoConId` not validated

#### Description:

The function parameter `SoConId` of function `DoIP_LocalIpAddrAssignmentChg()` and `DoIP_SoConModeChg()` is not validated.



Rationale:

DoIP does not maintain a overview of the socket connection IDs. It gets the SoConId from the SoAd and will use this parameter only for API calls to SoAd again. Wrong SoConId will be detected by subsequent SoAd API calls.

Requirements:

SWS\_DoIP\_00194, SWS\_DoIP\_00196

- ▶ DoIPVinInvalidityPattern used for vehicle identification number

Description:

The function Dcm\_GetVin() is not called. Instead DoIPVinInvalidityPattern is used for vehicle identification number.

Requirements:

SWS\_DoIP\_00070, SWS\_DoIP\_00072, SWS\_DoIP\_00156

- ▶ The features cancel transmit and cancel receive are not supported

Description:

The functionality for the APIs DoIP\_CancelTransmit() and DoIP\_CancelReceive() is not supported. The APIs return E\_NOT\_OK.

Requirements:

SWS\_DoIP\_00023, SWS\_DoIP\_00024, SWS\_DoIP\_00257, SWS\_DoIP\_00258

- ▶ Different syntax for DoIP\_SoAdTpCopyTxData()

Description:

The API DoIP\_SoAdTpCopyTxData() does not contain a const for the parameter PduInfoPtr.

Rationale:

SoAd expects AUTOSAR 4.0.3 syntax for this API.

Requirements:

SWS\_DoIP\_00031

- ▶ Diagnostic message negative acknowledge codes clarification

Description:



Interpretation of the requirements SWS\_DoIP\_00126, SWS\_DoIP\_00174, SWS\_DoIP\_00216 is the following:

Negative acknowledge code is set to 0x08 (according to SWS\_DoIP\_00174) in case upper layer returns error code BUFREQ\_E\_NOT\_OK or BUFREQ\_E\_BUSY. Negative acknowledge code is set to 0x05 (according to SWS\_DoIP\_00126) in case upper layer returns error code BUFREQ\_E\_OVFL.

Requirements:

SWS\_DoIP\_00216

- ▶ Lower multiplicity set to 0

Description:

Interpretation of the requirements SWS\_DoIP\_00126, SWS\_DoIP\_00174, SWS\_DoIP\_00216 is the following:

Lower multiplicity set to 0 for ..//DoIPTester/DoIPRoutingActivationRef due to introduction of automatic routing activation.

Requirements:

ECUC\_DoIP\_00034\_Conf, ECUC\_DoIP\_00062\_Conf

- ▶ DoIP does not use internal buffers for UDP communication.

Description:

When SoAd calls the DoIP module via the Interface DoIP\_SoAdIfRxIndication, DoIP module does not copy the message into the internal UDP buffer for further processing, instead it does all the message processing within a context of DoIP\_SoAdIfRxIndication, using message buffer originally provided with a function call.

Consequently when DoIP\_SoAdIfTxConfirmation is called no buffers need to be released.

Requirements:

SWS\_DoIP\_00197, SWS\_DoIP\_00199

- ▶ DoIP only supports streaming mode for Tp communication with SoAd

Description:

The parameter TpSduLength of DoIP\_SoAdTpStartOfReception() needs to be set to 0, because DoIP supports only streaming mode of Tp communication with SoAd.

Requirements:



### SWS\_DoIP\_00018

- ▶ DoIP sets output parameter AvailableDataPtr of DoIP\_SoAdCopyTxData() to the size of the complete available diagnostic message

Description:

When the function DoIP\_SoAdCopyTxData() is called for the use case "diagnostic message", and the parameter the PduInfoPtr.SduLength is set to 0, DoIP shall set the parameter availableDataPtr to the size of generic and diagnostic headers plus the length of currently available payload.

Requirements:

### SWS\_DoIP\_00231

- ▶ AUTOSAR GID Synchronization process not implemented

Description:

AUTOSAR GID Synchronization process is not implemented due to reasons described in [https://bugzilla.autosar.org/show\\_bug.cgi?id=80770](https://bugzilla.autosar.org/show_bug.cgi?id=80770). Instead of it EB solution for dynamic GID synchronization is implemented as described in Enhancements document.

Requirements:

SWS\_DoIP\_00050, SWS\_DoIP\_00056, SWS\_DoIP\_00057, SWS\_DoIP\_00076, SWS\_DoIP\_00077, SWS\_DoIP\_00078, SWS\_DoIP\_00079, SWS\_DoIP\_00080, SWS\_DoIP\_00081, SWS\_DoIP\_00085, SWS\_DoIP\_00088, SWS\_DoIP\_00089, SWS\_DoIP\_00263, SWS\_DoIP\_00264

- ▶ User\_DoIPGetGID modified

Description:

User\_DoIPGetGID> is modified to the original SWS behavior described in [SWS\_DoIP\_00051] in the following way: SWS: "If the return value is not E\_OK DoIP shall use the default GID." EB: "If the return value is not E\_OK, DoIP shall keep calling this function within DoIP\_MainFunction() until it returns E\_OK."

Requirements:

### SWS\_DoIP\_00051

- ▶ DoIPWriteDhcpHostNameOption modified

Description:

Original SWS behavior described in [SWS\_DoIP\_00155] is modified in the following way: SWS: "DoIP module shall call the SoAd\_WriteDhcpHostNameOption with a pointer to the string "DoIP-" in order to set the hostname." EB: "DoIP module shall call the SoAd\_WriteDhcpHostNameOption with a pointer to the string "DoIP-" and the string DoIPDhcpHostNamePrefix, if not empty, in order to set the hostname."



Requirements:

SWS\_DoIP\_00155

- ▶ Upper multiplicity set to 65535 for DoIPConnections/DoIPTargetAddress and DoIPRoutingActivation/DoIP-TargetAddressRef.

Description:

Upper multiplicity for DoIPConnections/DoIPTargetAddress and DoIPRoutingActivation/DoIPTargetAddressRef has been increased to 65535 due to Rfc 79727.

Requirements:

ECUC\_DoIP\_00032\_Conf, ECUC\_DoIP\_00034\_Conf

- ▶ DoIP needs to calculate the value of parameter Result of PduR\_DoIPRxIndication(), when called within DoIP\_SoAdTpRxIndication().

Description:

When diagnostic message is fragmented over two Ethernet frames, and Tcp connection got reset before the second frame is received, SoAd will call DoIP\_SoAdTpRxIndication() with NTFRSLT\_OK, because SoAd is not aware that the second part of diagnostic message is not yet received. That is why DoIP needs to calculate value of parameter Result based on internal state when propagating the call to PduR\_-DoIPRxIndication().

Requirements:

SWS\_DoIP\_00200

- ▶ DoIP shall ignore Alive check response message when received before route is activated.

Description:

If Alive check response message is received before the connection is in "registered" state, i.e. before the valid routing activation message is received on that connection, DoIP shall ignore it.

Requirements:

SWS\_DoIP\_00141

- ▶ DoIP\_IfTransmit is not implemented

Requirements:

SWS\_DoIP\_00277

- ▶ DoIP shall not send Diagnostic message negative acknowledge code 0x06 when Diagnostic message is received before route is activated, if configuration parameter DoIPResponseBeforeRoutingActivation is



set to DOIP\_DISABLE\_DIAG\_NACK\_0x06. If config parameter DoIPResponseBeforeRoutingActivation is set to DOIP\_ENABLE\_ALL\_DIAG\_NACK then Diagnostic message negative acknowledge code 0x06 will always be sent.

Requirements:

SWS\_DoIP\_00127

- ▶ TCP closure with FIN used instead of RST when followed by NACK transmission.

Description:

When a configuration parameter DolPEnableTcpClosureWithFIN is ENABLED as a consequence DoIP shall call SoAd\_CloseSoCon(Abort=FALSE) whenever a NACK/negative response needs to be sent together with a TCP closure, meaning TCP closure with FIN instead of RST. In all other cases behaviour shall remain the same Abort=TRUE (TCP closure with RST).

Requirements:

SWS\_DoIP\_00058, SWS\_DoIP\_00115, SWS\_DoIP\_00144, SWS\_DoIP\_00146, SWS\_DoIP\_00140, SWS\_DoIP\_00141,

### 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.

- ▶ Configuration parameter `DoIPMaxRequestBytes` limited

Description:

The configuration parameter `DoIPMaxRequestBytes` is limited based on a size of PduLengthType as follows:

1) When PduLengthType equals uint16 then the maximum value of `DoIPMaxRequestBytes` is 65535. This is also the maximum size of diagnostic message user data without headers.

2) When PduLengthType equals uint32 then the maximum value of `DoIPMaxRequestBytes` is 4294967295. This is also the maximum size of diagnostic message user data with diagnostic message header (4 bytes). The maximum size of diagnostic message user data without headers is 4294967295 - 4.

- ▶ Configuration parameter `DoIPMaxChannels` limited

Description:

Configuration parameter `DoIPMaxChannels`, that defines maximum configurable number of DoIP channels, is limited to 65535.



Rationale:

The largest value is used at out-of-range delimiter.

- ▶ Lower multiplicity of DoIPRoutingActivation/DoIPTargetAddressRef set to 0

Description:

Lower multiplicity of the following node changed from 1 to 0: DoIP/DoIPConfigSet/DoIPRoutingActivation/DoIPTargetAddressRef

- ▶ Lower multiplicity of DoIPTester/DoIPRoutingActivationRef set to 0

Description:

Lower multiplicity of the following node has been changed from 1 to 0: DoIP/DoIPConfigSet/DoIPTester/DoIPRoutingActivationRef

Rationale:

The change is a consequence of implementation of automatic routing activation feature. For channels with DoIPRoutingActivationType set to DOIP\_ROUTING\_ACTIVATION\_AUTOMATIC, it is not necessary to receive Routing activation message to activate the route, hence lower multiplicity is set to 0.

- ▶ Lower multiplicity of /DoIP/DoIPGeneral/DoIPPowerModeCallback set to 0

Description:

Lower multiplicity of the following node has been changed from 1 to 0: /DoIP/DoIPGeneral/DoIPPowerModeCallback

Rationale:

Since the parameter DoIPPowerModeCallback is optional, its lower multiplicity must be 0.

- ▶ DoIP supports only 2 Protocol versions of DoIP packets

Description:

Since only Protocol Versions 0x02 and 0xFF are supported, check is added to only allow these two Protocol Versions.

- ▶ Multiplicity of DoIPTcpConnection/DoIPTcpConnectionSecurityRequired set to 1

Description:

Multiplicity of the following node has been changed from 0..1 to 1. DoIPTcpConnection/DoIPTcpConnectionSecurityRequired

Rationale:



Making parameter `DoIPTcpConnectionSecurityRequired` optional has no impact.

- ▶ Multiplicity of `DoIPRoutingActivation/DoIPRoutingActivationSecurityRequired` set to 1

Description:

Multiplicity of the following node has been changed from 0..1 to 1. `DoIPRoutingActivation/DoIPRoutingActivationSecurityRequired`

Rationale:

Making parameter `DoIPRoutingActivationSecurityRequired` optional has no impact.

### 3.3.1.6. Open-source software

DoIP does not use open-source software.

## 3.3.2. Ethif module release notes

- ▶ AUTOSAR R4.3 Rev 0
- ▶ AUTOSAR SWS document version: 4.3.0
- ▶ Module version: 1.9.19.B466224
- ▶ Supplier: Elektrobit Automotive GmbH

### 3.3.2.1. Change log

This chapter lists the changes between different versions.

#### Module version 1.9.19

2021-10-08

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

#### Module version 1.9.18

2021-06-25

- ▶ Removed Cable diagnostic feature.
- ▶ Implemented support of post-build variant handling.



- ▶ ASCETHIF-642 Fixed known issue: BswM\_EthIf\_PortGroupLinkStateChg() / EthSM\_TrsvLinkStateChg() get called with LINK\_DOWN before EthIfSwitchOffPortTimedelay elapsed.

#### **Module version 1.9.17**

2021-03-05

- ▶ ASCETHIF-623 Fixed known issue: EthIf accesses wrong Ethernet transceiver.
- ▶ Implemented support for drivers that contain vendorId and vendorApilnfix.
- ▶ Implemented support of multiple Eth controller and Eth transceiver.

#### **Module version 1.9.16**

2020-10-23

- ▶ ASCETHIF-596 Fixed known issue: EthIf accesses wrong Eth Controller if referenced EthCtrlIdx are not zero based and consecutive.

#### **Module version 1.9.15**

2020-06-19

- ▶ ASCETHIF-589 Fixed known issue: Successfully received non VLAN tagged frames may be counted as frame drops while a received VLAN ID which is not configured is dropped but not counted.

#### **Module version 1.9.14**

2020-02-21

- ▶ Implemented EthIf\_GetVlanId API.

#### **Module version 1.9.13**

2019-10-11

- ▶ ASCETHIF-553 Fixed known issue: Compilation error occurs if EthTrcv with AUTOSAR version below 4.-3.0 is used.

#### **Module version 1.9.12**

2019-09-06

- ▶ Added locking per switch port to prevent a preemptive access of same Trcv registers in the chip



### Module version 1.9.11

2019-06-14

- ▶ ASCETHIF-573 Fixed known issue: EthIf does not correctly notify the BswM about port group link state changes.
- ▶ ASCETHIF-516 Fixed known issue: Compilation error occurs if virtual controller support is enabled and development error detection is disabled.
- ▶ Added EthIf support for external ASR 4.3.0 Eth drivers. Integration requirements EB\_INTREQ\_EthIf\_0006 and EB\_INTREQ\_EthIf\_0007 should be taken into account.
- ▶ ASCETHIF-536 Fixed known issue: EthIf doesn't provide EthSM\_CtrlModeIndication on partial deactivation of virtual EthIfControllers connected to EthIfSwitchPortGroups.

### Module version 1.9.10

2019-02-15

- ▶ Added EthIf support for Device Authentication

### Module version 1.9.9

2019-01-24

- ▶ ASCETHIF-502 Fixed known issue: Port reference counter underflow

### Module version 1.9.8

2018-10-26

- ▶ ASCETHIF-460 Fixed known issue: RTE can not schedule EthIf\_MainFunctionState().
- ▶ Added Measurement data support
- ▶ Implemented Post-build binary support

### Module version 1.9.7

2018-06-22

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

### Module version 1.9.6

2018-05-25

- ▶ ASCETHIF-428 Fixed known issue: BeginIdx in EthIf\_SwtPortGroupType could get overflow



- ▶ Added support for interrupt mode

#### **Module version 1.9.5**

2018-04-20

- ▶ Updated support of DEM reporting for hardware link failures

#### **Module version 1.9.4**

2018-03-16

- ▶ Added support for DEM reporting for hardware link failures

#### **Module version 1.9.3**

2018-02-16

- ▶ Updated Tx frame preprocessing to allow switch delay compensation.
- ▶ ASCETHIF-380 Fixed known issue: Exclusive areas are not generated in Rte if EthIfMiiApiEnable is set to FALSE
- ▶ Updated design according to new template

#### **Module version 1.9.2**

2017-12-15

- ▶ Added frame preprocessing to allow switch delay compensation.

#### **Module version 1.9.1**

2017-11-17

- ▶ Added support for async handling of transceiver/controller mode API (Configuration)
- ▶ Added Link quality monitoring support

#### **Module version 1.9.0**

2017-09-22

- ▶ Replaced EthIf\_GetPhySignalQuality() API
- ▶ Updated to MISRA 2012
- ▶ ASCETHIF-344 Fixed known issue: Configuration of exclusive area for EthIf is not possible in RTE



### **Module version 1.8.3**

2017-08-25

- ▶ Added Wake-up support

### **Module version 1.8.2**

2017-07-28

- ▶ Improved switching of switch port groups support

### **Module version 1.8.1**

2017-06-30

- ▶ Added support of switching of switch port group feature according to AUTOSAR SWS 4.3.0. specification

### **Module version 1.8.0**

2017-06-02

- ▶ Added support of asynchronous handling of transceiver/controller mode API according to AUTOSAR SWS 4.3.0. specification

### **Module version 1.7.2**

2017-05-05

- ▶ Updated MII protection to postpone link state check if MII is in use

### **Module version 1.7.1**

2017-03-31

- ▶ Updated EthIf to AUTOSAR 4.3.0

### **Module version 1.7.0**

2017-03-03

- ▶ Implemented alignment of configuration scheme according to AUTOSAR SWS 4.3.0
- ▶ Added abstracted access to registers for diagnostic and testing purposes according to AUTOSAR SWS 4.3.0. specification



### Module version 1.6.7

2017-01-05

- ▶ Updated QoS support and config and transceiver config structure according to AUTOSAR SWS 4.3.0. specification

### Module version 1.6.6

2016-11-04

- ▶ Added quality of service (QoS) support.
- ▶ Added Ethernet Switch Support according to Autosar V4.2.1.
- ▶ Updated EthIf to use Eth\_BuflIdxType
- ▶ ASCETHIF-176 Fixed known issue: Invalid Bswmd file is generated if EthIfMiiApiEnable is enabled

### Module version 1.6.5

2016-04-01

- ▶ Added function `EthIf_ReadMii()` and `EthIf_WriteMii()` to support access to MII interface.
- ▶ ASCETHIF-150 Fixed known issue: The call of `EthIf_SetTransceiverMode()` with more configured transceiver than controller may lead to invalid memory access

### Module version 1.6.4

2016-02-05

- ▶ Changed configuration parameter `EthIfEthTrcvRef` to be enabled by default

### Module version 1.6.3

2015-11-06

- ▶ Updated recommended configuration for EthTSyn

### Module version 1.6.2

2015-06-19

- ▶ Improved usability of generic upper layer support



- ▶ Added task auto assign of EthIf\_MainFunctionRx() and EthIf\_MainFunctionTx() for RTE

#### **Module version 1.6.1**

2015-02-23

- ▶ Added support for Ethernet time synchronization
- ▶ ASCETHIF-126 Fixed known issue: Not configured frames without VLAN tag forwarded to upper layer
- ▶ ASCETHIF-129 Fixed known issue: Incorrect type in call to EthTrcv\_GetTransceiverMode()

#### **Module version 1.6.0**

2014-10-02

- ▶ Added generic upper layer support
- ▶ Added Tx confirmation polling
- ▶ ASCETHIF-92 Fixed known issue: EthIf\_SetTransceiverMode() may cause an memory write access violation
- ▶ ASCETHIF-91 Fixed known issue: EthIf does not compile if used with Eth according to AUTOSAR release 4.1.1 and Det is enabled
- ▶ Added defensive programming
- ▶ Improved transceiver link state change detection

#### **Module version 1.5.0**

2014-03-21

- ▶ Added parameter TxConfirmation to EthIf\_Transmit()
- ▶ Updated EthIf to AUTOSAR 4.1.2

#### **Module version 1.4.1**

2013-06-14

- ▶ Improved internal module. This module version update does not affect module functionality

#### **Module version 1.4.0**

2013-04-08

- ▶ Replaced (renamed) container EthIfCtrl to EthIfController in module config schema



### Module version 1.3.3

2013-02-08

- ▶ Improved internal module. This module version update does not affect module functionality

### Module version 1.3.2

2012-10-12

- ▶ Added support for MAC Groupcast reception
- ▶ Added support for multiple virtual EthIf controllers per EthCtrl/EthTrcv

### Module version 1.3.1

2012-07-16

- ▶ ASCETHIF-28 Fixed known issue: Transition to COMM\_FULL\_COMMUNICATION fails

### Module version 1.3.0

2012-07-12

- ▶ Added support for EthTrcv

### Module version 1.2.4

2012-06-21

- ▶ ASCETHIF-14 Fixed known issue: EthIfTransceiverInit() always returns E\_NOT\_OK
- ▶ ASCETHIF-18 Fixed known issue: EthIf might call Eth\_Receive() with the wrong CtrlIdx
- ▶ ASCETHIF-19 Fixed known issue: EthIf polls Eth for data reception even if the Ethernet controller is not active

### Module version 1.1.0

2012-04-13

- ▶ Initial AUTOSAR 4.0 version

#### 3.3.2.2. New features

- ▶ Added support for post-build selectable variant handling.



### 3.3.2.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- ▶ **EthIf provides** `EthIf_Cbk_RxIndication()` **and** `EthIf_Cbk_TxConfirmation()`

Description:

In addition to the AUTOSAR 4.3.0 specified functions `EthIf` also provides the functions `EthIf_Cbk_RxIndication()` **and** `EthIf_Cbk_TxConfirmation()` as specified in AUTOSAR 4.0.3.

- ▶ **EthIf provides** `EthIf_ReadMii()` **and** `EthIf_WriteMii()`

Description:

In addition to the AUTOSAR 4.3.0 specified functions `EthIf` also provides the functions `EthIf_ReadMii()` **and** `EthIf_WriteMii()` to provide read and write access to registers of an Ethernet Transceiver via the MII interface of the Ethernet Controller.

- ▶ **EthIf provides** `EthIf_SwtGetCounterValues()`

Description:

In addition to the AUTOSAR 4.3.0 specified functions `EthIf` also provides the function `EthIf_SwtGetCounterValues()` to get counter values from values of an Ethernet Switch.

- ▶ **EthIf provides** `EthIf_SetCorrectionTime()`

Description:

In addition to the AUTOSAR 4.3.0 specified functions `EthIf` also provides the function `EthIf_SetCorrectionTime()` to allow the Time Slave to adjust the local ETH Reference clock in HW.

- ▶ **EthIf provides** `EthIf_SetGlobalTime()`

Description:

In addition to the AUTOSAR 4.3.0 specified functions `EthIf` also provides the function `EthIf_SetGlobalTime()` to allow the Time Master to adjust the global ETH Reference clock in HW.

- ▶ **EthIf** can optionally initialize physical controllers and transceivers.

Description:

`EthIf` provides configuration parameter in General container `EthIfInitControllersTransceivers` to enable / disable `EthIf` to initialize configured physical controllers and transceivers.

- ▶ **EthIf** can switch between sync/asyn communication of physical controllers and transceivers.

Description:



**EthIf** additionally provides two configuration parameters in General container to switch between sync/asyn communication - `EthIfAsyncEthTrcvModeSupport` for `EthTrcv` and `EthIfAsyncEthCtrlModeSupport` for physical controllers.

- ▶ Added frame preprocessing to allow switch delay compensation.

Description:

The configuration parameter `EthIfSwtPreProcessRxFrame` allows the `EthSwt` to preprocess received frames to allow EB specific switch delay compensation.

- ▶ `EthIf` can optionally enable periodic execution of `EthIf_MainFunctionState()`.

Description:

`EthIf` provides configuration parameter in General container `EthIfMainFunctionStatePeriod` which enables periodic execution of `EthIf_MainFunctionState()`. If parameter is not set this function will be executed as part of `EthIf_MainFunctionRx`.

- ▶ `EthIf` provides `EthIf_EnableRelatedEthIfCtrls()` and `EthIf_DisableRelatedEthIfCtrls()`

Description:

In addition to the AUTOSAR 4.3.0 specified functions `EthIf` also provides the functions `EthIf_EnableRelatedEthIfCtrls()` and `EthIf_DisableRelatedEthIfCtrls()` to disable and enable `EthIf` Controllers in order to support Device Authentication module.

- ▶ `EthIf` provides `EthIf_Retransmit()`

Description:

In addition to the AUTOSAR 4.3.0 specified functions `EthIf` also provides function `EthIf_Retransmit()` to retransmit the current buffer.

- ▶ `EthIf` provides locking mechanism

Description:

In addition to the AUTOSAR 4.3.0 specified functionality `EthIf` also provides locking per transceiver and per switch port for all Switch APIs in order to prevent a preemptive access of the same Trcv registers in the chip.

- ▶ `EthIf` optionally provides support for drivers that contain vendorId and vendorApiInfix as well as support of multiple Eth controllers and Eth transceivers.

Description:

`EthIf` provides lists of Bswmd references: `EthIfEthControllerBswmdImplementationRefs`, `EthIfEthTrcvBswmdImplementationRefs` and `EthIfEthSwtBswmdImplementationRefs`. If one or multiple Bswmd refer-



ences are configured in these lists they will be used to obtain vendorId and vendorApiInfix needed for supporting multiple Eth controllers and Eth transceivers or for function name mangling for a single Eth driver.

### 3.3.2.4. Deviations

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

- ▶ Operation in interrupt mode not supported

Description:

`EthIf` does not support operation in interrupt context. Main function `EthIf_MainFunctionRx` always polls for new frames. This behavior cannot be disabled. The main function `EthIf_MainFunctionTx` always polls for new Tx confirmations. This behavior cannot be disabled. The configuration parameters `EthIfEnableRxInterrupt` and `EthIfEnableTxInterrupt` are unused.

Rationale:

`EthIf` is designed to be used in combination with other EB IP-stack modules. In this composition polling is the default behavior.

Requirements:

ECUC\_EthIf\_00005, ECUC\_EthIf\_00006, SWS\_EthIf\_00099, SWS\_EthIf\_00100

- ▶ Different polling behavior for transceiver link state

Description:

`EthIf` polls for the transceiver link state with each period of `EthIf_MainFunctionState` if config param `EthIfMainFunctionStatePeriod` is set, otherwise with each period of `EthIf_MainFunctionRx`. Configuration parameter `EthIfTrcvLinkStateChgMainReload` is unused.

Requirements:

ECUC\_EthIf\_00009, SWS\_EthIf\_00101

- ▶ No macro variant of `EthIf_GetVersionInfo` available

Description:

The API service `EthIf_GetVersionInfo` is not available as a macro implementation. The configuration parameter `EthIfVersionInfoApiMacro` is unused.

Requirements:

ECUC\_EthIf\_00008

- ▶ EthIf supports config variant post build only



#### Description:

EthIf supports config variant post build only. However, it does not support initialization by the PbcfgM module.

#### Requirements:

SWS\_EthIf\_00005

- ▶ No consistency check between code files and header files

#### Description:

The inter-module version checks as specified by the EthIf SWS are not implemented.

#### Rationale:

- ▶ The required compile-time version checks would result in an inflexible, hardly integratable basic software stack.
- ▶ EB tresos AutoCore is an already integrated product.
- ▶ The project handling of EB tresos Studio provides means to enforce that only modules with the same EB tresos AutoCore release version can be added to the project.

#### Requirements:

SWS\_EthIf\_00007

- ▶ Initialization check in `EthIf_MainFunctionState()`, `EthIf_MainFunctionTx()` and `EthIf_MainFunctionRx()`

#### Description:

If `EthIf_MainFunctionState()`, `EthIf_MainFunctionTx()` or `EthIf_MainFunctionRx()` is called while the module is not yet initialized the function returns immediately without performing any functionality and without raising any Det error. This initialization check is always performed independent of the development error detection setting.

The SchM module may schedule the modules main functions before the module is initialized. This would result in lots of Det errors during start up. Therefore the module's main function does not throw a Det error if the module is not yet initialized and simply returns in this case.

#### Requirements:

SWS\_EthIf\_00098, SWS\_EthIf\_00124, SWS\_EthIf\_00278

- ▶ `EthIfMaxTxBufsTotal` not required

#### Description:



There is no need for EthIf to have any knowledge about Tx buffer index. The parameter `EthIfMaxTxBufsTotal` is unused. An invalid check of the Tx buffer index is not performed.

Rationale:

See [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=28868](http://www.autosar.org/bugzilla/show_bug.cgi?id=28868).

Requirements:

ECUC\_EthIf\_00002, SWS\_EthIf\_00079, SWS\_EthIf\_00094

- ▶ Optional APIs `GetCounterState`, `GetBaudRate` and `StartAutoNegotiation` not supported

Description:

The optional APIs for `GetCounterState`, `GetBaudRate`, and `StartAutoNegotiation` are not supported. The corresponding configuration parameters are disabled.

Rationale:

See [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=28868](http://www.autosar.org/bugzilla/show_bug.cgi?id=28868).

Requirements:

ECUC\_EthIf\_00033, ECUC\_EthIf\_00034, ECUC\_EthIf\_00035,

- ▶ Generic upper layer handling can only be configured at link time

Description:

The generic upper layer handling can only be configured at link time.

This also leads to a violation of rule `TpsEcuc_06051_ASR41`: The `implementationConfigClass` of an `EcucParameterDef` or `EcucAbstractReferenceDef` in VSMD shall be the same or higher (where `PreCompile` configuration class is considered to be the lowest and `PostBuild` the highest) as in StMD with respect to the selected subset defined by the actually implemented `supportedConfigVariant`.

- ▶ StMD-Node:/AUTOSAR/EcucDefs/EthIf/EthIfConfigSet/EthIfFrameOwnerConfig/EthIfFrameType
- ▶ StMD-Node:/AUTOSAR/EcucDefs/EthIf/EthIfConfigSet/EthIfFrameOwnerConfig/EthIfOwner
- ▶ StMD-Node:/AUTOSAR/EcucDefs/EthIf/EthIfConfigSet/EthIfRxIndicationConfig/EthIfRxIndicationFunction
- ▶ StMD-Node:/AUTOSAR/EcucDefs/EthIf/EthIfConfigSet/EthIfTrcvLinkStateChgConfig/EthIfTrcvLinkStateChgFunction
- ▶ StMD-Node:/AUTOSAR/EcucDefs/EthIf/EthIfConfigSet/EthIfTxConfirmationConfig/EthIfTxConfirmationFunction
- ▶ StMD-Node:/AUTOSAR/EcucDefs/EthIf/EthIfGeneral/EthIfPublicCddHeaderFile



#### Rationale:

Upper layer function names must be available at link time.

#### Requirements:

ECUC\_EthIf\_00012, ECUC\_EthIf\_00013, ECUC\_EthIf\_00015, ECUC\_EthIf\_00017, ECUC\_EthIf\_00019

- ▶ `EthIf` does not support Wireless Ethernet feature

#### Description:

The following functions have not been implemented: `EthIf_GetBufWRxParams`, `EthIf_GetBufWTxParams`, `EthIf_SetBufWTxParams`, `EthIf_SetRadioParams`, `EthIf_SetChanRxParams`, `EthIf_SetChanTxParams`, `EthIf_GetChanRxParams`.

#### Requirements:

SWS\_EthIf\_00340, SWS\_EthIf\_91002, SWS\_EthIf\_00341, SWS\_EthIf\_00342, SWS\_EthIf\_00343, SWS\_EthIf\_00344, SWS\_EthIf\_00345, SWS\_EthIf\_00346, SWS\_EthIf\_91009\_GetBufWTxParams, SWS\_EthIf\_00347, SWS\_EthIf\_00348, SWS\_EthIf\_00349, SWS\_EthIf\_00350, SWS\_EthIf\_00351, SWS\_EthIf\_00352, SWS\_EthIf\_91017, SWS\_EthIf\_00353, SWS\_EthIf\_00354, SWS\_EthIf\_00355, SWS\_EthIf\_00356, SWS\_EthIf\_00357, SWS\_EthIf\_00358, SWS\_EthIf\_00359, SWS\_EthIf\_91026, SWS\_EthIf\_00360, SWS\_EthIf\_00361, SWS\_EthIf\_00362, SWS\_EthIf\_00363, SWS\_EthIf\_00364, SWS\_EthIf\_00365, SWS\_EthIf\_91034, SWS\_EthIf\_00366, SWS\_EthIf\_00367, SWS\_EthIf\_00368, SWS\_EthIf\_00369, SWS\_EthIf\_00370, SWS\_EthIf\_00371, SWS\_EthIf\_00372, SWS\_EthIf\_91042, SWS\_EthIf\_00373, SWS\_EthIf\_00374, SWS\_EthIf\_00375, SWS\_EthIf\_00376, SWS\_EthIf\_00377, SWS\_EthIf\_00378, SWS\_EthIf\_00379, SWS\_EthIf\_91050, SWS\_EthIf\_00380, SWS\_EthIf\_00381, SWS\_EthIf\_00382, SWS\_EthIf\_00383, SWS\_EthIf\_00384, SWS\_EthIf\_00385, SWS\_EthIf\_00386

- ▶ `EthIf_CheckWakeups` calls `EthTrcv_CheckWakeups` for all configured `EthIfTransceivers`

#### Description:

Although `EthIf_CheckWakeups()` is called with a parameter `WakeupSource` specifying which wake-up sources should be checked and reported, the function `EthTrcv_CheckWakeups()` does not have the parameter `WakeupSource`. This means that it will report all active wake-up sources, not only those requested by EcuM. This requires EcuM to filter-out unwanted wake-up sources. Since this mechanism is in place `EthIf` can poll all configured transceivers, with no need to parse transceiver configuration related to wake-up source configuration.

#### Requirements:

`SWS_EthIf_00245`

- ▶ `EthIf_GetPortMacAddr` is not implemented, i.e. the call is not forwarded to Ethernet Switch Driver.

#### Requirements:



SWS\_EthIf\_00191

- ▶ **EthIf\_GetCtrlIdxList is not implemented**

Requirements:

SWS\_EthIf\_91053, SWS\_EthIf\_00298, SWS\_EthIf\_00300

- ▶ The feature Switch Management Info is not implemented

Description:

The following functions have not been implemented: EthIf\_SwitchMgmtInfoIndication, EthIf\_SetSwitchMgmtInfo.

Requirements:

SWS\_EthIf\_91003, SWS\_EthIf\_00279, SWS\_EthIf\_00280, SWS\_EthIf\_00281, SWS\_EthIf\_00282  
SWS\_EthIf\_00283, SWS\_EthIf\_00284, SWS\_EthIf\_91006, SWS\_EthIf\_00291, SWS\_EthIf\_91006\_User,  
SWS\_EthIf\_00295

- ▶ **EthIf\_SwitchEnableTimeStamping is not implemented**

Requirements:

SWS\_EthIf\_91007, SWS\_EthIf\_00387, SWS\_EthIf\_00285, SWS\_EthIf\_00286, SWS\_EthIf\_00287,  
SWS\_EthIf\_00288, SWS\_EthIf\_00289, SWS\_EthIf\_00290

- ▶ **EthIf\_VerifyConfig is not implemented**

Requirements:

SWS\_EthIf\_91012, SWS\_EthIf\_00305

- ▶ **EthIf\_SetForwardingMode is not implemented**

Requirements:

SWS\_EthIf\_91013, SWS\_EthIf\_00307

- ▶ **EthIf\_TxConfirmation will not pass the parameter Result received within EthIf\_TxConfirmation to the configured upper layer via UL\_TxConfirmation.**

Requirements:

SWS\_EthIf\_00255, SWS\_EthIf\_00106

- ▶ The feature of retrieving ingress/egress time stamp value out of the switch is not supported

Description:



The following functions have not been implemented: `EthIf_SwitchEgressTimeStampIndication`, `EthIf_SwitchIngressTimeStampIndication`.

Requirements:

`SWS_EthIf_91009_SwitchEgressTimeStampInd`, `SWS_EthIf_00293`, `SWS_EthIf_91008`, `SWS_EthIf_00294`, `SWS_EthIf_91009_User_SwitchEgressTimeStampInd`, `SWS_EthIf_00296`, `SWS_EthIf_91008_User`, `SWS_EthIf_00297`

- ▶ `EthIf_GetPhySignalQuality()` is replaced by `EthIf_GetTrcvSignalQuality()`.

Description:

The function `EthIf_GetPhySignalQuality()` deprecated and replaced by `EthIf_GetTrcvSignalQuality()` to retrieve the actual signal quality from the transceiver.

Rationale:

See [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=77676](http://www.autosar.org/bugzilla/show_bug.cgi?id=77676).

Requirements:

`SWS_EthIf_91019`, `SWS_EthIf_00320`, `SWS_EthIf_00321`, `SWS_EthIf_00322`, `SWS_EthIf_00323`

- ▶ BswM controlled port group link state is `ETHTRCV_LINK_STATE_DOWN` when port group mode is `ETH_MODE_DOWN`.

Description:

In case that `EthIfSwitchPortGroup` mode is set to `ETH_MODE_DOWN` then port group link state shall be `ETHTRCV_LINK_STATE_DOWN`. Consequently when accumulated link state of `EthIfSwitchPortGroup` is `ETHTRCV_LINK_STATE_ACTIVE`, and its mode changes to `ETH_MODE_DOWN`, `EthIf` shall report to BswM the change of port group link state to `ETHTRCV_LINK_STATE_DOWN`.

Rationale:

Rationale: This behavior is in line with port groups that are under control of `EthIfController` and `EthSM`. `EthSM` depends on this behavior for correct operation.

Requirements:

`SWS_EthIf_00261`

- ▶ Value configuration class changed and variant handling support added for `EthIfSwitchIdx` and `EthIfSwitchPortGroupIdx` configuration parameters.

Description:



The following configuration parameters are modified to be post-build changeable and support variant handling: - EthIfSwitch/EthIfSwitchIdx - EthIfSwitchPortGroup/EthIfSwitchPortGroupIdx.

Rationale:

Rationale: Although according to SWS, EthIfSwitchIdx and EthIfSwitchPortGroupIdx have pre-compile configuration class and variant handling is not supported, according to TPS\_ECUC\_08002, it shall be possible to add new container elements of specific types.

Requirements:

ECUC\_EthIf\_00037\_Conf, ECUC\_EthIf\_00058\_Conf

### 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.

- ▶ **EthCtrlIdx** must be consecutive and zero-based.

Description:

The Ethernet driver configuration parameter `EthCtrlIdx` must be configured consecutive and zero-based.

Rationale:

This restriction allows `EthIf` to perform a fast translation to other indexes using look-up tables.

- ▶ **EthTrcvIdx** must be consecutive and zero-based.

Description:

The Ethernet transceiver configuration parameter `EthTrcvIdx` must be configured consecutive and zero-based.

Rationale:

This restriction allows `EthIf` to perform a fast translation to other indexes using look-up tables.

- ▶ **EthTrcvIdx** shall not be configured to 255.

Description:

The Ethernet transceiver configuration parameter `EthTrcvIdx` shall not be configured to 255.

Rationale:



Since the use of Ethernet transceiver is optional, 255 is used to replace `EthTrcvIdx` in case the transceiver is disabled for a certain Ethernet controller. Therefore, 255 can not be used as valid `EthTrcvIdx`.

- ▶ `EthIfVlanId` shall not be configured to 0 or 4095.

Description:

Virtual-LAN is identified by this attribute according to IEEE 802.1Q which states that hexadecimal values of 0x000 and 0xFFFF are reserved.

Rationale:

The reserved value 0x000 indicates that the frame does not carry a VLAN ID and VID value 0xFFFF is reserved for implementation use.

- ▶ Counting of Measurement data for measurement index `ETHIF_MEAS_DROP_CRTLIDX`

Description:

Measurement data counter shall also be incremented in following cases: - invalid CrtlIdx - according to `SWS_EthIf_00309` - VLAN is not enabled, but received frame has VLAN tag - according to `SWS_EthIf_00309` - no EthIfController exists with matching VLAN ID - according to `SWS_EthIf_00309` - EthIfController which would match to VLAN ID is not in `MODE_ACTIVE` - according to `SWS_EthIf_00317` - VLAN is not enabled and EthIfController is not in `MODE_ACTIVE` - according to `SWS_EthIf_00309` Measurement data counter will not be incremented in following cases: - received FrameType is not configured - received frame is VLAN tagged but has not enough bytes for the EtherType or no payload at all

Rationale:

This Limitation defines all counted frame drops which are according to ASR requirements and highlights not counted frame drops that are not explicitly described in ASR.

- ▶ `EthIf` APIs should be called with parameters within the context of the Ethernet Interface module. The exception is `SwitchPortIdx` parameter, as there is no configuration of `SwitchPort` in `EthIf`, so `SwitchPortIdx` of Ethernet Switch Driver is used.

Description:

Following APIs shall use indexes within the context of `EthIf` instead of within the context of the Ethernet Switch Driver: - `EthIf_GetArTable` - `EthIf_GetBufferLevel` - `EthIf_SwtGetCounterValues` - `EthIf_VerifyConfig` - `EthIf_SetForwardingMode`

Rationale:

These APIs are defined in ASR Specification of Ethernet Interface to use indexes in non `EthIf` context - which is not correct.

### 3.3.2.6. Open-source software

Ethif does not use open-source software.

## 3.3.3. EthSM module release notes

- ▶ AUTOSAR R4.3 Rev 0
- ▶ AUTOSAR SWS document version: 4.3.0
- ▶ Module version: 1.6.14.B466224
- ▶ Supplier: Elektrobit Automotive GmbH

### 3.3.3.1. Change log

This chapter lists the changes between different versions.

#### Module version 1.6.14

2021-06-25

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

#### Module version 1.6.13

2021-03-05

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

#### Module version 1.6.12

2020-10-23

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

#### Module version 1.6.11

2020-06-19

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

#### Module version 1.6.10

2020-02-21



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

#### **Module version 1.6.9**

2019-06-14

- ▶ Implemented support for DevAuth

#### **Module version 1.6.8**

2019-02-15

- ▶ ASCETHSM-223 Fixed known issue: CONST variables are not defined in an appropriate MemMap section.
- ▶ Internal module improvement. This module version update does not affect module functionality.

#### **Module version 1.6.7**

2018-10-26

- ▶ Implemented BSW Distribution support
- ▶ Implemented Post-build binary support

#### **Module version 1.6.6**

2018-06-22

- ▶ Changed reporting of Dem events to PREPASSED and PREFAILED to allow debouncing

#### **Module version 1.6.5**

2018-02-16

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

#### **Module version 1.6.4**

2017-11-17

- ▶ Removed VSMD violations

#### **Module version 1.6.3**

2017-09-22

- ▶ Updated to MISRA 2012



### **Module version 1.6.2**

2017-08-25

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

### **Module version 1.6.1**

2017-06-30

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

### **Module version 1.6.0**

2017-06-02

- ▶ Implemented Support of async handling for transceiver/controller mode API

### **Module version 1.5.11**

2017-05-05

- ▶ Updated APIs based on AUTOSAR 4.3.0

### **Module version 1.5.10**

2017-03-31

- ▶ Updated EthSM to AUTOSAR 4.3.0.

### **Module version 1.5.9**

2017-03-03

- ▶ Updated post build config support

### **Module version 1.5.8**

2017-02-03

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

### **Module version 1.5.7**

2016-10-07

- ▶ Improved state machine transition to ETHSM\_STATE\_OFFLINE for VLANs



### Module version 1.5.6

2016-04-01

- ▶ Adapted the network mode state machine of the EthSM

### Module version 1.5.5

2016-02-05

- ▶ Added service needs assistance support for DEM event `ETHSM_E_LINK_DOWN`

### Module version 1.5.4

2015-11-06

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

### Module version 1.5.3

2015-06-19

- ▶ Added production error reporting of `ETHSM_E_LINK_DOWN`
- ▶ Added task auto assign of `EthSM_MainFunction()` for RTE

### Module version 1.5.2

2015-02-20

- ▶ ASCETHSM-78 Fixed known issue: State changes might be lost in case of API function preemptions

### Module version 1.5.1

2014-10-02

- ▶ ASCETHSM-66 Fixed known issue: API `EthSM_TcpIpModeIndication` reports unattended DET error `ETHSM_E_INVALID_TcpIpMode`

### Module version 1.5.0

2014-03-21

- ▶ Updated EthSM to AUTOSAR 4.1.2.
- ▶ Changed configuration parameter name from `EthSMControllerRef` to `EthSMEthIfControllerRef`



#### **Module version 1.4.1**

2013-06-14

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

#### **Module version 1.4.0**

2013-04-08

- ▶ Updated reference paths of `EthSMEthIfControllerRef` for the renamed container `EthIfController` in `EthIf`

#### **Module version 1.3.0**

2013-02-08

- ▶ Changed to asynchronous execution of all transitions in `EthSM_MainFunction`

#### **Module version 1.2.3**

2012-10-16

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

#### **Module version 1.2.2**

2012-09-18

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

#### **Module version 1.2.1**

2012-07-16

- ▶ ASCETHSM-18 Fixed known issue: Transition to `COMM_FULL_COMMUNICATION` fails

#### **Module version 1.2.0**

2012-07-12

- ▶ Add support for `EthTrcv`

#### **Module version 1.1.1**

2012-06-21



- ▶ Added generation of BSWMD
- ▶ ASCETHSM-10 Fixed known issue: The code generation crashes in case of invalid references
- ▶ Added support for `COMM_SILENT_COMMUNICATION`

#### Module version 1.1.0

2012-04-13

- ▶ First release of EB tresos AutoCore EthSM module
- ▶ Initial prototype implementation of EthSM module

#### 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.

- ▶ Optimization: Single network usage

With the parameter `EthSMGeneral/EthSMSingleNetworkOptEnable` it is possible to enable single network optimization.

- ▶ EthSM provides `EthSM_GetCurrentComMode()`

Description:

In addition to the AUTOSAR 4.3.0 specified functions `EthSM` also provides the functions `EthSM_GetCurrentComMode()` to report `ETHSM_E_PARAM_POINTER` to the DET, if the pointer of the parameter list is invalid.

This is only valid if `EthSMDevErrorDetect` is enabled. Extension: The API shall return `E_NOT_OK` in this case.

- ▶ EthSM provides `EthSM_TrcvLinkStateChg()`

Description:

In addition to the AUTOSAR 4.3.0 specified functions `EthSM` also provides the functions `EthSM_TrcvLinkStateChg()` to report `ETHSM_E_INVALID_TRCV_LINK_STATE` to the DET, if it does not accept the transceiver link state of the function call.

This is only valid if `EthSMDevErrorDetect` is enabled.



- ▶ EthSM reports `DEM_EVENT_STATUS_PREPASSED`

Description:

EthSM will also report `DEM_EVENT_STATUS_PREPASSED` during the transition from `ETHSM_STATE_WAIT_TRCVLINK` to `ETHSM_STATE_WAIT_ONLINE` as well as in transition from `ETHSM_STATE_ONHOLD` to `ETHSM_STATE_ONLINE`. During the transition from `ETHSM_STATE_ONLINE` to `ETHSM_STATE_ONHOLD` event `DEM_EVENT_STATUS_PREFAILED` shall be reported.

This is only valid if `EthSMDemCtrlTestResultReportToDem` is set to `DEM` and optional configuration parameter `ETHSM_E_LINK_DOWN` exists.

- ▶ EthSM supports relocatable post-build module variant.

Description:

Relocatable post-build configuration is enabled/disabled with configuration parameter `EthSMRelocatablePbcfgEnable`.

#### 3.3.3.4. Deviations

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

- ▶ `EthSMDummyMode` is not supported

Description:

EthSM does not provide a dummy mode for ComM.

Rationale:

Feature is not implemented.

Requirements:

`SWS_EthSM_00078`, `ECUC_EthSM_00079`

- ▶ `EthSM_Init()` syntax based on AUTOSAR 4.1.1

Description:

`EthSM_Init()` syntax follows AUTOSAR 4.1.1.

Rationale:

The AUTOSAR 4.1.1 syntax of `EthSM_Init()` allows to pass a configuration to the module during initialization.

Requirements:



SWS\_EthSM\_00043

- ▶ Some transitions not supported

Description:

Some transitions from states ETHSM\_STATE\_WAIT\_TRCVLINK, ETHSM\_STATE\_WAIT\_ONLINE, ETHSM\_STATE\_WAIT\_OFFLINE and ETHSM\_STATE\_ONHOLD are not supported

Requirements:

SWS\_EthSM\_00088, SWS\_EthSM\_00127, SWS\_EthSM\_00128, SWS\_EthSM\_00130, SWS\_EthSM\_00141, SWS\_EthSM\_00143, SWS\_EthSM\_00144, SWS\_EthSM\_00026, SWS\_EthSM\_00160, SWS\_EthSM\_00161, SWS\_EthSM\_00163, SWS\_EthSM\_00165, SWS\_EthSM\_00178, SWS\_EthSM\_00179, SWS\_EthSM\_00181, SWS\_EthSM\_00182, SWS\_EthSM\_00184

- ▶ Internal states ETHSM\_STATE\_WAIT\_CTRLMODEIND and ETHSM\_STATE\_WAIT\_LINK\_DOWN are introduced to the network mode state machine of the EthSM.

Description:

Introduce the both states ETHSM\_STATE\_WAIT\_CTRLMODEIND and ETHSM\_STATE\_WAIT\_LINK\_DOWN to wait for EthSM\_CtrlModeIndication() after a call of EthIf\_SetControllerMode() with ETH\_MODE\_ACTIVE or ETH\_MODE\_DOWN respectively.

Requirements:

SWS\_EthSM\_00178, SWS\_EthSM\_00179, SWS\_EthSM\_00181, SWS\_EthSM\_00182, SWS\_EthSM\_00184, SWS\_EthSM\_00160, SWS\_EthSM\_00161, SWS\_EthSM\_00163, SWS\_EthSM\_00165, SWS\_EthSM\_00088

- ▶ Dem reporting changed

Description:

To allow for debouncing of the ETHSM\_E\_LINK\_DOWN event in the Dem, this event shall be reported as PREFAILED and PREPASSED.

Requirements:

SWS\_EthSM\_00188, SWS\_EthSM\_00196

- ▶ EthSM supports config variant post build

Description:

EthSM supports config variant post build.

Requirements:

## ECUC\_EthSM\_00108\_Conf

### 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.

- ▶ Configuration parameter `EthSMNetwork` limited

Description:

Upper multiplicity of Configuration parameter `EthSMNetwork`, that defines maximum configurable number of Ethernet networks, is set to 255.

### 3.3.3.6. Open-source software

EthSM does not use open-source software.

## 3.3.4. Sd module release notes

- ▶ AUTOSAR R4.2 Rev 1
- ▶ AUTOSAR SWS document version: 4.2.1
- ▶ Module version: 1.4.11.B466224
- ▶ Supplier: Elektrobit Automotive GmbH

### 3.3.4.1. Change log

This chapter lists the changes between different versions.

#### Module version 1.4.11

2021-10-08

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

#### Module version 1.4.10

2021-08-30



- ▶ Added implementation API to provide Instance ID and IP address to an external SWC.

#### **Module version 1.4.9**

2021-06-25

- ▶ Added implementation of DEM error to indicate when a server service is no longer available.
- ▶ Added implementation of a callback to indicate when a malformed Service Discovery message is received.
- ▶ Added implementation of a callback to indicate when a Subscribe Nack entry is received.

#### **Module version 1.4.8**

2021-03-05

- ▶ Implemented Post-build binary (loadable or relocatable) support
- ▶ Implemented Post-build configuration manager support
- ▶ Implemented Retry for EventGroup subscription for OFFERs with TTL infinite
- ▶ Implemented extended SD subscription retry for cyclic offers
- ▶ Implemented SdVersionDrivenFindBehavior flag to handle the acceptance and sending of FIND entries
- ▶ Implemented Blacklisted Minor Versions of a Client Service

#### **Module version 1.4.7**

2020-10-23

- ▶ ASCSD-716 Fixed known issue: The client service UDP unicast end-point is not sent in the subscribe event group response for multicast only consumed event groups.

#### **Module version 1.4.6**

2020-06-19

- ▶ Added SD ConnectionReady functionality to avoid a SD message to be sent before COM stack is ready to process it.
- ▶ Added support for DEM error reporting.

#### **Module version 1.4.5**

2020-02-21

- ▶ Post-build selectable support for ServerServiceAutoAvailable, ServerServiceID, ServerServiceInstanceID, ServerServiceMajorVersion and ServerServiceMinorVersion.



#### **Module version 1.4.4**

2019-11-06

- ▶ ASCSD-635 Fixed known issue: An incoming FIND may not cause the transmission of an OFFER even if the service is provided from the ECU.

#### **Module version 1.4.3**

2019-10-11

- ▶ Checks for SD configuration requirements have been implemented.

#### **Module version 1.4.2**

2019-06-14

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

#### **Module version 1.4.1**

2019-03-06

- ▶ ASCSD-625 Fixed known issue: SD calls SoAd\_GetSoConMode with a wrong SoConId.
- ▶ Added release of unused SocketConnections.

#### **Module version 1.4.0**

2019-02-15

- ▶ ASCSD-574 Fixed known issue: A subscribe may not be sent from the client even if the service was offered.
- ▶ Added Event Subscription via TCP/IP and reception of multiple Server-EventGroups via one multicast SoAd-SocketConnection.

#### **Module version 1.3.19**

2018-10-26

- ▶ ASCSD-538 Fixed known issue: In case of reset of remote connection the local state flags of all server services were not correctly reset.
- ▶ Added TCP datapath support on server side: eventgroup handlers support.
- ▶ Added TCP datapath support on client side: methods support.



- ▶ Added TCP datapath support on client side: consumed eventgroups support.

#### **Module version 1.3.18**

2018-06-22

- ▶ Improved performance by adding a binary search algorithm to look for a match along the list of services.
- ▶ ASCSD-509 Fixed known issue: Multicast IP option of ClientService SubscribeAck cannot be activated.
- ▶ ASCSD-515 Fixed known issue: Reset caused by out of bound access of array Sd\_ServerServiceStates[].

#### **Module version 1.3.17**

2018-02-16

- ▶ Added support for multicast only eventgroups on the server side.
- ▶ Added support for multicast only consumed eventgroups on client side.

#### **Module version 1.3.16**

2018-01-19

- ▶ Added multicast socketconnectiongroup on client side (Seamless service relocation - multiple senders).
- ▶ ASCSD-435 Fixed known issue: Possible index out of bounds exception if a server without UDP ref receives an valid EventrgroupId

#### **Module version 1.3.15**

2017-12-15

- ▶ Added Seamless service relocation - unicast socket connection group on client side

#### **Module version 1.3.14**

2017-09-22

- ▶ Added IPv6 support.

#### **Module version 1.3.13**

2017-07-28

- ▶ Improved performance for code generator.



### Module version 1.3.12

2017-06-30

- ▶ Improved configuration check for SdEventHandler.

### Module version 1.3.11

2017-03-03

- ▶ ASCSD-373 Fixed known issue: Sd fails to compile if no ServerService is configured
- ▶ ASCSD-367 Fixed known issue: Sd wrongly detects a remote node reboot if two consecutive frames with same session ID are received. Updated Release notes for implementation of Sd according to Rfc 76513.
- ▶ Add support for handling OfferService entry when consecutive StopOfferService entry is received before next call of Sd\_Mainfunction.

### Module version 1.3.10

2017-01-05

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

### Module version 1.3.9

2016-11-04

- ▶ Add support for usage of SoAdRoutingGroup with Unicast and Multicast Routing in SdConsumedEvent-GroupUdpActivationRef
- ▶ ASCSD-340 Fixed known issue: Sd uses wrong ModuleId 616 instead of 171

### Module version 1.3.8

2016-08-05

- ▶ Added support for combining as much entries as possible into one SD message.

### Module version 1.3.7

2016-07-01

- ▶ Added support for the reuse of SoAdPduRoute for unicast and multicast datapath. Therefore the multicast SoAdPduRoute shall reference a SoAdSocketConnectionGroup.



### Module version 1.3.6

2016-05-25

- ▶ Added support for unicast flag handling.
- ▶ Added support for race condition prevention in ClientService datapath as proposed in Rfc 73062.
- ▶ Implement transmission of SubscribeEventgroupNack entry if the received SubscribeEventgroup entry does not match to a configured service.

### Module version 1.3.5

2016-03-04

- ▶ Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeader-File`
- ▶ ASCSD-259 Fixed known issue: Invalid memory access for socket connection states
- ▶ Added support for configuration options containing the hostname and the capability records of the services. Service matching algorithm is based on string comparison.

### Module version 1.3.4

2016-02-05

### Module version 1.3.3

2016-01-15

- ▶ ASCSD-209 Fixed known issue: Service discovery detects reboot of remote node by accident
- ▶ Corrected Development error value in `Sd_ConsumedEventGroupSetState`
- ▶ Added support for infinite TTL in `SubscribeEventgroupAck` entry
- ▶ Added support for reboot detection on multiple SD instances

### Module version 1.3.2

2015-11-06

- ▶ ASCSD-168 Fixed known issue: Calling `Sd_ClientServiceSetState()` might result in undefined behavior
- ▶ ASCSD-179 Fixed known issue: Services might use wrong timer configuration values
- ▶ Add config parameter which defines the maximum number of remote nodes supported



- ▶ ASCSD-182 Fixed known issue: Sd client service socket connection might not be opened
- ▶ Implement support for random delays in InitialWait phase and multicast Offer response.
- ▶ Implement support for infinite TTL in SubscribeEventgroup and SubscribeEventgroupAck entry.

#### **Module version 1.3.1**

2015-07-13

- ▶ ASCSD-168 Fixed known issue: Transmission to multicast socket connections does not work

#### **Module version 1.3.0**

2015-06-19

- ▶ Reimplemented handling of received messages and message transmission
- ▶ Implemented basic support for transmission of ServerService Offer with TCP endpoint option
- ▶ Implemented support for seamless service relocation

#### **Module version 1.2.5**

2015-02-20

- ▶ ASCSD-98 Fixed known issue: Sd\_ServerServiceSetState reports an incorrect error Id in case it is called with an invalid SdServerServiceHandleId
- ▶ ASCSD-107 Fixed known issue: Sd does not call SoAd\_DisableRoute() if a server service is halted
- ▶ ASCSD-108 Fixed known issue: When the Sd server service is halted the associated socket connections are not closed
- ▶ ASCSD-112 Fixed known issue: Client Services with infinite TTL in Offer entry expires
- ▶ ASCSD-132 Fixed known issue: Receiving SubscribeEventgroupAck results in wrong behavior

#### **Module version 1.2.4**

2014-10-02

- ▶ ASCSD-75 Fixed known issue: Sd does not handle Endpoint Options correctly on little-endian platforms

#### **Module version 1.2.3**

2014-04-25



- ▶ ASCSD-57 Fixed known issue: Sd server ECU might not handle a restart of a Client ECU correctly

#### **Module version 1.2.2**

2013-10-11

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

#### **Module version 1.2.1**

2013-06-14

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

#### **Module version 1.2.0**

2013-04-08

- ▶ Update to AUTOSAR SWS Service Discovery 1.0.0

#### **Module version 1.1.0**

2012-10-24

- ▶ Update to AUTOSAR SWS Service Discovery 0.4.6

#### **Module version 1.0.0**

2012-07-12

- ▶ First release of EB tresos AutoCore SD module.

#### **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.

- ▶ This module provides no EB-specific enhancements.



### 3.3.4.4. Deviations

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

- ▶ Limited support for Configuration Options

Description:

The `Sd` module supports only one Configuration Option per entry.

Requirements:

`SWS_SD_00664`, `SWS_SD_00665`

- ▶ No support for `Sd` Endpoint Options

Description:

The `Sd` module does not support `Sd` Endpoint Options.

Requirements:

`SWS_SD_00670`, `SWS_SD_00671`, `SWS_SD_00672`, `SWS_SD_00673`, `SWS_SD_00674`, `SWS_SD_00675`, `SWS_SD_00676`, `SWS_SD_00677`, `SWS_SD_00678`, `SWS_SD_00679`, `SWS_SD_00680`, `SWS_SD_00681`, `SWS_SD_00682`, `SWS_SD_00683`, `SWS_SD_00684`, `SWS_SD_00685`, `SWS_SD_00686`, `SWS_SD_00687`

- ▶ No support for AUTOSAR Debugging

Description:

The `Sd` module is not instrumented for the usage with AUTOSAR Debugging.

Requirements:

`SWS_SD_00112`, `SWS_SD_00113`, `SWS_SD_00114`, `SWS_SD_00115`, `SWS_SD_00116`

- ▶ No support for calling `Sd_Init()` during operation

Description:

Calling `Sd_Init()` more than once will reset the internal state of the `Sd` module and will probably leave the Communication stack in an inconsistent state. This means, the following functions will not be called:

- ▶ `BswM_Sd_ClientServiceCurrentState`
- ▶ `BswM_Sd_ConsumedEventGroupCurrentState`
- ▶ `BswM_Sd_EventHandlerCurrentState`
- ▶ `SoAd_CloseSoCon`
- ▶ `SoAd_DisableRouting`



Requirements:

SWS\_SD\_00354

- ▶ No support for verification of topological correctness of endpoints

Description:

The `Sd` module does not check that the IP Addresses received in Endpoint options and SD Endpoint options are topological correct.

Rationale:

The `Sd` module does not check if the remote node is in the same subnet to allow the usage in routed networks.

Requirements:

SWS\_SD\_00688

- ▶ No support for debouncing of Socket Connection interaction

Description:

The `Sd` module does not prevent the consecutive closing and opening of a Socket Connection.

Rationale:

Closing and opening a Socket Connection is debounced in the `SoAd` module.

Requirements:

SWS\_SD\_00696

- ▶ Partial support for identical Consumed Eventgroups

Description:

The `Sd` module does not support the usage of identical configured Consumed Eventgroups on the Client.

Rationale:

In contrast to Unix based systems it is not necessary to subscribe multiple times to the same Consumed Eventgroup to forward the received Events to several Software Components. The `Sd` module supports Event Handler that can be subscribed by a client multiple times.

Requirements:

SWS\_SD\_00693



- ▶ Parameters `SdSubscribeEventgroupRetryMax` and `SdSubscribeEventgroupRetryDelay` can only be set pro Instance

Description:

The `Sd` module does not support a `SdSubscribeEventgroupRetryMax` and `SdSubscribeEventgroupRetryDelay` for each Client Service.

Rationale:

Having to potentially maintain a different timer for each Client Service would have a huge impact on runtime. The `Sd` module optimizes the response time pro instance and subscriptions of OFFERs are packed in a single frame to avoid a burst on the network.

Requirements:

[SWS\\_SD\\_00735](#)

### 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.

- ▶ Limitation on number of entries in container `SdInstance`

Description:

The `Sd` can only handle 254 Service Discovery Instances.

Rationale:

This limitation reduces code size and execution time.

- ▶ Limitation on Eventgroup-subscription for same Unicast Endpoint

Description:

All Eventgroup-subscriptions of an `SdInstance` that shall be transmitted to the same remote Unicast Endpoint must be subscribed by the same `Sd` client.

Rationale:

This limitation reduces memory consumption and execution time.

- ▶ Limitation on number of subscriptions to Eventhandler

Description:



The number of subscriptions to a SdEventHandler ist limited to the number of associated unicast SdSocketConnections.

Rationale:

This limitation reduces memory consumption and execution time.

- ▶ Limitation on socket connection usage

Description:

Socket connections used within service discovery shall not be used with TP.

Rationale:

The usage of TP on socket connections may prevents the configuration of the remote address.

- ▶ Limitation on number of remote nodes

Description:

If the maximum number of remote nodes (configuration parameter SdMaximumRemoteNodes) for an Sd instance is exceeded, Sd messages from further nodes will be discarded.

- ▶ Limitation on capability record strings

Description:

The content of capability record strings is limited to a basic character set excluding quotation marks, escape sequences and trigraphs.

- ▶ Limitation on Configuration Options in SubscribeEventgroupNack entries

Description:

The Sd module does not add config options to SubscribeEventgroupNack entries for SubscribeEventgroup entries with an unknown combination of Service Id, Instance Id, Major Version and referenced Configuration Option or when the SubscribeEventgroup references erroneous options.

- ▶ Limitation on number of nodes hosting the same service instance

Description:

A service instance shall not be hosted by more than one ECU in the network at the same time.

### 3.3.4.6. Open-source software

SD does not use open-source software.



### 3.3.5. SoAd module release notes

- ▶ AUTOSAR R4.2 Rev 2
- ▶ AUTOSAR SWS document version: 4.2.2
- ▶ Module version: 1.8.18.B466224
- ▶ Supplier: Elektrobit Automotive GmbH

#### 3.3.5.1. Change log

This chapter lists the changes between different versions.

##### Module version 1.8.18

2021-10-08

- ▶ ASCSOAD-1534 Fixed known issue: SoConState is not changed to SOAD\_SOCON\_RECONNECT if Tx-Confirmation is handled for UDP SocketConnections.
- ▶ ASCSOAD-1545 Fixed known issue: SoAd disables wrong SoAdSocketRouteDestination.

##### Module version 1.8.17

2021-08-30

- ▶ ASCSOAD-1526 Fixed known issue: TP reception in non header mode may lead to endless loop.
- ▶ ASCSOAD-1538 Fixed known issue: SoAd enables wrong SoAdSocketRouteDestination.
- ▶ Added support for NPDU Buffer Pooling.

##### Module version 1.8.16

2021-06-25

- ▶ Added Support for Multiple Provided Service Instances.
- ▶ Improved error handling to keep TCP connection alive if transmission was aborted and no data was sent.
- ▶ ASCSOAD-1479 Fixed known issue: Out-of-bounds access may occur if Tcplp\_TcpTransmit() returns a value other than TCPIP\_OK.

##### Module version 1.8.15

2021-03-05



- ▶ ASCSOAD-1441 Fixed known issue: Incorrect Tx buffering of UDP frames.

#### **Module version 1.8.14**

2020-10-23

- ▶ ASCSOAD-1411 Fixed known issue: Closing of a UDP socket connection always closes the underlying Tcplp socket.

#### **Module version 1.8.13**

2020-06-19

- ▶ Improved performance of SoAd\_IfTransmit().
- ▶ ASCSOAD-1337 Fixed known issue: SoAd generation error occurs due to PduLength bigger than 65535.
- ▶ ASCSOAD-1358 Fixed known issue: A socket connection group with TRIGGER\_NEVER and TRIGGER\_ALWAYS Tx PDUs may cause an out-of-bounds access.
- ▶ Added API SoAd\_IsConnectionReady.
- ▶ ASCSOAD-1338 Fixed known issue: SoAdRouteMax set to INDEX\_UINT8 can lead to incorrect mapping between PduRouteDestVirtualIds and SoCons.

#### **Module version 1.8.12**

2020-02-21

- ▶ ASCSOAD-1296 Fixed known issue: SoAd may not choose an ephemeral local port if SoAdSocketLocalPort is configured with zero.
- ▶ ASCSOAD-1312 Fixed known issue: Compilation error occurs when PostBuild variants are configured.
- ▶ ASCSOAD-1318 Fixed known issue: SoAdPduRoutes are not enabled correctly at initialization on big-endian platform.
- ▶ ASCSOAD-1323 Fixed known issue: TCP client may not immediately connect to TCP server.
- ▶ ASCSOAD-1306 Fixed known issue: SoAdRelocatableCfgEnable set to true can lead to wrong socket connection mapping.
- ▶ ASCSOAD-1313 Fixed known issue: Discarding of received frames can cause an out-of-bounds access.

#### **Module version 1.8.11**

2019-10-11



- ▶ ASCSOAD-1264 Fixed known issue: Java exception occurs if "SoAdSocketnPduUdpTxBufferMin" is enabled.
- ▶ ASCSOAD-1274 Fixed known issue: Wrong initialization of SoAd PBRAM.
- ▶ Added caching for XPath function getSoAdRoutesForPdu().
- ▶ Fixed IPv6 address detection logic in code generator.
- ▶ Improved file structure.
- ▶ ASCSOAD-1276 Fixed known issue: The reception of a UDP frame may cause an out-of-bounds access.
- ▶ ASCSOAD-1293 Fixed known issue: Reception ring buffer can cause an out-of-bounds access.

### Module version 1.8.10

2019-09-06

- ▶ Improved performance of checks that verify consistency of SoAd and Sd configuration data.
- ▶ Improved performance of Routing Group Handling in SoAd for SoAd\_IfTransmit.
- ▶ Added functionality to skip If Tx confirmation handling at PDU level.

### Module version 1.8.9

2019-07-05

- ▶ ASCSOAD-1198 Fixed known issue: SoAd calls Tcplp APIs with invalid socket ID if socket gets closed while data are buffered.
- ▶ ASCSOAD-1215 Fixed known issue: Reception of TCP frame with length in header bigger than 64 KB may cause an out-of-bounds read access.
- ▶ ASCSOAD-1130 Fixed known issue: SoAd blocks transmission on TCP socket connection if Tcplp\_TcpTransmit() returns a value other than E\_OK.

### Module version 1.8.8

2019-06-14

- ▶ ASCSOAD-1131 Fixed known issue: Remote address of first socket connection is locked with SoAd\_SetUniqueRemoteAddr() instead of the matching socket connection in this socket connection group.
- ▶ ASCSOAD-1142 Fixed known issue: Unintended reset of remote address for Tcp and Udp socket connections if UdpSupervisionAliveTimeout is used for any socket connection.
- ▶ ASCSOAD-1173 Fixed known issue: Buffer overflow can be triggered on reception of an invalid frame.
- ▶ Improved SoAd\_MainFunction() runtime at idle time.



- ▶ ASCSOAD-1199 Fixed known issue: SoAd\_ReleaseRemoteAddr doesn't reset the remote address immediately if called by upper layer in context of SoAd\_RxIndication().

### Module version 1.8.7

2019-03-07

- ▶ ASCSOAD-1053 Fixed known issue: SoAd triggers SEGMENT FAULT on reception of TP PDUs on TCP connection with enabled header mode.
- ▶ Added full support of SoAd\_ReleaseRemoteAddr(). Important note: This update of SoAd requires an ACG 8 Sd version 1.4.1 or later
- ▶ ASCSOAD-1105 Fixed known issue: TcplpEvent() called with TCPIP\_UDP\_CLOSED sets transition change only for first SoCon in UDP SoConGroup.

### Module version 1.8.6

2019-02-22

- ▶ ASCSOAD-1050 Fixed known issue: SoAd triggers SEGMENT FAULT on TP routing.

### Module version 1.8.5

2019-02-15

- ▶ Added support of double buffering for concurrent If Tx PDUs using the same UDP socket.
- ▶ Added support to disable SoAdSocketTpRxBufferMin even for TCP connections and PDUs using Tp API.
- ▶ ASCSOAD-1050 Fixed known issue: No subscriptions are sent for SD OFFER messages received while client service is in DOWN state and socket connection is closed.
- ▶ Improved RAM usage by splitting internal data structure.

### Module version 1.8.4

2019-01-24

- ▶ ASCSOAD-1057 Fixed known issue: SoAdSocketRoutes are not sorted consecutively based on header ID.

### Module version 1.8.3

2018-12-13



- ▶ Added support for TLS extension.
- ▶ Added support measurement data.

### Module version 1.8.2

2018-10-26

- ▶ ASCSOAD-901 Fixed known issue: SoAdTxUdpTriggerMode TRIGGER\_NEVER does not work in combination with IP fragmentation.
- ▶ Added support for SoAd\_IfTransmit() with SduDataPtr = NULL\_PTR on a UDP connection to retrieve data with Up\_[SoAd][If]TriggerTransmit().
- ▶ ASCSOAD-930 Fixed known issue: SoAd may corrupt the last TCP segment of a received PDU if header mode and IF API are used.

### Module version 1.8.1

2018-09-20

- ▶ Added Post-build selectable support.
- ▶ ASCSOAD-922 Fixed known issue: Wrong PDU header ID may be inserted before the PDU is transmitted via a socket connection.
- ▶ Added Post-build binary support.

### Module version 1.8.0

2018-06-22

- ▶ ASCSOAD-834 Fixed known issue: The nUdpPduBuffer is never triggered by timeout if SoAdTxUdpTriggerTimeout is less than SoAdMainFunctionPeriod.
- ▶ Separated the transmit part of SoAd\_MainFunction to SoAd\_MainFunctionTx and added an option to make the latter externally callable.
- ▶ Optimized handling of client/server configuration using meta data handling.

### Module version 1.7.24

2018-05-30

- ▶ ASCSOAD-823 Fixed known issue: Updating LAST-IS-BEST PDU could lead to buffer inconsistency.
- ▶ Internal module improvement. This module version update does not affect module functionality.



### Module version 1.7.23

2018-03-16

- ▶ ASCSOAD-813 Fixed known issue: A UDP Tx buffer is provided only for the first socket connection if a PDU route destination refers to the socket connection group.

### Module version 1.7.22

2018-02-16

- ▶ Implemented N:1 routing.
- ▶ ASCSOAD-778 Fixed known issue: SoAd sends incorrect data if SoAdTxPduCollectionSemantics is set to last-is-best.
- ▶ ASCSOAD-787 Fixed known issue: Incorrect configuration check for number of configured UDP sockets for IPv6.
- ▶ ASCSOAD-788 Fixed known issue: Incorrect configuration check for number of configured TCP sockets for IPv6.

### Module version 1.7.21

2017-12-15

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

### Module version 1.7.20

2017-11-17

- ▶ Updated TpTxConfirmation() to trigger another transmission for the same PDU.

### Module version 1.7.19

2017-09-22

- ▶ Added support for SoAdTxPduCollectionSemantics(last-is-best vs. queued).
- ▶ Added API SoAd\_ReleaseRemoteAddr() to reset the remote address.
- ▶ Updated to MISRA 2012
- ▶ ASCSOAD-640 Fixed known issue: Message lost or corruption with multiple PduRouteDest + nPduUdp-TxBuffer + TRIGGER\_NEVER.



#### **Module version 1.7.18**

2017-08-25

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

#### **Module version 1.7.17**

2017-07-28

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

#### **Module version 1.7.16**

2017-06-30

- ▶ Time consumption in matching InternalRoutingGroup and ExternalRoutingGroup.
- ▶ ASCSOAD-638 Fixed known issue: SoAd might reject API calls for specific routing groups.

#### **Module version 1.7.15**

2017-05-05

- ▶ Added support of post build RAM greater than 64kB.

#### **Module version 1.7.14**

2017-03-03

- ▶ ASCSOAD-605 Fixed known issue: Invalid memory access occurs if a single routing group is configured.

#### **Module version 1.7.13**

2017-02-03

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

#### **Module version 1.7.12**

2016-12-14



- ▶ ASCSOAD-572 Fixed known issue: A global routing group is treated as specific routing group if reference to SocketConnectionGroup exists for this routing group.
- ▶ ASCSOAD-588 Fixed known issue: SoAd sends frames to wrong destination if initiated by SoAd\_IfSpecificRoutingGroupTransmit().

#### **Module version 1.7.11**

2016-12-02

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

#### **Module version 1.7.10**

2016-11-04

- ▶ Updated handling of Tcplp\_ReturnType according to AUTOSAR 4.2.2.

#### **Module version 1.7.9**

2016-10-07

- ▶ Replaced config parameter SoAdSoConMax and SoAdRoutingGroupMax with SoAdSoConIdType and SoAdRoutingGroupIdType.
- ▶ ASCSOAD-527 Fixed known issue: SoAd tries to request already provided data from Tp upper layer if UDP retry is performed.
- ▶ SoAd\_CloseSoCon() called with Abort = FALSE shall not stop ongoing Rx/Tx.
- ▶ ASCSOAD-526 Fixed known issue: Trigger transmit requests are handled incorrect if specific routing groups are triggerable and have multiple socket connections.
- ▶ Added support for Tcplp config schema according to AUTOSAR 4.2.2.

#### **Module version 1.7.8**

2016-09-09

- ▶ ASCSOAD-499 Fixed known issue: No UDP retry might performed if buffer gets overwritten by newer frame.
- ▶ Added Udp alive supervision feature.
- ▶ ASCSOAD-500 Fixed known issue: Reception of invalid header values leads to invalid memory access.
- ▶ ASCSOAD-501 Fixed known issue: SoAd fills segmented Rx TCP PDUs with wrong data.



- ▶ ASCSOAD-502 Fixed known issue: SoAd transmits invalid payload.

#### **Module version 1.7.7**

2016-08-05

#### **Module version 1.7.6**

2016-07-01

- ▶ ASCSOAD-449 Fixed known issue: The call of SoAd\_CloseSoCon() closes an active TCP connection with RST instead of FIN.
- ▶ ASCSOAD-454 Fixed known issue: Preemption of SoAd\_IfTransmit() causes a copy attempt from NULL\_-PTR.

#### **Module version 1.7.5**

2016-05-25

- ▶ ASCSOAD-433 Fixed known issue: Tcplp\_Close() is called twice for the listen socket.
- ▶ ASCSOAD-425 Fixed known issue: The call of SoAd\_CloseSoCon() and SoAd\_TcplpEvent(TCPIP\_TCP/UDP\_CLOSED) within the same main function triggers Tcplp\_Close() with invalid socket ID.

#### **Module version 1.7.4**

2016-04-29

- ▶ Updated memory section macros to AUTOSAR 4.0 naming convention.
- ▶ ASCSOAD-414 Fixed known issue: SoAdPduRoute of specific routing groups try to send over the first socket connection in group only.
- ▶ ASCSOAD-416 Fixed known issue: SoAd might confirm successful TP interface transmissions with error code NTFRSLT\_E\_NOT\_OK.
- ▶ ASCSOAD-418 Fixed known issue: SoAd writes to wrong memory location when Rx only socket connection is closed.

#### **Module version 1.7.3**

2016-04-01



- ▶ Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeader-File`.
- ▶ ASCSOAD-388 Fixed known issue: SoAd does not handle a return value other than E\_OK for `Tcplp_Bind()`, `Tcplp_TcpListen()`, and `Tcplp_TcpConnect()` correctly.
- ▶ ASCSOAD-399 Fixed known issue: SoAd rejects API calls for specific routing groups.

### Module version 1.7.2

2016-02-05

- ▶ Added check when a `SoAdRoutingGroup` is not referenced by any `SoAdSocketRoute` or `SoAdPduRoute`.
- ▶ Implemented support for connection specific TCP Keep-Alive configuration.

### Module version 1.7.1

2015-11-06

- ▶ ASCSOAD-323 Fixed known issue: SoAd might not be able to reestablish a TCP connection if it was previously closed by the remote node.
- ▶ ASCSOAD-325 Fixed known issue: Inconsistent SoAd configuration might lead to EB tresos Studio errors.
- ▶ ASCSOAD-326 Fixed known issue: SoAd might enable, disable or trigger a different `SoAdRoutingGroup` then intended.
- ▶ ASCSOAD-322 Fixed known issue: SoAd calls of `Tcplp_DhcpRead/WriteOption()` pass wrong parameter formats and wrong option values.
- ▶ ASCSOAD-334 Fixed known issue: `SoAd_ChangeParameter()` calls `Tcplp_ChangeParameter()` with wrong `SocketId` parameter value.
- ▶ ASCSOAD-333 Fixed known issue: SoAd might close wrong socket instead of listen socket.
- ▶ ASCSOAD-335 Fixed known issue: The SoAd incorrectly reports a configuration error if all `SoAdRoutingGroups` are triggerable.
- ▶ Created recommended configuration.
- ▶ ASCSOAD-340 Fixed known issue: The call of `SoAd_LocallpAddrAssignmentChg()` with `TcplpAddrId` which is not referred by SoAd leads to DET error report if enabled or might cause invalid memory access if disabled.
- ▶ ASCSOAD-342 Fixed known issue: SoAd uses wrong destination port for outgoing UDP/IPv6 datagrams.

### Module version 1.7.0

2015-06-19



- ▶ Changed internal type for `Tcplp_SocketIdType` from `uint8` to `uint16`, thus supporting `Tcplp`-stacks providing socketIds larger than 255.
- ▶ Added Seamless Service Relocation Support.

#### **Module version 1.6.5**

2015-02-20

- ▶ ASCSOAD-296 Fixed known issue: SoAd might stop forwarding received data on TCP connections.
- ▶ ASCSOAD-297 Fixed known issue: SoAd might not recover from an aborted TCP connection attempt.

#### **Module version 1.6.4**

2014-10-02

- ▶ Added support of IPv6.
- ▶ Changed module to include only `Tcplp.h` as `Tcplp` module interface file.
- ▶ ASCSOAD-284 Fixed known issue: SoAd might perform an invalid memory access in case upper layers use indirect data provision.

#### **Module version 1.6.3**

2014-04-25

- ▶ Added support of large (`length > 255 Byte`) PDU transmission via `IfRoutingGroupTransmit`.

#### **Module version 1.6.2**

2013-10-11

- ▶ ASCSOAD-249 Fixed known issue: SoAd might not acknowledge TCP data during disconnection and delay the closure procedure.
- ▶ ASCSOAD-254 Fixed known issue: SoAd might transmit invalid `PduHeader` information on TCP socket connections that use the Nagle algorithm.
- ▶ Added support of `nPduUdpTxBuffer` functionality.

#### **Module version 1.6.1**

2013-06-14

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



### Module version 1.6.0

2013-04-08

- ▶ ASCSOAD-220 Fixed known issue: SoAd might perform an invalid reading memory access if a socket connection is only configured for transmissions.
- ▶ ASCSOAD-221 Fixed known issue: `SoAd_SetRemoteAddr()` refuses the request for open UDP socket connections with disabled PDU header mode.
- ▶ Added support for multiple PDU routing group references in `SocketRoutes` and `PduRoutes`.
- ▶ ASCSOAD-232 Fixed known issue: SoAd configuration does not allow UDP and TCP `SocketConnection` with the identical local address and local port.
- ▶ Added support of PDU transmission via `IfRoutingGroupTransmit` API.
- ▶ ASCSOAD-237 Fixed known issue: SoAd might not receive UDP datagrams in `SoAdSocketConnectionGroups` with more than one `SoAdSocketConnection`.

### Module version 1.5.0

2013-02-20

- ▶ ASCSOAD-175 Fixed known issue: SoAd might not compile if an upper layer provides only one of the notification APIs `SoAdSoConModeChg` and `SoAdLocalIpAddrAssignmentChg`.
- ▶ ASCSOAD-176 Fixed known issue: SoAd might not reopen a TCP listen socket connection correctly.
- ▶ ASCSOAD-186 Fixed known issue: `SoAd_GetLocalAddr` does not provide local port information.
- ▶ Added support of multiple UDP `SocketConnections` per `SocketConnectionGroup`.
- ▶ ASCSOAD-209 Fixed known issue: SoAd might not correctly process multiple PDUs within one received frame.
- ▶ Added support of PDU routing groups.
- ▶ Updated AUTOSAR SWS SocketAdaptor 2.0.24 R4.1 Rev 1.
- ▶ Implemented provision of `<Up> [SoAd] [Tp] TxConfirmation()` for UDP transmissions in the context of `SoAd_MainFunction`.
- ▶ ASCSOAD-211 Fixed known issue: SoAd might transmit invalid data if the UDP retry feature is enabled.
- ▶ Added support of TCP immediate TP transmit confirmation.

### Module version 1.4.1

2012-10-16

- ▶ ASCSOAD-144 Fixed known issue: SoAd might perform an invalid memory access in case of a UDP socket connection for which no TP receive buffer is configured.



- ▶ ASCSOAD-147 Fixed known issue: Upper Layer of SoAd might not receive the full TP-PDU in case of a UDP socket connection.
- ▶ ASCSOAD-152 Fixed known issue: Upper Layer of SoAd might not receive the full TP-PDU in case the upper layer accepts only a part of the TP-PDU.
- ▶ ASCSOAD-153 Fixed known issue: Upper layer of SoAd might receive invalid data in case the upper layer accepts only a part of the TP-PDU and a new TP-PDU is received by SoAd in the meantime.
- ▶ ASCSOAD-155 Fixed known issue: SoAd might not be able to reopen a socket connection if it was closed by Tcplp module with `TCPIP_TCP_RESET`.
- ▶ Added qualifier `const` to parameter `PduInfoPtr` of `CopyRxData`.
- ▶ Updated AUTOSAR SWS SocketAdaptor 2.0.19 R4.1 Rev 0

#### Module version 1.4.0

2012-09-18

- ▶ Updated Tcplp types (`TcpIp_ParamIdType`, `TcpIp_DomainType`, `TcpIp_ProtocolType`).
- ▶ Added `TxQuota` functionality.
- ▶ Added name macros for configuration parameters which have `SYMBOLICNAMEVALUE` set to true.

#### Module version 1.3.1

2012-07-31

- ▶ Updated PDU fan-out TX (i.e. more than one `SoAdPduRouteDest` per `SoAdPduRoute`).

#### Module version 1.3.0

2012-07-13

- ▶ ASCSOAD-115 Fixed known issue: TCP socket connection with PDU header mode disabled might be blocked for upper layers with TP-API.
- ▶ Updated signature of `TcpIp_UdpTransmit` and `TcpIp_TcpTransmit`.
- ▶ Updated SoAd config to SWS 2.0.13.

#### Module version 1.2.2

2012-06-27

- ▶ Added support of immediate shutdown with `SoAd_CloseSoCon`.



- ▶ Implemented SoAd extension for better support of `UdpNm`.
- ▶ Added support of `UdpRetry` functionality.
- ▶ Added support of multiple TCP `SocketConnections` per `SocketConnectionGroup`.
- ▶ ASCSOAD-107 Fixed known issue: `SoAdTp_Transmit()` cannot be called within `<Up>_[SoAd]` [`Tp`] `TxConfirmation()` for the same PDU related to a TCP socket connection.
- ▶ ASCSOAD-108 Fixed known issue: SoAd might handle socket connections incorrectly if the related socket is closed by Tcplp.

#### **Module version 1.2.1**

2012-06-01

- ▶ Added SoAd TP Rx-PDU cancellation support.
- ▶ ASCSOAD-86 Fixed known issue: SoAd requires transmit confirmation enabled for upper layers with If-API.

#### **Module version 1.2.0**

2012-05-21

- ▶ Added SoAd TP support.
- ▶ Updated SoAd config to SWS 2.0.9.

#### **Module version 1.1.0**

2012-04-13

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

#### **Module version 1.0.0**

2012-03-16

- ▶ Initial AUTOSAR 4.0 version.

#### **3.3.5.2. New features**

- ▶ Added Support for Multiple Provided Service Instances.



- ▶ Added support for NPDU Buffer Pooling.

### 3.3.5.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- ▶ Configurable reporting of `SOAD_E_INV_PDUHEADER_ID` to DET

Description:

A pre-compile time configuration parameter is provided that enables/disables the reporting of `SOAD_E_INV_PDUHEADER_ID` to DET.

- ▶ `SoAd_GetLocalAddr()` allows to retrieve the address family

Description:

The call of `SoAd_GetLocalAddr()` with domain of the local address set to `TCPIP_AF_UNSPEC` will return the domain of the configured address family.

- ▶ API support of `SoAd_ReleaseRemoteAddr()` to reset the remote address.

Description:

The call of `SoAd_ReleaseRemoteAddr()` can be used to reset the remote address of a UDP connection immediately to configured value.

- ▶ Support to increase Tx Tp frame processing

Description:

`SoAd` provides `SoAdEnableMainFunctionTx` which enables the TP transmit section of `SoAd_MainFunction()` to be callable externally via `SoAd_MainFunctionTx()`. Additional calls of `SoAd_MainFunctionTx()` speed up the Tx Tp frame processing.

- ▶ Support of double buffering Udp Tx If frame

Description:

`SoAd` uses the nPDU buffer `SoAdSocketnPduUdpTxBufferMin` to store incoming Udp If PDUs if it interrupts a transmission on the corresponding Udp socket. In case that the interrupt occurs during transmission of current nPDU buffer, the remaining buffer space is used as temporary buffer.

Rational:

According to [https://bugzilla.autosar.org/show\\_bug.cgi?id=59416](https://bugzilla.autosar.org/show_bug.cgi?id=59416) `TcpIp_UdpTransmit()` is not reentrant for the same UDP socket. If Tx PDUs for the same Udp socket interrupt each other, SoAd needs either to reject the Tx request or to buffer them.



- ▶ Support of Udp Tx If PDU bigger than buffer

Description:

If a Udp Tx If PDU does not fit into the buffer `SoAd` will try to transmit it. In case that transmission got rejected by `TcpIp`, then `SoAd` will reject the PDU.

- ▶ Support to disable `SoAdSocketTpRxBufferMin` even for TCP connections and PDUs using Tp API

Description:

If upper layer always accepts received data for a TCP connection and PDUs with Tp API, it is no longer mandatory to configure a reception buffer. The data will be provided in context of `SoAd_RxIndication()`. In case that no buffer is configured and upper layer rejects data reception, then either the PDU gets dropped if header mode is disabled or the TCP connection gets closed if header mode is disabled.

- ▶ Disabling of If Tx confirmation handling

Description:

The If Tx confirmation handling is a time consuming task which can affect the performance. This can be optimized either by disabling Tx confirmation for a complete upper layer by disabling the config parameter `SoAdBswModules/SoAdIfTxConfirmation` or at PDU level by disabling the config parameter `SoAdConfig/SoAdPduRoute/SoAdSkipIfTxConfirmation`. Both measures reduce the main function execution time.

- ▶ `SoAd_IsConnectionReady` API

Description:

The call of `SoAd_IsConnectionReady()` can be used to check if an ARP entry or IpSec SA exists for this socket connection.

### 3.3.5.4. Deviations

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

- ▶ Unsupported SoAd Features

Description:

SoAd does not yet support the following features:

- ▶ Resource Management
- ▶ Udp StrictHeaderLen Check
- ▶ Socket connection notification enabling/disabling

`SWS_SoAd_00649, SWS_SoAd_00125, SWS_SoAd_00126, SWS_SoAd_00597`



► Initialization check in main function

Description:

If the main function is called while the module is not yet initialized, the main function returns immediately without performing any functionality and without raising any Det error. This initialization check is always performed independent of the development error detection setting.

Rationale:

The SchM module may schedule the modules main function before the module is initialized. This would result in lots of Det errors during start up. Therefore the module's main function does not throw a Det error if the module is not yet initialized and simply returns in this case.

► Configuration check

Description:

`SoAd_Init()` does not check the `SoAdConfigPtr` for containing a valid configuration. Instead it will perform a basic `NULL_PTR` check.

Rationale:

SoAd configuration does not include signatures to verify if a configuration is valid or not. A valid configuration is expected.

SWS\_SoAd\_00216

► Symbolic name for `SoAdSocketId` and `SoAdRxPduId` do not follow the AUTOSAR naming scheme

Description:

The symbolic name macros for `SoAdSocketId` and `SoAdRxPduId` do not follow the AUTOSAR 4.0 Rev 3 naming scheme. AUTOSAR only specifies about the inclusion of the name of direct parent in the symbolic macros but instead the macros are generated as follows: `SoAdSocketId`: `SoAdConf_[SoAdSocketConnectionGroup]_[SoAdSocketConnection]` `SoAdRxPduId`: `SoAdConf_[SoAdSocketRoute]_[SoAdSocketRouteDest]`

Rationale:

`SoAdSocketConnection` short names are only distinct within the context of the superior `SoAdSocketConnectionGroup`. The generation of `SymbolicName` macros as specified within the ECU configuration specification could lead to multiple macro redefinitions.

`SoAdSocketRouteDest` short names are only distinct within the context of the superior `SoAdSocketRoute`. The generation of `SymbolicName` macros as specified within the ECU configuration specification could lead to multiple macro redefinitions.



- ▶ SoAd does not support <Up>\_[SoAd][Tp]StartOfReception API according to AUTOSAR 4.1.1 and higher

Description:

SoAd does not support the extended StartOfReception API that was introduced in all TP related modules in AUTOSAR version 4.1.1. The StartOfReception API was extended with the new parameter PduInfoType\* info.

Rationale:

This deviation is required for the compatibility to AUTOSAR version prior to 4.1.1. Thus RFC [https://www-autosar.org/bugzilla/show\\_bug.cgi?id=47264](https://www-autosar.org/bugzilla/show_bug.cgi?id=47264) is not considered.

- ▶ SoAd\_LocalIpAddrAssignmentChg() silently ignores invalid a call with not expected TcpIpAddrId

Description:

If Tcplp calls SoAd\_LocalIpAddrAssignmentChg() with not expected TcpIpAddrId, the function shall not call Det but silently ignore the call.

Rationale:

Tcplp calls SoAd\_LocalIpAddrAssignmentChg() for all configured TcpIpLocalAddr even if SoAd does not refer to it. SoAd tolerate such calls and silently ignore not expected IDs. See also [https://www-autosar.org/bugzilla/show\\_bug.cgi?id=71116](https://www-autosar.org/bugzilla/show_bug.cgi?id=71116).

Requirements:

SWS\_SoAd\_00280

- ▶ RequestIpAddrAssignment() syntax according to AUTOSAR 4.2.2 not supported

Description:

The syntax of SoAd\_RequestIpAddrAssignment() does not contain the parameters Netmask, DefaultRouterPtr introduced by AUTOSAR 4.2.2. SoAd also calls Tcplp\_RequestIpAddrAssignment() with syntax based on AUTOSAR 4.1.1. Thus RFC [https://www.autosar.org/bugzilla/show\\_bug.cgi?id=62672](https://www.autosar.org/bugzilla/show_bug.cgi?id=62672) is not not considered.

Rationale:

The syntax of RequestIpAddrAssignment() based on AUTOSAR 4.1.1 is kept for backwards compatibility.

Requirements:

SWS\_SoAd\_00520

- ▶ SoAd\_TpChangeParameter() API not supported



Description:

The API `SoAd_TpChangeParameter()` is not supported. Instead the EB specific API `SoAd_ChangeParameter()` can be used to change Tcplp specific parameters.

Requirements:

SWS\_SoAd\_00508, SWS\_SoAd\_00630, SWS\_SoAd\_00631

- ▶ A socket connection leaving the state `SOAD_SOCON_OFFLINE` always reports `SOAD_SOCON_RECONNECT`

Description:

If during the call of `SoAd_MainFunction()` a socket connection leaves the state `SOAD_SOCON_OFFLINE`, the module always calls `Up_SoConModeChg()` with the state `SOAD_SOCON_RECONNECT` to the upper layer. If the precondition are fulfilled to reach `SOAD_SOCON_ONLINE`, the module calls `Up_SoConModeChg()` again within the same `SoAd_MainFunction()`.

Rationale:

The `SoAd_MainFunction()` can handle multiple socket connection state transition within one call. Therefore, direct transition from `SOAD_SOCON_OFFLINE` to `SOAD_SOCON_ONLINE` does not provide any benefit but would required extra treatment.

Requirements:

SWS\_SoAd\_00591

- ▶ `SoAdSocketTcpNoDelay` not supported.

Description:

The API `TcpIp_ChangeParameter()` will not be called to change `TCPIP_PARAMID_TCP_NAGLE` when allocating a new socket.

Requirements:

SWS\_SoAd\_00689

- ▶ SoAd supports AUTOSAR 4.0.3 EcuC references only.

Description:

SoAd only supports EcuC references according to AUTOSAR 4.0.3. The container `EcuCConfigSet` is not part of the EcuC path.

Rationale:



ECuC is based on AUTOSAR 4.0.3 to stay compatible to other AUTOSAR 4.0.3 modules. Thus RFC [https://www.autosar.org/bugzilla/show\\_bug.cgi?id=53369](https://www.autosar.org/bugzilla/show_bug.cgi?id=53369) is not considered.

Requirements:

ECUC\_SoAd\_00038, ECUC\_SoAd\_00030

- ▶ SoAd only support <Up>\_[SoAd] [If] RxIndication API according to AUTOSAR 4.1.2

Description:

SoAd uses the API syntax of AUTOSAR version 4.1.2 of <Up>\_[SoAd] [If] RxIndication which does not require a constant pointer for the parameter PduInfoPtr.

Rationale:

This deviation is required for the compatibility to upper layer modules with AUTOSAR version prior to 4.1.3.

Requirements:

SWS\_SoAd\_00106

- ▶ Configuration parameters SoAdSoConMax and SoAdRoutingGroupMax unused.

Description:

The configuration parameters SoAdSoConMax and SoAdRoutingGroupMax obsolete.

Rationale:

SoAd allows to change the number of socket connection and routing groups at post build time without the need of limits at precompile time. The number is only limited by the used type as well as the reserved post build RAM.

Requirements:

ECUC\_SoAd\_00127, ECUC\_SoAd\_00126, SWS\_SoAd\_00518, SWS\_SoAd\_00519

- ▶ Initialization check in main function

Description:

If the main function is called while the module is not yet initialized the main function returns immediately without performing any functionality and without raising any Det error. This initialization check is always performed independent of the development error detection setting.

Rationale:



The RTE module may schedule the modules main function before the module is initialized. This would result in lots of Det errors during start up. Therefore the module's main function does not throw a Det error if the module is not yet initialized and simply returns in this case.

Requirements:

SWS\_SoAd\_00283

- ▶ Limited Det error reporting

Description:

In the context of `SoAd_RxIndication()` the module only reports `SOAD_E_INV_ARG` if the parameter `RemoteAddrPtr` is `NULL_PTR`. In case that the remote address is valid but there is no match, the frame will be discarded.

Requirements:

SWS\_SoAd\_00268

- ▶ Pdu fan out not supported for TP

Description:

Fan out(i.e. multiple SoAdPduRouteDest specified) is only supported for upper layers with IF APIs.

Requirements:

SWS\_SoAd\_00561

- ▶ SoAd does not support IP fragmentation in combination with functionality Last-Is-Best

Description:

SoAd does not support Ip fragmentation in combination with Last-Is-Best nPDU transmission.

Requirements:

SWS\_SoAd\_00543

- ▶ SoAd supports `SoAdSocketUdpAliveSupervisionTimeout` handling according to AUTOSAR 4.3.0

Description:

In order to support "Release of unused socket connections" functionality, SoAd starts the Udp alive supervision timer according to AUTOSAR 4.3.0 requirements `SWS_SoAd_00694` and `SWS_SoAd_00742` instead of AUTOSAR 4.2.2 requirements `SWS_SoAd_00694`. Therefore, the Udp alive timer is started and updated after each received Udp frame which passes the message acceptance filter.

Rationale:



This deviation is required to support the "Release of unused socket connections" functionality. Otherwise, no Udp alive timer is started if a method call is received after the remote address was set with `SoAd_SetUniqueRemoteAddr()`. Hence, the call of `SoAd_ReleaseRemoteAddr()` would immediately release the remote address and interrupt an ongoing method call.

Requirements:

`SWS_SoAd_00694`

- ▶ `SoAd` does not forward the parameter `Abort` for UDP sockets.

Description:

`SoAd` does not forward the parameter `Abort` from `SoAd_CloseSoCon()` to `TcpIp_Close()` for UDP sockets. Instead `Abort` will always be FALSE for `TcpIp_Close()` of UDP sockets.

Rationale:

In case of UDP the call of `TcpIp_Close()` belongs to a group of socket connection. Only the last socket connection which gets closed will call `TcpIp_Close()`. Passing the parameter `Abort` of `SoAd_CloseSoCon()` call from the last socket connection would not reflect the other socket connections. Furthermore, `TcpIp` only specifies the usage of `Abort` for TCP sockets and ignores `Abort` for UDP sockets.

Requirements:

`SWS_SoAd_00642`

- ▶ State transition for UDP socket connection of type automatic with wildcard remote address is handled in next `SoAd_MainFunction` instead of directly before transmit confirmation function is called.

Description:

For a UDP socket connection of type automatic i.e. configuration parameter `SoAdSocketAutomaticSoConSetup` set to TRUE) which uses a wildcard in the configured remote address (i.e. an ANY-String for IP address or port), `SoAd` is not changing the state of the socket connection to `SOAD_SOCON_RECONNECT` directly before the related transmit confirmation function is called (or would be called if such a function is not configured). Instead the state transition is handled in the next `SoAd_MainFunction`.

Rationale:

To improve run-time performance and avoid preemption issues, the state transition will be handled in the next `SoAd_Mainfunction` after transmit confirmation function is called.

Requirements:

`SWS_SoAd_00582`



### 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.

- ▶ No support for multiple configurations

Description:

Only one configuration is supported. Multiple configurations are not allowed.

- ▶ Reception of a segmented PDU via a disabled `SoAdSocketRoute` causes reconnection

Description:

When only a segment of a PDU is received that is part of a disabled `SocketRoute` (i.e. all related `RoutingGroups` are disabled), the TCP socket connection is closed and reopened within the next `MainFunction()`. This behavior is similar to TP receptions that are refused by the upper layer and prevents the remote node from sending huge amounts of data that is not of interest for the upper layer anyway.

- ▶ `SoAdSocketTcpKeepAliveProbesMax` is limited to `uint8`

Description:

`SoAdSocketTcpKeepAliveProbesMax` can have values from 1 to 255 only.

- ▶ Only one `TcpIp` client is allowed for the same local IP address and port.

Description:

If `SoAdSocketTcpInitiate` is set to true, the socket connection group can only contain one socket connection.

- ▶ Linking of socket routes to TCP socket connection group is not supported.

Description:

It is not supported to link `SoAdRxSocketConnOrSocketConnBundleRef` to a `SoAdSocketConnectionGroup` with `SoAdSocketProtocol = SoAdSocketTcp` Only direct linkage to `SoAdSocketConnection` are supported.

- ▶ `SoAd_IfTransmit()` with `SduDataPtr = NULL_PTR` is not supported for TCP connections.

Description:

`SoAd` does not support calls of `SoAd_IfTransmit()` with `SduDataPtr = NULL_PTR` for TCP connections. In this case a Det is reported.

- ▶ `SoAdGetAndResetMeasurementDataApi` is not supported in ACG 8.5.2

Description:



The functionality of `SoAdGetAndResetMeasurementDataApi` is not supported in ACG 8.5.2. The config parameter shall not be enabled.

- ▶ `SoAdTlsEnabled` is not supported in ACG 8.5.2

Description:

The functionality of `SoAdTlsEnabled` is not supported in ACG 8.5.2. The config parameter shall not be enabled.

- ▶ Pdu length maximum if PduHeaderMode is enabled.

Description:

To support buffering of PDUs if PduHeaderMode is enabled, the Pdu length is limited to `PDULengthType-MAX - PduHeaderLength`.

### 3.3.5.6. Open-source software

SoAd does not use open-source software.

## 3.3.6. SomelpTp module release notes

- ▶ AUTOSAR R4.3 Rev 0
- ▶ AUTOSAR SWS document version: 4.3.0
- ▶ Module version: 1.0.30.B466224
- ▶ Supplier: Elektrobit Automotive GmbH

### 3.3.6.1. Change log

This chapter lists the changes between different versions.

#### Module version 1.0.30

2021-09-17

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

#### Module version 1.0.29

2021-07-28



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

#### **Module version 1.0.28**

2021-06-25

- ▶ ASCSOMEIPTP-108 Fixed known issue: SomelpTp notifies the application of a reception failure while reception is not yet started

#### **Module version 1.0.27**

2021-03-05

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

#### **Module version 1.0.26**

2021-01-22

- ▶ Support for Det Runtime Errors reporting using Det\_ReportRuntimeError() API

#### **Module version 1.0.25**

2020-10-23

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

#### **Module version 1.0.24**

2020-07-31

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

#### **Module version 1.0.23**

2020-06-19

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

#### **Module version 1.0.22**

2020-02-21

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



### Module version 1.0.21

2019-11-08

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

### Module version 1.0.20

2019-10-11

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

### Module version 1.0.19

2019-09-06

- ▶ ASCSOMEIPTP-86 Fixed known issue: SomelpTp generator creates redundant internal buffer names for meta-data
- ▶ ASCSOMEIPTP-87 Fixed known issue: During an ongoing assembly process, the reception of a segment with TP-Flag = 0 is incorrectly considered as a new single PDU

### Module version 1.0.18

2019-08-09

- ▶ Allow SomelpTpNPduSeparationTime to have value Zero
- ▶ Det reporting SOMEIPTP\_E\_INCONSISTENT\_SEQUENCE in case assembly interrupted by single segment

### Module version 1.0.17

2019-07-12

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

### Module version 1.0.16

2019-06-14

- ▶ Internal module improvement. This module version update does not affect module functionality
- ▶ ASCSOMEIPTP-77 Fixed known issue: SomelpTp does not throw an error if SomeIpTpRxSduRef or SomeIpTpRxNPduRef are not unique



### Module version 1.0.15

2019-05-17

- ▶ Handling negative return of PduR\_SomeIpTpTransmit() so that the ongoing disassimble is cancelled and TxConfirmation is called with NTFRSLT\_E\_NOT\_OK
- ▶ Allowed transmission request of the same PDU in the context of PduR\_SomeIpTpTxConfirmation()

### Module version 1.0.14

2019-04-18

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

### Module version 1.0.13

2019-03-22

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

### Module version 1.0.12

2019-02-15

- ▶ ASCSOMEIPTP-70 Fixed known issue: The call to TxConfirmation() is missing when SomeIpTp\_Transmit() is called for the same PDU ID
- ▶ ASCSOMEIPTP-72 Fixed known issue: SomeIpTp\_Transmit() copies data from SduDataPtr even though it may be NULL\_PTR

### Module version 1.0.11

2018-10-26

- ▶ ASCSOMEIPTP-56 Added new feature: Retry mechanism in CopyTxData specially for SoAd Fanout configuration.
- ▶ Allow the call of SomeIpTp\_TriggerTransmit from the context of PduR\_SomeIpTpTransmit.

### Module version 1.0.10

2018-09-28

- ▶ ASCSOMEIPTP-50 Fixed known issue: Incorrect copying for the received first segment data to the upper layer.



### Module version 1.0.9

2018-08-24

- ▶ ASCSOMEIPTP-46 Fixed known issue: Invalid check for the remaining buffer length returned from previous PduR\_SomeIpTpCopyTxData().
- ▶ Support for TxConfirmation Timeout support.
- ▶ SomeIpTp shall provide PduR Tp and IF APIs compliant to ASR 403.
- ▶ ASCSOMEIPTP-47 Fixed known issue: Invalid check for the SduLength passed through SomeIpTp\_TriggerTransmit() for last segment transmission.

### Module version 1.0.8

2018-07-27

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

### Module version 1.0.7

2018-06-22

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

### Module version 1.0.6

2018-05-25

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

### Module version 1.0.5

2018-04-20

- ▶ Internal module improvement. This module version update does not affect module functionality.
- ▶ Add support for PduLengthType uint32.

### Module version 1.0.4

2018-03-16

- ▶ Limit the configured (PduLength-12) for TxNPdus to be dividable by 16.

### Module version 1.0.3

2018-02-16



- ▶ Create Design.
- ▶ Integrate Com Transformer Support.

#### **Module version 1.0.2**

2018-01-19

- ▶ Reworked the code to limit the usage of critical sections on shared variables that need protection.
- ▶ ASCSOMEIPTP-20 Fixed known issue: RxTimeout is not restarted after the reception of a new consecutive segment.
- ▶ Internal module improvement. This module version update does not affect module functionality.

#### **Module version 1.0.1**

2017-11-17

- ▶ Usage of Det\_ReportError instead of Det\_ReportRuntimeError.
- ▶ Internal module improvement. This module version update does not affect module functionality.

#### **Module version 1.0.0**

2017-09-22

- ▶ Initial AUTOSAR 4.3 version

#### **3.3.6.2. New features**

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

#### **3.3.6.3. EB-specific enhancements**

This chapter lists the enhancements provided by the module.

- ▶ SomeIpTp module supports TxConfirmation Timeout.

Description:

SomeIpTp shall monitor a successful transmission of an Npdu by having a timeout for receiving a TxConfirmation that indicates a successful transmission. It is added to have compatibility with PduR 4.0.3 IfTxConfirmation API (i.e. `SomeIpTp_TxConfirmation()`) which is called only if successful transmission occurs otherwise if transmission failed no `SomeIpTp_TxConfirmation()` will be called. So a timeout is added to handle a failed transmission of the Npdu.



- ▶ SomeIpTp module supports the immediate call of `SomeIpTp_TriggerTransmit()`.

Description:

SomeIpTp shall allow calling `SomeIpTp_TriggerTransmit()` in the following contexts.

- ▶ `PduR_SomeIpTpTransmit()` calls `SoAd_IfTransmit()`, which will immediately call `PduR_SoAdIfTriggerTransmit()` in this context, leading to a call to `SomeIpTp_TriggerTransmit()`
- ▶ `PduR_SomeIpTpTransmit()` calls `SoAd_IfTransmit()`, and by next `SoAd_MainFunction()`, it will call `PduR_SoAdIfTriggerTransmit()` in the context of next `SoAd_MainFunction()`, leading to a call to `SomeIpTp_TriggerTransmit()`

### 3.3.6.4. Deviations

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

- ▶ Only pre-compile configuration is supported

Description:

The SomeIpTp module only supports the VARIANT-PRE-COMPILe configuration variant. VARIANT-POST-BUILD and VARIANT-LINK-TIME are not supported.

Requirements:

ECUC\_SomeIpTp\_00001

- ▶ Multiplicity for `SomeIpTpTxNSduHandleId`

Description:

`SomeIpTpTxNSduHandleId` has a multiplicity of 1.

Requirements:

ECUC\_SomeIpTp\_00020

- ▶ Multiplicity for `SomeIpTpTxNPduHandleId`

Description:

`SomeIpTpTxNPduHandleId` has a multiplicity of 1.

Requirements:

ECUC\_SomeIpTp\_00017

- ▶ Storing More Segment Flag for every PDU ID



#### Description:

The `SomeIpTp` module does not store the status of the `More Segment Flag` for every PDU ID separately, which is passed by a call of `SomeIpTP_RxIndication()`.

#### Requirements:

SWS\_SomeIpTp\_00029

- ▶ Payload length in Tx path depends on `availableDataPtr`

#### Description:

To calculate the possible maximum size for all consecutive `SomeIpTp` segments, the `SomeIpTp` module does not consider the available buffer size of the upper layer by evaluating the `availableDataPtr`. This complies with PRS\_SOMEIP\_00734. Bugzilla RFC(78197).

#### Requirements:

SWS\_SomeIpTp\_00019

- ▶ SWS\_SomeIpTp\_00045 at Rx to include case of SWS\_SomeIpTp\_00009 at Tx

#### Description:

SWS\_SomeIpTp\_00009 mandates the following: *If the provided SDU fits into one single PDU, the provided SOME/IP header shall be used with no modification.* This requires that sufficiently small SOME/IP messages are not modified by the `SomeIpTp` module in the transmission path. However, SWS\_SomeIpTp\_00045 contradicts such behavior since it mandates that `SomeIpTp_RxIndication()` shall check and enforce that the TP-Flag is set to 1. The `SomeIpTp` module is implemented in a way that allows SWS\_SomeIpTp\_00045 at Rx to include the case of SWS\_SomeIpTp\_00009 from Tx. Bugzilla RFC(78084).

#### Requirements:

SWS\_SomeIpTp\_00045

- ▶ Payload length for all segments but the last shall be identical and as close to maximum as possible

#### Description:

SWS\_SomeIpTp\_00019 and SWS\_SomeIpTp\_00024 suggest that the `SomeIpTp` module shall use the value that is returned from `PduR_SomeIpTpCopyTxData()` via `availableDataPtr` to dynamically compute the size of the SOME/IP segment, which shall be transmitted next. This violates PRS\_SOMEIP\_00734. The EB `SomeIpTp` module is implemented according to PRS\_SOMEIP\_00734, which states the following: *The sender shall segment in a way that all segments with the More Segment Flag set to 1 are of the same size.* Bugzilla RFC(78197).

#### Requirements:



SWS\_SomeIpTp\_00024

- ▶ PduR\_SomeIpTp API compliance

Description:

The `SomeIpTp` module is compliant with the `PduR` module of AUTOSAR 4.0.3.

Requirements:

SWS\_SomeIpTp\_00024 SWS\_SomeIpTp\_00025 SWS\_SomeIpTp\_00054

- ▶ ComStack APIs for TP are compliant with AUTOSAR 4.0.3

Description:

The `SomeIpTp` provides TP APIs that are compliant with AUTOSAR 4.0.3.

Requirements:

SWS\_SomeIpTp\_00056 SWS\_SomeIpTp\_00021 SWS\_SomeIpTp\_00023 SWS\_SomeIpTp\_91001

- ▶ SomeIpTp check for provided SduLength for the single and last segment

Description:

SomeIpTp checks the given `PduInfoPtr->SduLength` if it is smaller than the actual PDU-length for all segments except for single and last where it checks for the expected last segment length (including SomeIpTp header for the last segment). and if so `SomeIpTp_TriggerTransmit()` shall not copy any data and return `E_NOT_OK`.

Requirements:

SWS\_SomeIpTp\_00055

- ▶ SomeIpTp calls `PduR_SomeIpTpCopyRxData` twice for the first segment

Description:

If a SOME/IP segment is successfully received with Offset Field set to 0 and after the SomeIpTp module has called the API `PduR_SomeIpTpStartOfReception()`, the SomeIpTp module checks the returned bufferSizePtr. If the bufferSizePtr is greater or equal to SOME/IP header, the SomeIpTp module calls the API `PduR_SomeIpTpCopyRxData()` to pass the SOME/IP header information after setting Tp-Flag to 0. This PDU contains the following content: Request ID [32 bit] Protocol Version [8 bit] Interface Version [8 bit] Message Type [8 bit] Return Code [8 bit] then the SomeIpTp module checks the returned bufferSizePtr. If the bufferSizePtr is greater or equal to first segment payload without SOME/IP Tp header, the SomeIpTp module calls the API `PduR_SomeIpTpCopyRxData()` to pass this payload. (RS\_SOMEIP\_00010)

Requirements:



SWS\_SomeIpTp\_00034

- ▶ SomeIpTp applies Tp retry mechanism

Description:

SomeIpTp applies Tp retry mechanism in its calls for PduR\_SomeIpTpCopyTxData.

Requirements:

SWS\_SomeIpTp\_00018

- ▶ SomeIpTp transmit with a PDU ID which is currently used

Description:

The AUTOSAR version 4.3 behaviour of this requirement is replaced by 4.4 AUTOSAR requirement.

Requirements:

SWS\_SomeIpTp\_00022

- ▶ SomeIpTp transmit shall not copy header from SduDataPtr

Description:

The AUTOSAR version 4.3 behaviour of this requirement is replaced by 4.4 AUTOSAR requirement.

Requirements:

SWS\_SomeIpTp\_00008

- ▶ SomeIpTpNPduSeparationTime can be zero

Description:

Name SomeIpTpNPduSeparationTime Description Sets the duration of the minimum time in seconds the SomeIpTp module shall wait between the transmissions of N-PDUs. Multiplicity 1 Type EcucFloat-ParamDef Range [0 .. INF[ Default value -- Post-Build Variant Value true Value Configuration Class Pre-compile time X VARIANT-PRE-COMPILe Link time X VARIANT-LINK-TIME Post-build time X VARIANT-POST-BUILD Scope / Dependency scope: local

Requirements:

ECUC\_SomeIpTp\_00006



### 3.3.6.5. 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.

### 3.3.6.6. Open-source software

SomelTp does not use open-source software.

## 3.3.7. Tcplp module release notes

- ▶ AUTOSAR R4.3 Rev 0
- ▶ AUTOSAR SWS document version: 4.3.0
- ▶ Module version: 3.5.14.B466224
- ▶ Supplier: Elektrobit Automotive GmbH

### 3.3.7.1. Change log

This chapter lists the changes between different versions.

#### Module version 3.5.14

2021-10-08

- ▶ Added support to handle different return values from Tcplp\_CopyTxDataAPI
- ▶ Added Measurement data support update

#### Module version 3.5.13

2021-06-25

- ▶ Added the support of Duplicate Address Detection for DHCPv4
- ▶ Added support to enable UDP sockets to always listen to limited broadcasts



- ▶ Added support for IPv4 Router extension
- ▶ ASCTCPIP-2777 Fixed known issue: Autolp NvM - First generation yields the same address

#### **Module version 3.5.12**

2021-03-05

- ▶ Added support to store IP address in NVM RAM
- ▶ Added the support of Duplicate Address Detection for DHCPv6
- ▶ ASCTCPIP-2632 Fixed known issue: Tcplp uses the wrong data type for the Eth buffer index

#### **Module version 3.5.11**

2020-10-23

- ▶ ASCTCPIP-2436 Fixed known issue: Out-of-bounds read access by packet reception in dual-stack and dual-protocol configuration
- ▶ ASCTCPIP-2438 Fixed known issue: Out-of-bounds read access by reception of corrupted DHCPv4 message
- ▶ ASCTCPIP-2452 Fixed known issue: Tcplp inverts byte order on big endian platforms and reports false generator error
- ▶ ASCTCPIP-2352 Fixed known issue: Tcplp does not release an allocated TCP socket
- ▶ ASCTCPIP-2489 Fixed known issue: Tcplp does not create a new initial sequence number for each passive connection
- ▶ ASCTCPIP-2500 Fixed known issue: Tcplp\_TcpConnect returns E\_NOT\_OK instead of E\_OK if SYN transmission fails

#### **Module version 3.5.10**

2020-08-07

- ▶ Added support to make TCP upper layer copying algorithm configurable
- ▶ Added the support of unpredictable sequence numbers according to IETF RFC 6528
- ▶ Added the support for ignoring TCP RST frames when in state TIME\_WAIT according to IETF RFC 1337
- ▶ ASCTCPIP-2375 Fixed known issue: Tcplp sends packages which are addressed to a link local address to the router
- ▶ Added the TCP timeout for closing sockets in state SYN\_RECEIVED



- ▶ ASCTCPIP-2399 Fixed known issue: Tcplp does not release DHCPv4 address if address conflict is detected

### Module version 3.5.9

2020-06-19

- ▶ ASCTCPIP-2081 Fixed known issue: Tcplp does not send configurable options in DHCP Request
- ▶ Added support for filtering received packets by source MAC address
- ▶ Added counters for IP frames dropped or passed due to firewall rule
- ▶ Added support for filtering received packets by traffic class and flow label
- ▶ Added security Architecture for the Internet Protocol according to IETF RFC 4301
- ▶ Added IP Authentication Header according to IETF RFC 4302
- ▶ Added the AES-CMAC-96 Algorithm for Authentication Header according to IETF RFC 4494
- ▶ Added the Galois Message Authentication Code (GMAC) Algorithm (AUTH\_AES\_128\_GMAC, AUTH\_AES\_256\_GMAC) for Authentication Header according to IETF RFC 4543
- ▶ Added the HMAC Algorithm (HMAC-SHA-256-128) for Authentication Header according to IETF RFC 4868

**Note:** If HMAC is used to secure IPv4 traffic with a Linux host, the respective transform state i.e. security association needs to be configured with the `align4` flag. For more information please refer to `man ip-xfrm`.

- ▶ Added new API, `Tcplp_IsConnectionReady`, to check if physical address is known and IpSec SA is established
- ▶ Added the support of SYN cookies according to IETF RFC 4987
- ▶ ASCTCPIP-2249 Fixed known issue: ARP creates entries for multicast remote Ip address
- ▶ Changed value range of parameter `TcplpArpTableEntryTimeout` (1 second to 65535 seconds or Infinity)
- ▶ Fixed processing of TCP SYN segments which contain a RST

### Module version 3.5.8

2020-02-21

- ▶ Improved Window Update transmission in ACK
- ▶ Improved precision of ARP timeout counter
- ▶ Rework DHCPv4 to use `UdpTransmit` API
- ▶ Rework DHCPv6 to use `UdpTransmit` API
- ▶ Updated `TcpIp_RxIndication` to always drop the packet if the total length is greater than MTU, regardless if DET reporting is enabled or not.



- ▶ Improved message transmission (TCP, UDP and ICMP use the same transmit API)
- ▶ Fixed unreachable code assert failing on reception of TCP FIN segment with retransmitted data.
- ▶ Fixed unreachable code assert failing on TCP reception when listen socket closed or upper layer not accepting.
- ▶ ASCTCPIP-1979 Fixed known issue: Possible buffer overflow on TCP segment reception if out-of-order buffering is activated
- ▶ Improved checking Server Identifier option in every DhcpV6 message
- ▶ Improved handling of the valid and preferred lifetime for DhcpV6 messages
- ▶ Improved handling of restarting the 2 MSL timeout when retransmitted FIN is received in the state Time-Wait
- ▶ ASCTCPIP-2012 Fixed known issue: Compilation fails when SoAd is not configured
- ▶ Improved handling of discarding DHCP messages in state init

#### **Module version 3.5.7**

2019-10-11

- ▶ ASCTCPIP-1823 Fixed known issue: Invalid controller transfer to OFFLINE state due to address with multiple assignment methods

#### **Module version 3.5.6**

2019-07-05

- ▶ ASCTCPIP-1792 Fixed known issue: Out-of-bounds read access caused by an invalid DHCPv4 router option
- ▶ ASCTCPIP-1794 Fixed known issue: Denial of service by reception of a Neighbor Solicitation/Advertisement with invalid option length
- ▶ ASCTCPIP-1795 Fixed known issue: Integer underflow causes wrong length information for the upper layer

#### **Module version 3.5.5**

2019-06-14

- ▶ ASCTCPIP-1646 Fixed known issue: Tcplp sends ARP reply with incorrect target hardware address
- ▶ ASCTCPIP-1694 Fixed known issue: NDP cache entry is not unlocked if neighbor solicitation transmission fails



- ▶ ASCTCPIP-1654 Fixed known issue: Tcplp does not drop Neighbor Advertisements/Solicitation with broadcast/multicast target/source link layer address
- ▶ Added support for IPv6 Global address Duplicate Address Detection
- ▶ ASCTCPIP-1735 Fixed known issue: Denial of service by reception of a corrupted DHCPv6 response on 16-bit platforms
- ▶ Added support for extracting and transmitting arbitrary configured DHCP options
- ▶ Added support for IPv6 Router extension
- ▶ Added basic support for out-of-order reception and buffering of TCP segments.

**Note:** With introduction of this feature, if out-of-order reception is disabled a duplicate ACK is not sent if an out-of-order segment is received. A duplicate ACK would indicate that out-of-order buffering is supported.

- ▶ ASCTCPIP-1759 Fixed known issue: Partial IPv4 and IPv6 checksum calculation might fail

#### Module version 3.5.4

2019-02-15

- ▶ Added support for Measurement data
- ▶ Added support for IPv6 Next hop determination
- ▶ ASCTCPIP-1619 Fixed known issue: Tcplp not be able to assign a DhcpV6 address
- ▶ ASCTCPIP-1606 Fixed known issue: The DhcpV4 client transmits a message over a closed DhcpV4 Udp socket
- ▶ ASCTCPIP-1617 Fixed known issue: Tcplp does not compile if the configurable DHCP option feature is enabled and Det is disabled
- ▶ Added Duplicate Address Detection conflict callout support for IPv4
- ▶ Added support for Measurement data for discarded/replaced ARP entries
- ▶ Added Api for accessing DHCP status
- ▶ ASCTCPIP-1630 Fixed known issue: Out-of-bounds read access caused by malformed NDP packet reception
- ▶ ASCTCPIP-1638 Fixed known issue: Tcplp transmits a malformed IPv4 DHCP FQDN option
- ▶ ASCTCPIP-1631 Fixed known issue: Out-of-bounds read access and potential out-of-bounds write access caused by IPv4 ICMP echo request packet reception
- ▶ ASCTCPIP-1629 Fixed known issue: Out-of-bounds read access caused by corrupted IPv6 packet reception
- ▶ Added support for IPv6 Static address and IPv6 Link Local address Duplicate Address Detection
- ▶ Added support for IP Stack Hardening: ACK Loop DoS Attack



### Module version 3.5.3

2018-11-23

- ▶ ASCTCPIP-1574 Fixed known issue: Potential corruption of internal data structure during reassembly of malicious fragments

### Module version 3.5.2

2018-10-26

- ▶ ASCTCPIP-1528 Fixed known issue: Malformed ICMPv6 Echo Reply is sent as a response to a 4-byte ICMPv6 Echo Request
- ▶ ASCTCPIP-1552 Fixed known issue: Incorrect inclusion of ComStack\_Types.h

### Module version 3.5.1

2018-09-20

- ▶ ASCTCPIP-1520 Fixed known issue: TCP option filter does not compile if TCP keep alive is turned off
- ▶ Added support for Post Build Selectable

### Module version 3.5.0

2018-09-06

- ▶ ASCTCPIP-1495 Fixed known issue: IPv4 Unicast address of controller is incorrectly generated
- ▶ ASCTCPIP-1466 Fixed known issue: Tcplp calls the external function <Up>\_CopyTxData() in a critical section
- ▶ Added support for more flexible memory allocation per socket

Note: With this feature the configuration of the memory for TCP sockets has changed. The conversion to the new memory configuration can be simply applied by adding one entry in TcplpConfig/TcplpMemoryConfig/TcplpMemoryPool. TcplpMemoryBlockSize needs to be set to TcplpBufferMemory divided by TcplpNumMemoryBlocks and TcplpMemoryBlockCount to TcplpNumMemoryBlocks.

### Module version 3.4.0

2018-06-22

- ▶ Added support for Configurable DSCP and Flow Label



- ▶ Updated requirements and configuration to AUTOSAR SWS 4.3.0
- ▶ ASCTCPIP-1376 Fixed known issue: Memory section conflicts between definitions and declarations
- ▶ Added support for handling arbitrary configured DHCP options
- ▶ Added support for configurable UDP checksum calculation
- ▶ ASCTCPIP-1452 Fixed known issue: TCP stops transmitting data after retransmission
- ▶ Added Tcplp\_MainFunctionTx() to allow immediate transmission of TCP segments
- ▶ ASCTCPIP-1443 Fixed known issue: Enabling DET in the dual stack version leads to incorrect function calls via the function pointers
- ▶ ASCTCPIP-1454 Fixed known issue: Tcplp might access data from wrong Tcplp controller
- ▶ ASCTCPIP-1457 Fixed known issue: Incorrect reassembly of fragmented IP message in case header size of IP fragments is not constant

### Module version 3.3.0

2018-02-16

- ▶ Added Support of router and prefix discovery
- ▶ ASCTCPIP-1327 Fixed known issue: Tcplp might get stuck in an endless loop if more than 255 UDP/TCP sockets are configured
- ▶ Improved IP header writing to use 16 + 32 bit writes (configurable by integrator)
- ▶ Improved Checksum computation to use aligned 64 bit reads
- ▶ Added Support for TCP option filter
- ▶ Added support for IPv6 Extension Header Filter
- ▶ Added support for Defensive Neighbor Solicitation/Advertisement Processing

### Module version 3.2.11

2017-12-15

- ▶ Added support for TCPIP\_IPADDR\_ASSIGNMENT\_ALL for Tcplp\_RequestIpAddrAssignment
- ▶ ASCTCPIP-1290 Fixed known issue: Tcplp\_GetIpAddr() does not return the correct DHCP address
- ▶ Added Support of local address "ANY" for Tcplp\_Request/ReleaseIpAddrAssignment

### Module version 3.2.10

2017-10-19



- ▶ Added mechanism to prevent ARP floods (configurable)

#### **Module version 3.2.9**

2017-09-22

- ▶ ASCTCPIP-1158 Fixed known issue: `Tcplp_GetIpAddr` does not return correct IP address
- ▶ Updated to MISRA 2012
- ▶ Added mechanism to prevent ARP floods (non-configurable)
- ▶ Added support for checksum offloading according to AUTOSAR 4.2.1

#### **Module version 3.2.8**

2017-08-25

- ▶ ASCTCPIP-1136 Fixed known issue: `Tcplp` unexpectedly removes ARP entries

#### **Module version 3.2.7**

2017-07-28

- ▶ ASCTCPIP-1103 Fixed known issue: `Tcplp` receives frames on wrong controller
- ▶ ASCTCPIP-1120 Fixed known issue: Incorrect checksum calculation of fragmented IPv6 UDP message
- ▶ ASCTCPIP-1127 Fixed known issue: Incorrect length information for Out-Of-Order or disabled fragmentation

#### **Module version 3.2.6**

2017-06-30

- ▶ ASCTCPIP-1077 Fixed known issue: `Tcplp_TcpTransmit` unexpectedly reports `TCPIP_E_NOBUFS`
- ▶ ASCTCPIP-1086 Fixed known issue: Incorrect checksum calculation of fragmented UDP message

#### **Module version 3.2.5**

2017-06-02

- ▶ Added support to read and write the NDP cache table



#### Module version 3.2.4

2017-05-05

- ▶ ASCTCPIP-1025 Fixed known issue: ICMP Echo Replies might exceed MTU
- ▶ Added support to read and write the ARP cache table
- ▶ ASCTCPIP-1029 Fixed known issue: Unreachable code assertion in LocalAddrSM causing execution to stop

#### Module version 3.2.3

2017-03-31

- ▶ Added support for Simple DHCPv4
- ▶ ASCTCPIP-932 Fixed known issue: Incorrect DHCPv6 timeout calculation if more than one DHCPv6 assignment is configured
- ▶ Added support of IPv6 source address selection algorithm according to IETF RFC 6724
- ▶ ASCTCPIP-989 Fixed known issue: Incorrect IPv4 Link Local timeout calculation
- ▶ ASCTCPIP-985 Fixed known issue: Link Local IPv6 address cannot be released
- ▶ ASCTCPIP-1001 Fixed known issue: Incorrect DHCPv6 retransmission timeout calculation

#### Module version 3.2.2

2017-03-03

- ▶ ASCTCPIP-931 Fixed known issue: DHCPv6 assignment fails if more than one controller is configured
- ▶ Added support for indicating change in the physical address table (Up\_PhysAddrTableChg) for IPv6
- ▶ Improved initialization of module (name of configuration can be used as symbol for Tcplp\_Init())

#### Module version 3.2.1

2017-02-03

- ▶ Support of Fully Qualified Domain Name (FQDN) Option for Dynamic Host Configuration Protocol for IPv6 Clients
- ▶ Added support for IPv4 Address Conflict Detection and Defense according to IETF RFC 5227
- ▶ ASCTCPIP-873 Fixed known issue: Tcplp might send datagrams to wrong destination MAC address
- ▶ ASCTCPIP-890 Fixed known issue: The FQDN option is incorrectly terminated
- ▶ ASCTCPIP-834 Fixed known issue: Dhcp does not release the IPv6 address after the lease time expires



- ▶ ASCTCPIP-905 Fixed known issue: The variable Tcplp\_TCP\_fragmentIDCounter is created without using the MemMap concept

### Module version 3.2.0

2016-11-04

- ▶ ASCTCPIP-564 Fixed known issue: TCPIP\_E\_NOTCONN is unexpectedly reported to Det because the socket is already closed
- ▶ ASCTCPIP-590 Fixed known issue: Tcplp might use a wrong netmask
- ▶ ASCTCPIP-664 Fixed known issue: ARP Callout is not called for received multicast datagrams
- ▶ ASCTCPIP-679 Fixed known issue: TCP does not transmit RST in LISTEN state
- ▶ ASCTCPIP-677 Fixed known issue: IP address is released if DHCP server does not acknowledge the request
- ▶ Updated Up\_IcmpMsgHandler to AUTOSAR SWS 4.3
- ▶ Updated Tcplp Configuration to AUTOSAR SWS 4.2.2
- ▶ Fixed Compiler abstractions and memory sections
- ▶ ASCTCPIP-398 Fixed known issue: Source address in DHCP messages is not set to the unspecified address
- ▶ Added support of Internet Protocol version 6 (IPv6) including Hop-By-Hop Option Header and Destination Option Header
- ▶ Added support for fragmentation of over sized IPv6 and IPv4 frames
- ▶ Added support for reception and reassembly of fragmented IPv6 and IPv4 frames
- ▶ Added support of Internet Control Message Protocol Version 6 (ICMPv6) including Destination Unreachable, Time Exceeded, Parameter Problem, Echo Request/Reply Messages
- ▶ Added support of Address Resolution and Neighbor Unreachability Detection (NDP)
- ▶ Added client support of Dynamic Host Configuration Protocol for IPv6
- ▶ Added support for Stateless Address Autoconfiguration of IPv6 Link-Local Addresses
- ▶ Updated Tcplp\_IpAddrAssignmentType and Tcplp\_ReturnType to AUTOSAR SWS 4.2.2
- ▶ Updated Tcplp to use Eth\_BufIdxType

### Module version 3.1.6

2016-05-25

- ▶ ASCTCPIP-565 Fixed known issue: Tcplp calls the wrong upper layer functions



### Module version 3.1.5

2016-04-29

- ▶ Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeader-File`
- ▶ ASCTCPIP-498 Fixed known issue: DHCP assignment fails if more than one DHCP assignment is configured
- ▶ ASCTCPIP-499 Fixed known issue: `TCPIP_E_INV_ARG` is unexpectedly reported to Det because the socket is already closed
- ▶ Added support for a user configurable packet filter, to enable ARP table updates from IPv4 datagrams based on payload content.
- ▶ ASCTCPIP-514 Fixed known issue: Retransmission timeout of a DHCP REQUEST might be less than 60 seconds
- ▶ ASCTCPIP-539 Fixed known issue: Link Local IP address is never assigned if `TcpIpAutoIpInitTimeout` is smaller than one
- ▶ ASCTCPIP-540 Fixed known issue: Address assignment change callouts for address ANY are not called
- ▶ ASCTCPIP-558 Fixed known issue: AutoIp/Dolp assignment fails if more than one AutoIp/Dolp assignment is configured

### Module version 3.1.4

2016-02-05

- ▶ ASCTCPIP-470 Fixed known issue: DHCP IP address assignment doesn't work if `TcpIpAssignmentTrigger` = `TCPIP_AUTO`
- ▶ ASCTCPIP-475 Fixed known issue: TCP does not free allocated memory if socket is unexpectedly closed
- ▶ ASCTCPIP-476 Fixed known issue: Reset is not transmitted if socket in state `CLOSE_WAIT` is closed with `TcpIp_Close` and `force = TRUE`

### Module version 3.1.3

2016-01-14

- ▶ ASCTCPIP-439 Fixed known issue: `TcpIp_TcpTransmit` might transmit wrong data
- ▶ ASCTCPIP-433 Fixed known issue: TCP might pass invalid data to upper layer
- ▶ ASCTCPIP-435 Fixed known issue: TCP might use an invalid MSS for segmentation
- ▶ ASCTCPIP-434 Fixed known issue: TCP might call `<UL>_TxConfirmation` with incorrect Length
- ▶ ASCTCPIP-432 Fixed known issue: `<UL>_TxConfirmation` might be called after `<UL>_TcpEvent(TCPIP_TCP_CLOSED)`



- ▶ ASCTCPIP-430 Fixed known issue: If an out of order FIN is received TCP calls <UL>TcplpEvent(TCPIP\_-TCP\_FIN\_RECEIVED)
- ▶ ASCTCPIP-441 Fixed known issue: TCP might not accept a valid segment from remote host
- ▶ ASCTCPIP-431 Fixed known issue: If a valid FIN/ACK with data is received in state SYN-RECEIVED data is not passed to upper layer
- ▶ ASCTCPIP-436 Fixed known issue: If an upper layer calls Tcplp\_Close a FIN might not be transmitted
- ▶ ASCTCPIP-440 Fixed known issue: TCP might close the connection although not all data has been transmitted
- ▶ ASCTCPIP-421 Fixed known issue: Initialization of Tcplp causes to call EthIf\_GetPhysAddr() before EthIf\_-ControllerInit()
- ▶ Added config check that EthIfctrlIdx does not exceed TcplpEthIfCtrlIndexMax
- ▶ ASCTCPIP-444 Fixed known issue: Tcplp might send more data segments than a single full sized segment if nagle is used
- ▶ ASCTCPIP-449 Fixed known issue: Tcplp does not immediately shutdown if state OFFLINE is requested in state ONHOLD
- ▶ Add checks to Tcplp\_Bind(), Tcplp\_Listen() and Tcplp\_Connect() to operate only if local address to use is assigned
- ▶ ASCTCPIP-457 Fixed known issue: Pending data might not be transmitted during transition from ONHOLD to ONLINE

### Module version 3.1.2

2015-11-06

- ▶ ASCTCPIP-407 Fixed known issue: Tcplp might access an invalid memory address
- ▶ ASCTCPIP-409 Fixed known issue: Tcplp accepts SYN,ACK with incorrect ACK number in state SYN-SENT
- ▶ ASCTCPIP-415 Fixed known issue: Tcplp might call <UL>\_TcplpEvent() with an invalid SocketId for Tcp listen sockets
- ▶ ASCTCPIP-418 Fixed known issue: Tcplp uses wrong netmask on big endian CPU

### Module version 3.1.1

2015-10-09

- ▶ ASCTCPIP-366 Fixed known issue: Checksum calculation might fail
- ▶ Change configuration parameter TcplpDefaultRouter to optional



- ▶ ASCTCPIP-383 Fixed known issue: Tcplp incorrectly acknowledges a FIN,ACK of the remote host
- ▶ Tcplp does not send unexpected acknowledgments in the next Mainfunction after Tcplp\_Close is called anymore
- ▶ ASCTCPIP-386 Fixed known issue: Tcplp controller statemachine might not switch to OFFLINE if DHCP is used
- ▶ Support of keep alive probes according to AUTOSAR 4.2.2
- ▶ ASCTCPIP-399 Fixed known issue: TCP retransmits a correctly acknowledged data segment

#### **Module version 3.1.0**

2015-06-19

- ▶ Support of Transmission Control Protocol (TCP) including Nagle Algorithm
- ▶ Support of Dynamic Host Configuration Protocol (DHCPv4)
- ▶ Support of Fully Qualified Domain Name (FQDN) Option for Dynamic Host Configuration Protocol for IPv4 Clients
- ▶ Support of Dynamic Configuration of IPv4 Link-Local Addresses (Auto-IP)
- ▶ Support of ISO 13400-2 recommended timing values for Dynamic Configuration of IPv4 Link-Local Addresses
- ▶ Support of Gratuitous ARP.
- ▶ Support of configurable upper layer

#### **Module version 3.0.1**

2014-12-12

- ▶ ASCTCPIP-313 Fixed known issue: Reception and transmission of certain multicast IPv4 datagrams fails

#### **Module version 3.0.0**

2014-11-30

- ▶ Initial mass production version (limited feature set).

#### **3.3.7.2. New features**

- ▶ Added support to handle different return values from Tcplp\_CopyTxDataAPI



- ▶ Added Measurement data support update

### 3.3.7.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- ▶ Ipv4 receive indication callout for ARP insertion

Description:

With configuration parameters contained in `TcpIpIpv4ArpPacketFilter()` the user is able to specify a callout which is invoked on successful reception of an IPv4 datagram. The callout decides based on the received IPv4 datagram's content, whether the sender's source address shall be inserted into the ARP table or not.

- ▶ IP fragmentation and reassembly support can be enabled and disabled separately.
- ▶ IP fragmentation supports two modes: in-order and out-of-order transmission.
- ▶ Handling of atomic packets according to IETF RFC 6946.

Description:

IETF RFC 6946 states that "atomic" fragments, i.e. packets which have a fragment header, but an 'offset' of zero and the 'more' flag also equal to zero - i.e. fragments that do contain the whole packet, shall NOT collide with partially-assembled packets using the same ID, while the partially-assembled fragments shall NOT be discarded due to this non-collision.

- ▶ Simple DHCPv4 Client

With the configuration parameter `TcpIpUseSimpleDhcpClient` the simple DHCPv4 client can be enabled. The IP address is assigned through an exchange of 2 messages with the DHCP server. Client sends a `DHCPDISCOVER` with XID set to the lower 4 bytes of the MAC address. If server responses with a `DHCPOFFER` with the XID set to the client's MAC address, the client sets its own IP address to that given in the `YIADDR` field.

- ▶ Set static ARP/NDP cache entries through API `Tcplp_SetRemotePhysAddr`

`Tcplp_SetRemotePhysAddr` allows to set an entry in the NDP or ARP cache to static, remove a static entry if no longer needed or clear the whole cache.

- ▶ Request the assignment of multiple assignment methods through `Tcplp_RequestIpAddrAssignment`

If `Tcplp_RequestIpAddrAssignment` is called with `LocalAddrId` configured as ANY, all assignment methods of all local addresses configured for the referenced controller are assigned.

If `Tcplp_RequestIpAddrAssignment` is called with `Type` equals `TCPIP_IPADDR_ASSIGNMENT_ALL`, all assignment methods of the specified `LocalAddrId` are assigned. See [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=74847](http://www.autosar.org/bugzilla/show_bug.cgi?id=74847)



▶ Mechanism to prevent ARP floods

After the transmission of an ARP request the Tcplp skips the transmission of any further ARP requests to the same destination within a duration of TcplpArpRequestTimeout seconds, according to the mechanism to prevent ARP flooding described in IETF RFC 1122, section 2.3.2.1 ARP Cache Validation. See [http://www.autosar.org/bugzilla/show\\_bug.cgi?id=80210](http://www.autosar.org/bugzilla/show_bug.cgi?id=80210)

▶ Trigger transmissions through Tcplp\_MainFunctionTx()

Tcplp\_MainFunctionTx() allows it to trigger an immediate transmission of TCP segments after the call of Tcplp\_TcpTransmit(). The API can be enabled through the configuration parameter TcplpEnableMainFunctionTx.

▶ Internet Protocol Security (IPsec)

▶ Security Architecture for the Internet Protocol according to IETF RFC 4301

The Tcplp implements the Security Architecture for the Internet Protocol defined in IETF RFC 4301. This includes the configuration of a Security Policy Database (SPD) in which traffic can be configured as BYPASSED, SECURED or DISCARD. For secured traffic the Tcplp allows to configure manual security associations in a Security Association Database (SAD). Every time a UDP/TCP/ICMP frame is received or transmitted, the Tcplp consults the SPD and decides if the frame shall be bypassed, secured or discarded. Deviations to the RFC are listed in [Section 3.3.7.4, “Deviations”](#).

▶ IP Authentication Header according to IETF RFC 4302

The Tcplp allows to secure traffic by the Authentication Header in transport mode defined in IETF RFC 4302. Deviations to the RFC are listed in [Section 3.3.7.4, “Deviations”](#).

▶ The AES-CMAC-96 Algorithm according to IETF RFC 4494

The Tcplp supports the AES-CMAC-96 integrity algorithm for the Authentication Header. Deviations to the RFC are listed in [Section 3.3.7.4, “Deviations”](#).

▶ The Galois Message Authentication Code (GMAC) Algorithm according to IETF RFC 4543

The Tcplp supports the following GMAC integrity algorithm for the Authentication Header:

- ▶ AUTH\_AES\_128\_GMAC
- ▶ AUTH\_AES\_256\_GMAC

Deviations to the RFC are listed in [Section 3.3.7.4, “Deviations”](#).

▶ The HMAC Algorithm according to IETF RFC 4868

The Tcplp supports the following HMAC integrity algorithm for the Authentication Header:

- ▶ HMAC-SHA-256-128

Deviations to the RFC are listed in [Section 3.3.7.4, “Deviations”](#).



**Note:** If HMAC is used to secure IPv4 traffic with a Linux host, the respective transform state i.e. security association needs to be configured with the `align4` flag. For more information please refer to `man ip-xfrm`.

- ▶ Transmission Control Protocol (TCP)
  - ▶ Transmission of the zero-window probes according to IETF RFC 1122

The TCP shall send the first zero-window probe immediately when `Tcplp_TcpTransmit` is called.

### 3.3.7.4. Deviations

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

- ▶ [IPV4] Protocol ARP not optional

Description:

Deactivation of core protocol ARP is not possible. The configuration parameter `TcpIpArpEnabled` is unused.

Rationale:

ARP is a core protocol and mandatory for IPv4. An alternative implementation using preconfigured and static address tables is not available.

Requirements:

`ECUC_Tcplp_00006`

- ▶ Certain TCP features are not supported

Description:

Tcplp does not support the following TCP features:

- Slow Start
- Congestion Avoidance
- Out of order reception
- Fast Retransmit/Recovery

Requirements:

`SWS_TCPIP_00062`, `SWS_TCPIP_00064`, `ECUC_Tcplp_00061`, `ECUC_Tcplp_00063`, `ECUC_Tcplp_-00062`, `ECUC_Tcplp_00060`, `ECUC_Tcplp_00019`, `SWS_TCPIP_00168`



- ▶ Some DET errors not supported

Description:

Module Tcplp does not support the following development errors:

- ▶ TCPIP\_E\_INIT\_FAILED

Requirements:

SWS\_TCPIP\_00042

- ▶ [IPV4] Path MTU discovery not supported

Description:

Discover the maximum transmission unit (MTU) for a path as defined in IETF RFC 1191 (Path MTU Discovery) is not supported. Configuration parameter `TcpIpPathMtuDiscoveryEnabled` is unused.

Requirements:

SWS\_TCPIP\_00055

- ▶ [IPV6] Path MTU discovery not supported

Description:

Discover the maximum transmission unit (MTU) for a path as required in IETF RFC 2460 (Path MTU Discovery) is not supported. Configuration parameter `TcpIpIpV6PathMtuDiscoveryEnabled`, `TcpIpIpV6PathMtuEnabled` and `TcpIpIpV6PathMtuTimeout` are unused.

Requirements:

SWS\_TCPIP\_00160, SWS\_TCPIP\_00158, ECUC\_Tcplp\_00090, ECUC\_Tcplp\_00107, ECUC\_Tcplp\_-00105

- ▶ [IPV4] IPv4 fragmentation/reassembly mirrors IPv6 fragmentation/reassembly behavior

Description:

Tcplp supports IPv4 fragmentation and reassembly in the IPv6 sense only:

- ▶ no overlapping fragments (see rfc5722).
- ▶ honoring of DF=1 to save on IDs and remove the bandwidth limitation incurred from incorrect usage (see rfc6864).
- ▶ non-colliding atomic fragments (see rfc6946, compared to rfc6864 for IPv4).
- ▶ fixed reassembly timeout (no updating from fragments' TTL field) see rfc1122
- ▶ end-to-end fragmentation without re-fragmentation in intermediate routers.



While this is the default for IPv6, we also employ IPv4 in this mode: the DF-flag is on by default, but can be set to off for packets routed out of the in-car-network. This default avoids the bandwidth limitations that stem from large reassembly timeouts and the 16 bit ID field of IPv4. (see rfc6864 for a discussion)

- ▶ no IPv6 path MTU discovery due to the fully-known and configurable environment.

#### Rationale:

IPv4 fragmentation according to rfc791, amended by rfc1122 to clarify timeout issues, then amended further by rfc6864 to overcome bandwidth limitations from the 16 bit fragmentID counter already comes close to the specification of IPv6's fragmentation as specified by rfc2460, amended by rfc5722 and rfc6946. Security concerns described in rfc6274 and further addressed in rfc1858 and rfc3128 suggest disallowing overlapping fragments altogether in a controlled automotive network - as specified in rfc5722 for IPv6. IPv4 did not go there, because of legacy hardware and complex routing in combination with re-fragmentation in the wild of the internet, but requires equality checks for the overlapping parts of overlapping fragments according to rfc6864.

Since in-car networks do not have the alternative routing possibilities (too much randomness), the use case for overlapping fragments is non-existent. Since the attack-surface and overhead nevertheless remain, it is best to disallow overlapping fragments within in-car-networks. What remains is virtually identical to the IPv6 specification. IPv6 provides the newer and more consistent specification without the compatibility support for 40 years of hardware and structures, hence more suitable for automotive applications.

Note: This does not affect the fact that IPv4 uses the tuple (Srclp32,DestIp32,Id16,Protocol8) as unique reassembling ID, whereas IPv6 omits the protocol, i.e.: only uses (Srclp128,DestIp128,Id32). This is maintained throughout this implementation.

#### Requirements:

SWS\_TCPIP\_00054 SWS\_TCPIP\_00102 SWS\_TCPIP\_00231

- ▶ IPv4 fragmentation/reassembly shares configuration with IPv6 fragmentation/reassembly

#### Description:

IPv4 and IPv6 share configuration parameters. The TcpIp module uses a single configuration for IP fragmentation and reassembly. It is named `TcpIpIpFragmentationConfig` and is located in the `TcpIpIpConfig` tab. The configuration parameters are:

- ▶ `TcpIpIpFragMemReserved`

Size of internal fragmentation and reassembly data in units of bytes (static memory allocation) - Memory required by post-build configuration must be smaller than this constant.

- ▶ `TcpIpIpFragmentationRxEnabled`

Enables or disables support for reassembling of incoming datagrams that are fragmented according to IETF RFC 815 (IP Datagram Reassembly Algorithms).



- ▶ true IP Datagram Reassembly enabled
- ▶ false IP Datagram Reassembly disabled

The following parameters configure the details of the reassembly mechanism:

- ▶ TcpIpIpReassemblyTimeout

Time after which an incomplete datagram gets discarded. RFC1122 (from 1989) suggests a value between 60 and 120 seconds. A large value can quickly lead to reassembly buffer exhaustion if fragments are lost.

- ▶ TcpIpIpReassemblyBufferCount

Number of fragmented IP datagrams that can be reassembled in parallel.

- ▶ TcpIpIpReassemblyBufferSize

Size of each reassembly buffer.

- ▶ TcpIpIpFragmentationTxEnabled

Enables or disables support for fragmenting outgoing datagrams according to IETF RFC 791 / RFC 2460.

Available choices:

- ▶ OFF

IP Datagram splitting disabled.

- ▶ OUTOFORDER

The header fragment with the checksum will be transmitted last to avoid buffering.

- ▶ INORDER

All data will be buffered in Ethernet transmit buffers, so the first fragment with the header and the checksum can be transmitted first. Additional data will be needed to keep track of the Ethernet buffer handles. This can be configured by the following values:

- ▶ TcpIpIpTxFragmentBufferCount

Maximum number of transmit buffers. Number of fragmented IP datagrams that can be sent in parallel.

- ▶ TcpIpIpTxFragmentBufferSize

Maximum size of a transmitted packet. INORDER fragmentation does not allocate memory for the data, but instead stores the data in Ethernet buffers. The maximum number of Ethernet buffers per packet is configured in TcpIpIpTxFragmentSegmentCount. Multiplying that with



the ethernet MTU size is the virtual buffer size, which is the limit for fragmented INORDER transmissions and must be configured here.

► `TcpIpIpTxFragmentSegmentCount`

Maximum number of transmit Ethernet buffers (fragments) per IP datagram and socket. Twelve bytes of data will be reserved per fragment and buffer to store the Ethernet buffer handles.

Rationale:

As the implementation is generic within the IP module and mostly independent of the IPv4 and IPv6 specifics, there can only be one configuration read in the init code. This is an IPv6-like configuration, using the IPv6 configuration parameter names (minus the IPv6 reference) and the EB extensions.

Hence the IPv4 and IPv6 parameters and references are unused in favor of the common configuration, which is placed in the IP configuration container.

Requirements:

ECUC\_Tcplp\_00077, ECUC\_Tcplp\_00078, ECUC\_Tcplp\_00079, ECUC\_Tcplp\_00080, ECUC\_Tcplp\_-00099, ECUC\_Tcplp\_00103, ECUC\_Tcplp\_00114, ECUC\_Tcplp\_00157, ECUC\_Tcplp\_00158, ECUC\_Tcplp\_00159, ECUC\_Tcplp\_00160, ECUC\_Tcplp\_00161, ECUC\_Tcplp\_00162

► [IPV4] Broadcast addresses must be explicitly configured for reception

Description:

If an IP datagram is received with a destination broadcast address, it is received only if there is an explicit broadcast local address configured, and there are sockets bound to this broadcast local address.

Rationale:

This allows detailed control of filtering the accepted destination IP addresses.

Requirements:

SWS\_TCPIP\_00106

► [IPV4] Only a single Unicast Internet address per physical/virtual interface is supported

Description:

This Tcplp implementation supports a single (unicast) internet address per physical/virtual interface (as specified in IETF RFC 791, Chapter 2.3).

Requirements:

SWS\_TCPIP\_00053



- ▶ [IPV4] IPv4 option fields are not supported

Description:

This Tcplp implementation ignores received Option fields as part of the IpV4 header (as specified in IETF RFC 791, Chapter 3.1). Transmission of Options as part of the IpV4 header is not supported.

Requirements:

SWS\_TCPIP\_00053

- ▶ [IPV4] Certain ICMP messages are not supported

Description:

This Tcplp implementation does not transmit the ICMP messages of type

- Parameter Problem Message

Time Exceeded Message

- Source Quench Message

- Redirect Message

- Timestamp Reply Message

(as specified in IETF RFC 792).

Requirements:

SWS\_TCPIP\_00059

- ▶ [IPV4] No local multicast loopback

Description:

This Tcplp implementation does not locally loop back transmitted multicast datagrams. Thus, multicast messages transmitted will not be received by the local node, even if it is assigned to the multicast address.

Rationale:

This is a use-case for systems with independent processes communicating via Tcplp. In AUTOSAR the local communication is usually performed within the RTE, thus this feature is assumed to be superfluous.

Requirements:

SWS\_TCPIP\_00097

- ▶ Unbound sockets will not be automatically closed



Description:

If the last Ethif controller reaches the Offline state, unbound sockets will not be automatically closed.

Rationale:

It is assumed that the upper layer (e.g. Soad) will close all unbound sockets if the Tcplp calls SoAd\_LocallpAddrAssignmentChg() with State TCPIP\_IPADDR\_STATE\_UNASSIGNED.

Requirements:

SWS\_TCPIP\_00077

- ▶ Tcplp\_TcpTransmit() does not queue data in state SYN-SENT and SYN-RECEIVED

Description:

If Tcplp\_TcpTransmit() is called with a socket in state SYN-SENT and SYN-RECEIVED the function reports TCPIP\_E\_NOTCONN to Det.

Rationale:

It is assumed that the upper layer will only transmit data in state ESTABLISHED after <Up>\_TcpConnected() or <Up>\_TcpAccepted() is called.

Requirements:

SWS\_TCPIP\_00061

- ▶ Tcplp\_Close() does not report TCPIP\_E\_NOTCONN to Det if socket is CLOSED

Description:

If Tcplp\_Close() is called with abort equals TRUE or FALSE and a socket in state CLOSED, <Up>-TcpEvent() with TCPIP\_TCP\_CLOSED is called.

Rationale:

If the socket is unused or bound the upper layer need to be informed that the socket can no longer be used.

Requirements:

SWS\_TCPIP\_00061

- ▶ Reception of SYN segment in state TIME-WAIT does not re-establish a connection

Description:

If SYN segment is received in state TIME-WAIT a reset is transmitted and the connection is closed.



Requirements:

SWS\_TCPIP\_00104

- ▶ Non-compliant deviations in the vendor-specific module definition file

Description:

The vendor-specific module definition file (VSMD) has non compliant deviations to the AUTOSAR specification:

The following configuration parameters are in the pre-compile configuration class instead of the link configuration class:

- ▶ TcpIpArpTableSizeMax
- ▶ TcpIpLocalAddrIpv4EntriesMax
- ▶ TcpIpLocalAddrIpv6EntriesMax
- ▶ TcpIpUdpSocketMax
- ▶ TcpIpTcpSocketMax

Rationale:

Making the parameters TcpIpArpTableSizeMax, TcpIpLocalAddrIpv4EntriesMax, TcpIpLocalAddrIpv6EntriesMax, TcpIpUdpSocketMax, TcpIpTcpSocketMax pre-compile configurable allows for significant performance optimizations.

- ▶ Non-compliant deviations in the vendor-specific module definition file

Description:

The vendor-specific module definition file (VSMD) has non compliant deviations to the AUTOSAR specification:

The valid multiplicity of the configuration parameter TcpIpNdpConfig is from 0 to 1, which exceeds the range of 1 to \* defined in the AUTOSAR specification.

Rationale:

Although, the configuration of NDP is required for IPv6, NDP does not need to be configured if the node supports IPv4 only.

- ▶ Non-compliant deviations in the vendor-specific module definition file

Description:

The vendor-specific module definition file (VSMD) has non compliant deviations to the AUTOSAR specification:



The valid range of the configuration parameter `TcpIpAssignmentPriority` is from 1 to 4, which exceeds the range of 1 to 3 defined in the AUTOSAR specification.

Rationale:

Extending the range of the configuration parameter `TcpIpAssignmentPriority` allows for the simultaneous configuration of all IPv4 assignment methods for one local address id.

- ▶ Tcplp does not support multiple configuration containers

Description:

Tcplp supports the configuration of a single container in the following lists only:

- `TcplpArpConfig`
- `TcplpAutolpConfig`
- `TcplpDhcpConfig`
- `TcplpV6DhcpConfig`
- `TcplpV6NdpConfig`

Requirements:

ECUC\_Tcplp\_00097, ECUC\_Tcplp\_00098, ECUC\_Tcplp\_00100, ECUC\_Tcplp\_00101, ECUC\_Tcplp\_00102

- ▶ Router functionality not supported

Description:

Tcplp does not support any router functionality.

Requirements:

SWS\_TCPIP\_00160, SWS\_TCPIP\_00163

- ▶ Tcplp generates an ICMPv6 error message when receiving a packet sent as a link-layer multicast.

Description:

If the Tcplp receives a packet sent as a link-layer multicast and the packet contains an error the Tcplp will respond with an ICMPv6 error message.

Requirements:

SWS\_TCPIP\_00163



Rationale:

The destination link layer address is not passed to the Tcplp and therefore it is not possible to detect a link-layer multicast address. Usually, a link-layer multicast address is sent in combination with an IPv6 multicast address. If a packet destined to an IPv6 multicast address is received an ICMPv6 error message is not generated.

- ▶ [IPV6] Tunneling mechanism not supported

Description:

Tcplp does not support the encapsulation of IPv4 in the IPv6 header or vice versa.

Requirements:

SWS\_TCPIP\_00160

- ▶ [IPV4] DhcpV4 server not supported

Description:

Tcplp does not implementation a DhcpV4 server.

Requirements:

SWS\_TCPIP\_00200, SWS\_TCPIP\_00201, SWS\_TCPIP\_00218, SWS\_TCPIP\_00058, ECUC\_Tcplp\_00183, ECUC\_Tcplp\_00195, ECUC\_Tcplp\_00187, ECUC\_Tcplp\_00190, ECUC\_Tcplp\_00189, ECUC\_Tcplp\_00188, ECUC\_Tcplp\_00191

- ▶ Allocated DHCP addresses cannot be stored

Description:

Tcplp does not support to store an IP address allocated through Dhcp in an Nvm block in the Nvm module.

Requirements:

SWS\_TCPIP\_00219, ECUC\_Tcplp\_00186

- ▶ Stored addresses cannot be reset

Description:

Tcplp does not support API Tcplp\_ResetIpAssignment for reseting all IP addresses stored in NvM block

Requirements:

SWS\_TCPIP\_00215, SWS\_TCPIP\_00216, SWS\_TCPIP\_00217, ECUC\_Tcplp\_00182

- ▶ [IPV4] IPv4 packet queuing not supported



Description:

Tcplp does not queue an IPv4 packet if the link layer address of the remote host does not exist in the ARP table and returns TCPIP\_E\_PHYS\_ADDR\_MISS to the caller.

Rationale:

If a UDP/ICMP frame is transmitted and an ARP entry does not exist the UDP/ICMP frame will be dropped and an ARP request will be transmitted instead. TCP frames will be retransmitted in the next mainfunction if an ARP entry does not exist. For UDP frames a packet queue can be configured in SoAd through configuration parameter SoAdSocketUdpRetryEnabled in SoAdSocketConnectionGroup. An alternative way would be to configure a TcpllpV4ArpPacketFilter callout function which is called for every IPv4 frame which is received and matches the configured IP address. Through this callout it is possible to decide by the return value of this function if an ARP entry shall be created for the remote host. Per default an ARP entry is not created when an IPv4 frame is received if none exists to avoid unnecessary ARP entries.

Requirements:

SWS\_TCPIP\_00191, SWS\_TCPIP\_00192, ECUC\_Tcplp\_00170

- ▶ [IPV6]Certain IPv6 Extension Headers are not supported

Description:

Tcplp does not support the reception and transmission of the following IPv6 Extension Headers:

- Authentication Header
- Encapsulating Security Payload Header

Tcplp does not support the transmission of the following IPv6 Extension Headers:

- Hop-by-Hop Options Header
- Routing Header
- Destination Options Header

Requirements:

SWS\_TCPIP\_00157, SWS\_TCPIP\_00160

- ▶ AUTOSAR API Tcplp\_RequestIpAddrAssignment v4.2.2 not supported

Description:

Tcplp implements the AUTOSAR API Tcplp\_RequestIpAddrAssignment according to v4.1.3.

Rationale:



Due to compatibility reasons to other modules (SoAd) the AUTOSAR API Tcplp\_RequestIpAddrAssignment will not be updated to v4.2.2

Requirements:

SWS\_TCPIP\_00037, SWS\_TCPIP\_00079

- ▶ AUTOSAR API Tcplp\_UdpTransmit v4.2.2 not supported

Description:

Tcplp implements the AUTOSAR API Tcplp\_UdpTransmit according to v4.1.3.

Rationale:

Due to compatibility reasons to other modules (SoAd) the AUTOSAR API Tcplp\_UdpTransmit will not be updated to v4.2.2

Requirements:

SWS\_TCPIP\_00025

- ▶ [IPV6] Certain rules of the IPv6 Source Address Selection are not supported

Description:

- The Tcplp does not support Rule 4: Prefer home addresses as described in (IETF RFC 6724). Rationale: IPv6 Mobility (IETF RFC 3375) which introduces home addresses is not required in AUTOSAR.

- The Tcplp does not support Rule 6: Prefer matching label as described in (IETF RFC 6724). Rationale: Prefix policy table cannot be configured in AUTOSAR.

- The Tcplp does not support Rule 7: Prefer temporary addresses. as described in (IETF RFC 6724). Rationale: According to SWS\_TCPIP\_00166 temporary addresses are not required in AUTOSAR.

Requirements:

SWS\_TCPIP\_00154

- ▶ [IPV6] IPv6 Loop back messages not supported

Description:

Tcplp does not verify if a neighbor solicitation which is sent to probe for a duplicate address is looped back. If a looped back neighbor solicitation is received, the Tcplp will interpret the neighbor solicitation as duplicate and will not assign the IP address to the interface

Requirements:

SWS\_TCPIP\_00157, SWS\_TCPIP\_00159



- ▶ [IPV6] Autoconfiguration issues related to MLD not supported

Description:

Tcplp does not send Multicast Listener Discovery messages. Tcplp does not support delaying of Neighbor Solicitation messages

Requirements:

SWS\_TCPIP\_00157

- ▶ [IPV6] IPv6 packet queuing not supported

Description:

Tcplp does not queue an IPv6 packet if the link layer address of the remote host does not exist in the NDP table and returns TCPIP\_E\_PHYS\_ADDR\_MISS to the caller.

Rationale:

If a UDP/ICMP frame is transmitted and a NDP entry does not exist the UDP/ICMP frame will be dropped and an Neighbor Solicitation will be transmitted instead. TCP frames will be retransmitted in the next main-function if an NDP entry does not exist. For UDP frames a packet queue can be configured in SoAd through configuration parameter SoAdSocketUdpRetryEnabled in SoAdSocketConnectionGroup.

Requirements:

SWS\_TCPIP\_00164, SWS\_TCPIP\_00165, SWS\_TCPIP\_00193, SWS\_TCPIP\_00194, ECUC\_Tcplp\_-00171, SWS\_TCPIP\_00164

- ▶ [IPV6] UDP/TCP IPv6 socket does not support IPv4 transmission

Description:

- A UDP/TCP IPv6 socket does not allow to bind an IPv4 local address.

- A UDP IPv6 socket does not support to transmit messages to an IPv4 address embedded in an IPv6 address.

- A TCP IPv6 socket does not support to connect to an IPv4 address embedded in an IPv6 address.

Requirements:

SWS\_TCPIP\_00162

- ▶ [IPV6] The DhcpV6 client does not collect multiple advertise messages

Description:



The DhcpV6 client does not wait and buffer multiple advertise messages until the first RT time elapses before responding to advertise message, instead it responds to first valid advertise message that is received.

Rationale:

Memory reservation if not required should be avoided.

Requirements:

SWS\_TCPIP\_00166

- ▶ [IPV6] API function Tcplp\_IcmpV6Transmit() is not supported

Description:

Tcplp does not support the API function Tcplp\_IcmpV6Transmit().

Rationale:

Tcplp\_IcmpTransmit is used to transmit an icmp message instead of Tcplp\_IcmpV6Transmit

Requirements:

SWS\_TCPIP\_00187, SWS\_TCPIP\_00230

- ▶ [IPV6] IpV6 configuration parameters not supported

Description:

The following IpV6 configuration parameters are not supported:

- TcplpVXCtrl

Requirements:

ECUC\_Tcplp\_00094

- ▶ [IPV6] DhcpV6 configuration parameters not supported

Description:

The following DhcpV6 configuration parameters are not supported:

- TcplpDhcpV6CnfDelayMax

- TcplpDhcpV6CnfDelayMin

- TcplpDhcpV6InfDelayMax

- TcplpDhcpV6InfDelayMin



- TcplpDhcpV6SolDelayMax
- TcplpDhcpV6SolDelayMin

Requirements:

ECUC\_Tcplp\_00116, ECUC\_Tcplp\_00117, ECUC\_Tcplp\_00118, ECUC\_Tcplp\_00119, ECUC\_Tcplp\_-  
00120, ECUC\_Tcplp\_00121

- ▶ [IPV6] Dynamic reconfiguration of MTU via Router Advertisements is not supported

Description:

The Tcplp does not update the MTU according to the value received in a router advertisement, the configured MTU (EthIfCtrlMtu) is used instead.

Requirements:

SWS\_TCPIP\_00153, SWS\_TCPIP\_00157, SWS\_TCPIP\_00160, SWS\_TCPIP\_00164

- ▶ [IPV6] Dynamic reconfiguration of hop limit via Router Advertisements is not supported

Description:

The Tcplp does not update the hop limit according to the value received in a router advertisement, the configured hop limit (TcplpUdpTtl, TcplpTcpTtl, TcplpIcmpTtl, TcplpIcmpV6HopLimit) is used instead.

Requirements:

SWS\_TCPIP\_00164, SWS\_TCPIP\_00157

- ▶ [IPV6] Dynamic reconfiguration of reachable time via Router Advertisements is not supported

Description:

The Tcplp does not update the reachable time according to the value received in a router advertisement, the configured reachable time (TcplpNdpDefaultReachableTime) is used instead.

Requirements:

SWS\_TCPIP\_00164, SWS\_TCPIP\_00157

- ▶ [IPV6] Dynamic reconfiguration of retransmit timer via Router Advertisements is not supported

Description:

The Tcplp does not update the retransmit timer according to the value received in a router advertisement, the configured retrnasmit timer (TcplpNdpDefaultRetransTimer) is used instead.

Requirements:



SWS\_TCPIP\_00164, SWS\_TCPIP\_00157

- [IPV6] Ndp configuration parameters not supported

Description:

The following Ndp configuration parameters are not supported:

- TcpIpNdpDynamicHopLimitEnabled
- TcpIpNdpDynamicMtuEnabled
- TcpIpNdpDynamicReachableTimeEnabled
- TcpIpNdpDynamicRetransTimeEnabled
- TcpIpNdpAddressResolutionUnreachabilityDetectionEnabled
- TcpIpNdpMinRandomFactor
- TcpIpNdpMaxRandomFactor
- TcpIpNdpDefaultReachableTime
- TcpIpNdpNeighborUnreachabilityDetectionEnabled
- TcpIpNdpRandomReachableTimeEnabled

Requirements:

ECUC\_TcpIp\_00146, ECUC\_TcpIp\_00147, ECUC\_TcpIp\_00148, ECUC\_TcpIp\_00145, ECUC\_TcpIp\_-  
00091, ECUC\_TcpIp\_00134, ECUC\_TcpIp\_00135, ECUC\_TcpIp\_00130, ECUC\_TcpIp\_00136, ECUC\_-  
TcpIp\_00137

- [IPV6] Certain DhcpV6 message types are not supported

Description:

DhcpV6 does not support the transmission/reception of the following message types:

- Information-request Message
- Release Message
- Confirm Message
- Reconfigure Message

Requirements:



SWS\_TCPIP\_00166

- ▶ [IPV6] Multiple IA\_ADDR options in DhcpV6 messages are not supported

Description:

DhcpV6 does not support the transmission of the multiple IA\_ADDR options in single IA\_NA option. DhcpV6 shall only process last IA\_ADDR option and ignore others in single IA\_NA option.

Requirements:

SWS\_TCPIP\_00166

- ▶ [IPV6] IA\_NA options with times T1 == T2 in DhcpV6 messages are not supported

Description:

DhcpV6 discards IA\_NA options with times T1 == T2

Requirements:

SWS\_TCPIP\_00166

- ▶ Oversized ICMPv4/v6 Echo Reply is not fragmented

Description:

If the Tcplp receives an Echo Request greater than the MTU the Tcplp will not transmit the Echo Reply in IP fragments, it will truncate the size of the Echo Reply to the MTU instead and send the reply in a single IP frame.

Requirements:

SWS\_TCPIP\_00163, SWS\_TCPIP\_00059

- ▶ [IPV6] Passing of ICMPv6 error messages to upper layer is not supported

Description:

UDP and TCP do not evaluate an ICMP error message from a remote host. The ICMP message is passed to the configurable Up\_IcmpMsgHandler instead.

Requirements:

SWS\_TCPIP\_00163

- ▶ [IPV6] Limitation of ICMPv6 packet transmissions not supported

Description:



Tcplp does not allow to configure a limit for the transmission of ICMPv6 error messages to the same destination and ICMPv6 error messages transmissions per second.

Requirements:

SWS\_TCPIP\_00163

- ▶ [IPV6] IcmpV6 Time Exceeded Message not supported

Description:

Tcplp does not transmit a IcmpV6 Time Exceeded Message when the Tcplp cannot complete the reassembly due to missing fragments within the configured time limit. It will discard the datagram only.

Requirements:

SWS\_TCPIP\_00161, SWS\_TCPIP\_00163

- ▶ Scalability classes are not supported

Description:

The configuration parameter TcplpScalabilityClass is not supported.

Requirements:

SWS\_TCPIP\_00148, SWS\_TCPIP\_00149, SWS\_TCPIP\_00150, ECUC\_Tcplp\_00169

- ▶ ICMP destination unreachable not supported

Description:

The Tcplp does not transmit a ICMP destination unreachable with code 3 (Address Unreachable) for each packet queued for address resolution if the node does not receive a Neighbor Advertisement after the transmission of the maximal number of Multicast Neighbor Solicitations. Instead, the neighbor is removed from the neighbor cache.

Requirements:

SWS\_TCPIP\_00164

- ▶ [IPV6] Automatic assignment of IPv6 link local address not supported

Description:

The Tcplp does not automatically assign the IPv6 link local address to every configured controller

Rationale:

The IPv6 link local address is configurable for every interface



Requirements:

SWS\_TCPIP\_00162

- ▶ [IPV6] identification via interface ID not supported

Description:

Tcplp does not use interface ID to identify interfaces on a link

Requirements:

SWS\_TCPIP\_00162

- ▶ [IPV6] loopback address not supported

Description:

This Tcplp implementation does not locally loop back messages transmitted to the loop back address and does prevent the assignment of a loop back address

Requirements:

SWS\_TCPIP\_00162

- ▶ [IPV6] multicast address reserved fields not supported

Description:

This Tcplp implementation ignores the reserved field of an IPV6 multicast address

Requirements:

SWS\_TCPIP\_00162

- ▶ [IPv6] DUID-LL shall not be used if network interface is not permanently attached to the device

Description:

DUID-LL (DHCP Unique Identifier - Link-layer Address) is used by the Tcplp to identify a server in messages where a server needs to be identified.

Rationale:

DUID-LL (DHCP Unique Identifier - Link-layer Address) is the only way to implement a DUID in Tcplp because there is no parameter in AUTOSAR to set an Enterprise Number [DUID-EN] and no clock to generate a DUID-LLT (DHCP Unique Identifier - Link-layer address plus time). Moreover the network interface of an ECU will most likely not change over time

Requirements:



SWS\_TCPIP\_00166

- ▶ [IPv6] Source Address Selection of unbound IPv6 UDP sockets

Description:

Tcplp does not select an interface that has a local address (TcplpAddrId) which uses the same network prefix as the destination address if data is transmitted using an unbound IPv6 UDP socket and then performs source address selection for the selected interface. Instead, the Tcplp loops over all local addresses and performs source address selection.

Requirements:

SWS\_TCPIP\_00185

- ▶ Anycast addresses not supported

Description:

Tcplp does not support the assignment of Anycast addresses to a Tcplp controller. Messages can still be transmitted to an Anycast address.

Requirements:

SWS\_TCPIP\_00269, SWS\_TCPIP\_00162

- ▶ IPv6 Redirect message not supported

Description:

Tcplp does not process received IPv6 Redirect messages, it discards the messages instead

Requirements:

SWS\_TCPIP\_00281, SWS\_TCPIP\_00164

- ▶ Configurable Path MTU discovery not supported

Description:

The path MTU discovery cannot be turned on or off for a socket through Tcplp\_ChangeParamter and paramId

- TCPIP\_PARAMID\_PATHMTU\_ENABLE

Requirements:

SWS\_TCPIP\_00267, SWS\_TCPIP\_00268

- ▶ [IPV6] Certain sections of the IPv6 Subnet Model are not supported



Description:

- The Tcplp does not support section 4: Host Rules in (IETF RFC 5942).
- The Tcplp does not support Section 6: updated definition of “on-link” in (IETF RFC 5942).

Requirements:

SWS\_TCPIP\_00265

- ▶ Some runtime errors not supported

Description:

Module Tcplp does not support the following runtime errors:

- ▶ TCPIP\_E\_TIMEDOUT
- ▶ TCPIP\_E\_CONNREFUSED
- ▶ TCPIP\_E\_HOSTUNREACH
- ▶ TCPIP\_E\_PACKETTOBIG
- ▶ TCPIP\_E\_DADCONFLICT

Requirements:

SWS\_TCPIP\_00157, SWS\_TCPIP\_00255, SWS\_TCPIP\_00256, SWS\_TCPIP\_00257, SWS\_TCPIP\_-  
00258, SWS\_TCPIP\_00259, SWS\_TCPIP\_00282

- ▶ [IPv6] On-link prefix list not supported

Description:

Tcplp does not support the following configuration parameters for on-link prefix configuration:

- TcplpNdpPrefixList
- TcplpNdpPrefixListEntry
- TcplpNdpPrefixListEntryPrefixLength
- TcplpNdpPrefixListEntryPrefixAddress

Requirements:

ECUC\_Tcplp\_00205, ECUC\_Tcplp\_00206, ECUC\_Tcplp\_00207, ECUC\_Tcplp\_00208

- ▶ Tcplp does not support notifications in case of a detected address conflict

Description:



The Tcplp does not notify the configuring agent when an IPv6 address conflict is detected during Ongoing Address Conflict Detection.

Requirements:

SWS\_TCPIP\_00283

- ▶ Tcplp allows IPv6 UDP packets with zero checksum.

Description:

Since introduction of configurable UDP checksum calculation UDP packets with checksum field set to zero are accepted if UDP checksum calculation is disabled

Rationale:

Per default the UDP packet is discarded if it contains a zero checksum but zero checksum fields can be allowed through Tcplp\_ChangeParameter.

Requirements:

SWS\_TCPIP\_00185

- ▶ Tcplp does not support variant handling for the following Tcplp parameters

Description:

The Tcplp does not support postbuild selectable or loadable for

- TcplpSocketOwnerUpperLayerType

Rationale:

Due to the fact that the underlying socket owner parameter, e.g. TcplpSocketOwnerCopyTxDataName, TcplpSocketOwnerLocallpAddrAssignmentChgName, ... are link time configurable it is not applicable that TcplpSocketOwnerUpperLayerType is postbuild selectable or loadable

Requirements:

ECUC\_Tcplp\_00174

- ▶ Tcplp does not support ESP

Description:

Tcplp does not support ESP. If an ESP Header is detected, the frame will be dropped!

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.



- ▶ Tcplp does not support Tunnel mode

Description:

Tcplp does not support Tunnel mode. If an inner IP header is detected, the frame will be dropped!

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ Tcplp does not support filtering by DSCP

Description:

Security Association database entry does not contain DSCP values and DSCP-specific filtering is not applied

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ Tcplp does not support to configure Multicast Security Associations

Description:

Multicast communication can only be performed unprotected and shall be configured as bypassed in the Security Policy Table

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ Tcplp does not support nested SAs

Description:

Tcplp allows to apply a single AH to a frame in the SPD only and does not support to combine multiple AHs or ESPs. If the Tcplp encounters more than one AH or an ESP in a frame, the frame will be dropped

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ Tcplp does not support Security Gateways

Description:

Tcplp does not support the following Security Gateway (SG) features:

- ▶ Discovery of and communication through SGs.
- ▶ Acting as a SG.



Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ Tcplp does not support additional SPD creation or changes during the runtime

Description:

Tcplp only allows the configuration of static security policies. Security policies do not change during runtime and cannot be updated, added or deleted through an API

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ Tcplp does not support mobility

Description:

Mobility Support in IPv6 as defined in IETF RFC 6275 and IP Mobility Support for IPv4 as defined in IETF RFC 5944 is not implemented

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ Certain IPsec error are not logged

Description:

Tcplp does not provide any kind of audit logs and does not log the following IPsec error:

- ▶ Invalid SPI received
- ▶ Sequence number overflow
- ▶ IP Fragment passed to AH processing
- ▶ No security association found
- ▶ ICV validation failed

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301 and 4302.

- ▶ Tcplp does not provide any support automated SA management/generation

Description:

Tcplp does not support the following automated SA management/generation protocols

- ▶ Internet Key Exchange version 1 (IKEv1) defined in IETF RFC 4109



- ▶ Internet Key Exchange Protocol Version 2 (IKEv2) defined in IETF RFC 7296

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301, 4302, 4494, 4543 and 4868

- ▶ Certain HMAC algorithms defined in IETF RFC 4868 are not supported

Description:

Tcpip does not support the following HMAC integrity algorithm for AH

- ▶ HMAC-SHA-384-192
- ▶ HMAC-SHA-512-256

Tcpip does not support the following HMAC integrity algorithm transformers for IKEv2

- ▶ HMAC-SHA-256-128
- ▶ HMAC-SHA-384-192
- ▶ HMAC-SHA-512-256

Tcpip does not support the following HMAC pseudo random function transformers for IKEv2

- ▶ PRF-HMAC-SHA-256
- ▶ PRF-HMAC-SHA-384
- ▶ PRF-HMAC-SHA-512

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4868.

- ▶ Certain GMAC algorithms defined in IETF RFC 4543 are not supported

Description:

Tcpip does not support the following GMAC integrity algorithm for AH

- ▶ AUTH\_AES\_192\_GMAC

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4543.

- ▶ GMAC Initialization Vector can repeat

Description:

Tcpip allows use of GMAC with statically configured keys, without re-keying IV can repeat

Requirements:



No AUTOSAR requirement. Deviation from IETF RFC 4543.

- ▶ Tcplp does not support anti-replay service and ESN according to IETF RFC 4302

Description:

Tcplp does not support anti-replay service and extended sequence number (ESN)

Rationale:

Manually configured security association shall not use anti-reply and Extended sequence number because the security association might not be synchronized (e.g. one of the two hosts might restart and reset the sequence number counter)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4302.

- ▶ Tcplp does not provide support for partially matching incoming packets to SAD entries.

Description:

Tcplp only supports full matching of incoming packets to Security Association Database entry, i.e. SPI, destination address and source address of the incoming packet has to the values of an SA in the database for a successful match. Partial matches e.g. SPI and destination address or SPI only are not supported.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4302.

- ▶ Tcplp does not provide support PFP flag

Description:

Tcplp does not support to configure the "populate from packet" flag for an SPD entry which states if the value (e.g. next header protocol, ip address) for an SAD entry shall be taken from the packet or the SPD. The Tcplp always takes the value from the SPD entry.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ Tcplp does not support ICMP filtering based on ICMP Code and Type

Description:

Tcplp does not support to configure an SPD entry based on the ICMP Code and Type. It is only possible to configure all ICMP frames for a specific local and remote IP address to be either secured, bypassed or discarded.



Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ TcpIp does not support certain identifier types

Description:

TcpIp does not support configuration in the SPD and identification of a remote host by the following identifier types:

- ▶ Fully qualified DNS name
- ▶ Fully qualified user name string (email)
- ▶ X.500 distinguished name
- ▶ Byte string

TcpIp identifies a remote host by an IP address only.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ TcpIp does not support the Sequence Counter Overflow in SAD

Description:

TcpIp does not support the configuration of a Sequence Counter Overflow flag which indicates if sequence number overflow is permitted. For security association which are manually configured the sequence number is not checked and it can overflow.

Rationale:

Manually configured security association shall not use anti-reply and Extended sequence number because the security association might not be synchronized (e.g. one of the two hosts might restart and reset the sequence number counter)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ TcpIp does not send ICMP Error Message when received frame is discarded

Description:

TcpIp does not send an ICMP error message when a frame is discarded in the following cases:

- ▶ No matching SPD entry was found



- ▶ The Tcplp reached the remote peer but was unable to negotiate the SA required by the SPD entry matching the packet because the remote peer is administratively prohibited from communicating with the initiator, the initiating peer was unable to authenticate itself to the remote peer, the remote peer was unable to authenticate itself to the initiating peer, or the SPD at the remote peer did not have a suitable entry.
- ▶ The Tcplp was unable to set up the SA required by the SPD entry matching the packet because the IPsec peer at the other end of the exchange could not be contacted.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ Tcplp does not support SPD cache

Description:

Tcplp does not support caching of recently used/created SPD. Additional entries cannot be added during runtime. Statically configured table is used for lookup.

Rationale:

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ Tcplp does not support SPD ID

Description:

Tcplp does not support SPD-ID instead it directly searches SPD-S and SPD-O

Rationale:

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

### 3.3.7.5. 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.

- ▶ Limitation on number of entries in container `TcpIpLocalAddr`

Description:



The Tcplp can only handle 253 local addresses. This limitation applies to configuration parameter `TcpIp-pLocalAddr`.

Rationale:

LocalAddrId 254 and 255 are reserved for special values.

- ▶ Limitation on configuration parameter `TcpIpArpTableEntryTimeout`

Description:

The range for the parameter `TcpIpArpTableEntryTimeout` is restricted to 1..65535 seconds or Infinity. Infinity indicates that when an entry is created in the ARP table it will never be removed.

Rationale:

0 cannot be configured and corresponds to Infinity.

- ▶ Limitation on number of predefined, static, unicast assignments

Description:

The Tcplp can only handle a single unicast assignment with assignment method `TCPIP_STATIC` per EthIf controller.

Rationale:

This limitation allows a reduced code complexity. Concurrency between multiple static assignments of described type must not be handled.

- ▶ Handling of illegal option length

Description:

If the Tcplp encounters a TCP segment with an illegal option length it will drop the segment but will not transmit a reset as suggested in <http://tools.ietf.org/html/rfc1122>, chapter 4.2.2.5.

- ▶ TCP Quiet Time Concept

Description:

If the Tcplp crashes, it will not delay emitting any TCP segments for at least the agreed Maximum Segment Lifetime (MSL) as suggested in <http://tools.ietf.org/html/rfc793>, chapter 3.3 "TCP Quiet Time Concept".

- ▶ Precedence and Security

Description:

The Tcp does not evaluate the Precedence and Security of receiving TCP segments and does not include the options in the IP header in any TCP segments.



► IP Identification

Description:

If a retransmitted TCP segment is identical to the original packet, the TCP uses a different IP Identification field.

► DHCP options

Description:

The DHCP client does not support the following DHCP Options and BOOTP Vendor Extensions:

- Time Offset
- Time Server Option
- Name Server Option
- Domain Name Server Option
- Log Server Option
- Cookie Server Option
- LPR Server Option
- Impress Server Option
- Resource Location Server Option
- Host Name Option
- Boot File Size Option
- Merit Dump File
- Domain Name
- Swap Server
- Root Path
- Extensions Path
- IP Forwarding Enable/Disable Option
- Non-Local Source Routing Enable/Disable Option
- Policy Filter Option
- Maximum Datagram Reassembly Size
- Default IP Time-to-live
- Path MTU Aging Timeout Option
- Path MTU Plateau Table Option
- Interface MTU Option



- ▶ All Subnets are Local Option
- ▶ Broadcast Address Option
- ▶ Perform Mask Discovery Option
- ▶ Mask Supplier Option
- ▶ Perform Router Discovery Option
- ▶ Router Solicitation Address Option
- ▶ Static Route Option
- ▶ Trailer Encapsulation Option
- ▶ ARP Cache Timeout Option
- ▶ Ethernet Encapsulation Option
- ▶ TCP Default TTL Option
- ▶ TCP Keepalive Interval Option
- ▶ TCP Keepalive Garbage Option
- ▶ Network Information Service Domain Option
- ▶ Network Information Servers Option
- ▶ Network Time Protocol Servers Option
- ▶ Vendor Specific Information
- ▶ NetBIOS over TCP/IP Name Server Option
- ▶ NetBIOS over TCP/IP Datagram Distribution Server Option
- ▶ NetBIOS over TCP/IP Node Type Option
- ▶ NetBIOS over TCP/IP Scope Option
- ▶ X Window System Font Server Option
- ▶ X Window System Display Manager Option
- ▶ Message
- ▶ Maximum DHCP Message Size
- ▶ Class-identifier
- ▶ Client-identifier
- ▶ DHCPINFORM messages

Description:

The DHCP client does not support the DHCPINFORM message. The DHCP client does not inform a DHCP server when obtaining an IP address through other means (e.g. manual configuration).



RFC 2131 describes the optional mechanism of the DHCPINFORM message. (see chapter 3.4.). This DHCP client implementation does not support this option.

► **DHCPRELEASE messages**

Description:

The DHCP client does not support the DHCPRELEASE message. The DHCP client does not inform a DHCP server that an IP address is no longer used.

RFC 2131 describes the optional mechanism of the DHCPRELEASE message. (see chapter 3.1.). This DHCP client implementation does not support this option.

► **Reusing previously allocated network addresses**

Description:

The DHCP client does not support to reuse a previously allocated network address to omit some of the steps for obtaining a network address. If the DHCP client wants to obtain a network address it always starts with sending a DHCPDISCOVER message.

RFC 2131 describes the optional mechanism for reusing a previously allocated network addresses. (see chapter 3.2.). This DHCP client implementation does not support this option.

► **The DHCP client continues using the previous network address**

Description:

The DHCP client silently discards a DHCPACK message with a different acknowledged network address than the IP address in the preceding DHCP Request. If the DHCP client wants to obtain a network address it always starts with sending a DHCPDISCOVER message.

► **Simple DHCPDISCOVER message transmission**

Description:

The simple DHCP client will transmit a DHCPDISCOVER message to the MAC and IP broadcast.

► **DAD duplicate address reinitialization**

Description:

When static or link local ipv6 address is detected as duplicate during Duplicate address detection when TcplpNdpSlaacOptimisticDadEnabled is enabled, it can only be reassigned by reinitialization of the Tcplp.

► **Mulitple UDP socket binds on same local address id and local port**

Description:



If a UDP socket is bound to a Unicast local address A and a local port B, a second UDP socket cannot be bound to the same local address A and local port B.

If a UDP socket is bound to a Multicast address A and a local port B it is possible to bind multiple UDP sockets to the same Multicast address A and local port B.

If a UDP socket is bound to a controller ANY address A and a local port B it is possible to bind multiple UDP socket to the same controller ANY address A and local port B. However if additional UDP sockets are bound to Unicast local address C or controller ANY address A and a local port B, only one of the sockets can receive messages addressed to the Unicast local address C. The same applies when multiple UDP sockets are bound to TCPIP\_LOCALADDRID\_ANY

- ▶ Range restriction on configuration parameter `TcpIpTcpKeepAliveProbesMax`

Description:

The range for the parameter `TcpIpTcpKeepAliveProbesMax` is restricted to 0..255 keep alive probes.

- ▶ Discarding DhcpV4 messages because of the lease time

Description:

If received DhcpV4 lease time is greater than 0xFFFFFFFF/7 (0x24924924) seconds, the message shall be silently discarded.

### 3.3.7.6. Open-source software

Tcplp does not use open-source software.

## 3.3.8. UdpNm module release notes

- ▶ AUTOSAR R4.1 Rev 3
- ▶ AUTOSAR SWS document version: 3.3.0
- ▶ Module version: 2.9.8.B466224
- ▶ Supplier: Elektrobit Automotive GmbH

### 3.3.8.1. Change log

This chapter lists the changes between different versions.



### Module version 2.9.8

2021-10-27

- ▶ Improved support for calling UdpNm APIs that rely on the current state from the context of the state change notification

### Module version 2.9.7

2021-06-25

- ▶ ASCUDPNM-386 Fixed known issue: UdpNm might not compile if memory mapping is used
- ▶ ASCUDPNM-392 Fixed known issue: UdpNm might unexpectedly not transmit NM messages after (re)entering Normal Operation State or Repeat Message State

### Module version 2.9.6

2021-03-05

- ▶ Added support for postbuild selectable config of UdpNmMsgCycleOffset

### Module version 2.9.5

2020-10-23

- ▶ ASCUDPNM-359 Fixed known issue: First NM message that is sent on the bus carries an outdated state change information when Normal Operation state is entered from Ready Sleep state
- ▶ Improved Active wakeup Bit functionality

### Module version 2.9.4

2020-06-19

- ▶ ASCUDPNM-353 Fixed known issue: UdpNm sends wrong number of Immediate Nm messages when transmission request(s) fail

### Module version 2.9.3

2020-02-21

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



### Module version 2.9.2

2019-10-11

- ▶ Changed maximum value for UdpNmPnInfoOffset to 31 and default value for UdpNmPnInfoLength to 1
- ▶ ASCUDPNM-327 Fixed known issue: Linker errors are reported due to incorrect memory mapping
- ▶ Changed generation of UdpNmNodeId based on channel specific UdpNmNodeIdEnabled
- ▶ ASCUDPNM-326 Fixed known issue: Active wake-up bit in CBV is incorrectly set when the network mode is reentered
- ▶ Changed UdpNm\_SoAdlfRxIndication to be reentrant for the same channel but with different pduids

### Module version 2.9.1

2019-06-14

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

### Module version 2.9.0

2019-02-15

- ▶ Improved robustness check for references, optional parameters property and enable parameters property
- ▶ ASCUDPNM-289 Fixed known issue: State change notification can be sent repeatedly from RepeatMessage state
- ▶ Implemented Post-build selectable support

### Module version 2.8.0

2018-10-26

- ▶ Implemented Multi-core support
- ▶ ASCUDPNM-263 Fixed known issue: UdpNm generates an invalid basic software module description if no configuration set is provided
- ▶ Changed the accesses in UdpNm to configuration
- ▶ ASCUDPNM-269 Fixed known issue: Generator error if Com user data is enabled and passive mode is set to true

### Module version 2.7.4

2018-06-22

- ▶ Added first message sending retry mechanism



### Module version 2.7.3

2018-02-16

- ▶ ASCUDPNM-183 Fixed known issue: API UdpNm\_CheckRemoteSleepIndication returns wrong status in Ready Sleep State
- ▶ Removed AUTOSAR 3.x compliant symbolic name value macros and updated the logic to only provide AUTOSAR 4.0.2 compliant macros

### Module version 2.7.2

2017-09-22

- ▶ Added UdpNmNodeEnabled, UdpNmRepeatMsgIndEnabled and UdpNmNodeDetectionEnabled as per channel configurable.
- ▶ ASCUDPNM-180 Fixed known issue: User data might be inconsistent if UdpNm\_SetUserData is interrupted by main function
- ▶ ASCUDPNM-179 Fixed known issue: Code generation error for user data with length zero
- ▶ ASCUDPNM-175 Fixed known issue: Existance of PDU referenced by UdpNmPnEraRxNSduRef in PduR is not checked
- ▶ ASCUDPNM-176 Fixed known issue: Tx timeout exception is generated for a channel which works correctly
- ▶ ASCUDPNM-174 Fixed known issue: UdpNmPnEiraRxNSduRef shall be available based on parameter UdpNmPnEiraCalcEnabled

### Module version 2.7.1

2017-06-30

- ▶ Added support for Ethernet Switch Port Groups
- ▶ Implemented support for Car Wake Up
- ▶ ASCUDPNM-163 Fixed known issue: Compilation error occurs if all UdpNmPnFilterMaskByteValues are set to zero
- ▶ ASCUDPNM-164 Fixed known issue: Out of bounds access if at postbuild more PNCs are configured than at precompile time

### Module version 2.7.0

2017-03-31

- ▶ Added compatibility of UdpNm\_SoAdIfRxIndication to ASR4.1.2



- ▶ Changed UserTxConfPduld member in the UdpNm\_ChannelDataType structure
- ▶ Removed UdpNmRepeatMessageTime - UdpNmMsgCycleTime multiplicity constrain
- ▶ ASCUDPNM-142 Fixed known issue: Wrong dependency UdpNmStateChangeIndEnabled - UdpNmPassiveModeEnabled
- ▶ Added support for immediate transmission
- ▶ ASCUDPNM-144 Fixed known issue: The user data transmitted in the NM PDU could be inconsistent
- ▶ Improved UdpNm could reject user requests even if the requests should be accepted in the current state
- ▶ Added support for PDU Length greater than 8 Bytes

#### **Module version 2.6.0**

2016-10-31

- ▶ ASCUDPNM-113 Fixed known issue: Incorrect consistency check of UdpNmComUserDataSupport against UdpNmUserDataEnabled
- ▶ Added support for UdpNmImmediateRestartEnabled

#### **Module version 2.5.0**

2016-06-24

- ▶ ASCUDPNM-96 Fixed known issue: Handle ID wizard error when computing UdpNmRxPduld
- ▶ Integrate the new handle ID policy from asc\_HidWiz

#### **Module version 2.4.0**

2016-02-10

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

#### **Module version 2.3.0**

2015-11-06

- ▶ Change the class of UdpNmPnHandleMultipleNetworkRequests to PostBuild

#### **Module version 2.2.0**

2015-06-24



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

#### **Module version 2.1.0**

2015-02-20

- ▶ Removed the `UdpNmNodeDetectionEnabled` constraint from the `editable` section of `UdpNmNodeId`
- ▶ Corrected issue: State machine gets stuck in repeat message state if repeat message time is zero
- ▶ Implemented multiplicity for `UdpNmRxPdu`
- ▶ Added configuration check for preventing the generation of Tx PDU as empty
- ▶ Replaced legacy UUIDs from the `UdpNm.xdm` file
- ▶ Removed `UdpNm_HsmMain` function

#### **Module version 2.0.0**

2015-01-13

- ▶ Updated to AUTOSAR version 4.1.3 (SWS UDP Network Management version 3.3.0)

#### **Module version 1.4.0**

2014-10-02

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

#### **Module version 1.3.0**

2014-04-28

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

#### **Module version 1.2.3**

2013-10-11

- ▶ Added debug instrumentation based on BSWMD
- ▶ Internal module improvement. This module version update does not affect module functionality

#### **Module version 1.2.2**

2013-06-14

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



### Module version 1.2.1

2013-04-08

- ▶ Updated SoAd API to V2.0.24 (`SoAdIf_Transmit` to `SoAd_IfTransmit`)

### Module version 1.2.0

2013-02-21

- ▶ Updated reference paths of `UdpNm` `ComMChannel` reference for the introduction of `ComMConfigSet` container

### Module version 1.1.0

2012-12-12

- ▶ Updated `AdjacentLayer.properties` according to new property file format
- ▶ Updated to AUTOSAR SWS UDP Network Management 2.0.6

### Module version 1.0.0

2012-07-12

- ▶ First release of `UdpNm` module

#### 3.3.8.2. New features

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

#### 3.3.8.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- ▶ COM Rx user data

Description:

COM Support for Rx user data is added.

- ▶ New container `UdpNmUserDataRxPdu` is added to configure the Rx Pdu of received user data.
- ▶ User can enable or disable this container.

When this feature is enabled, then user must configure the respective Pdu in EcuC and provide correct routing path in PduR. When this feature is disabled, the user can still receive data using `UdpNm_GetUser-Data()` API.



#### Rationale:

User has freedom of receiving the user data over COM.

- ▶ Allow a configuration where some of the channels support user data and some not

#### Description:

As per AUTOSAR requirement SWS\_UdpNm\_00086, when `UdpNmUserDataEnabled` is enabled, the `UdpNmUserDataLength` should not be zero.

The module deviates from this requirement. The module allows a user to configure a mixture of channels where some channels support user data and some channels doesn't support user data.

#### Rationale:

More flexibility and freedom of configuration for user is achieved.

- ▶ Function tracing support via AUTOSAR Debugging

#### Description:

The module UdpNm 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
- ▶ Support for Side Allocation

#### Description:

The Side Allocation feature allow flashing of two different ECUs with the same software. The behaviour of each ECU will differ at runtime based on a flag(eg: stored in EEPROM or the level of a pin).

The following parameter differ between the two variants: `UdpNmNodeId` UdpNm supports configuring a callout function to be called everytime the UdpNm module needs to retrieve a NodId for an ECU.

- ▶ Extended the range for the offset of the PN request information in the NM message

#### Description:

Starting from AUTOSAR requirement ECUC\_UdpNm\_00068, `UdpNmPnInfoOffset` range has changed from 1..7 to 1..31.



Rationale:

More flexibility and freedom of configuration for user is achieved.

### 3.3.8.4. Deviations

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

- ▶ UdpNmNodeld shall be available only in case UdpNmNodeldEnabled is set

Description:

Contrary to specifications UdpNmNodeld shall be available only in case UdpNmNodeldEnabled is set for the channel.

Requirements:

ECUC\_UdpNm\_00031

- ▶ UDPNM\_E\_PARAM\_POINTER not supported

Description:

In case a pointer is passed to a function and that pointer is Null Pointer UDPNM\_E\_NULL\_POINTER shall be reported to DET instead of UDPNM\_E\_PARAM\_POINTER.

Requirements:

SWS\_UdpNm\_00018

- ▶ UDPNM\_E\_UNINIT not supported

Description:

In case a function is called from the module and module is uninitialized UDPNM\_E\_NO\_INIT shall be reported to DET instead of UDPNM\_E\_UNINIT.

Requirements:

UdpNm.ASR431.SWS\_UdpNm\_00039, SWS\_UdpNm\_00039

- ▶ Invalid channel handle not used in UdpNm\_SoAdlfTxConfirmation

Description:

UdpNm\_SoAdlfTxConfirmation informs the DET if it's called with an invalid chanle handle. The parameter used for DET is UDPNM\_E\_INVALID\_PDUID instead of UDPNM\_E\_INVALID\_CHANNEL.

Requirements:



### EB\_SWS\_UdpNm\_00229\_1

- ▶ UdpNm\_TriggerTransmit feature is not available

Description:

Functionality regarding UdpNm\_TriggerTransmit is not implemented.

Requirements:

UdpNm.ASR431.SWS\_UdpNm\_91001, UdpNm.ASR431.SWS\_UdpNm\_00377, UdpNm.ASR431.SWS\_UdpNm\_00378

- ▶ UdpNm does not handle PduR\_UdpNmTriggerTransmit returning E\_NOT\_OK

Description:

It is assumed that in case PduR\_UdpNmTriggerTransmit returns E\_NOT\_OK it won't change the input pointer's data.

Requirements:

SWS\_UdpNm\_00365

- ▶ UdpNmRepeatMsgIndEnabled should not be true if UdpNmPassiveModeEnabled is true

Description:

UdpNmRepeatMsgIndEnabled have no implemented dependency: If (UdpNmPassiveModeEnabled == False) then Equal (NmRepeatMsgIndEnabled) else Equal(False).

Requirements:

UdpNm.ASR431.ECUC\_UdpNm\_00092

- ▶ UdpNmMainFunctionPeriod range changed with AUTOSAR 4.3.1 specifications

Description:

Parameter UdpNmMainFunctionPeriod is implemented with range: 0.001 .. 0.255 and AUTOSAR V4.3.-1 specify range: 0 .. INF.

Requirements:

UdpNm.ASR431.ECUC\_UdpNm\_00032

- ▶ UdpNmUserDataTxPdu changed with AUTOSAR 4.3.1

Description:



According to AUTOSAR V4.3.1 UdpNmUserDataTxPdu is a preprocessor that switch for enabling the Tx path of Com User Data. UdpNmUserDataTxPdu is a container not a preprocessor switch.

Requirements:

UdpNm.ASR431.ECUC\_UdpNm\_00056

- ▶ UdpNmComUserDataSupport dependency not implemented

Description:

Dependency for UdpNmComUserDataSupport not implemented (if UdpNmPassiveModeEnabled == True then UdpNmComUserDataSupport = False).

Requirements:

UdpNm.ASR431.ECUC\_UdpNm\_00055

- ▶ UdpNmNodeIdEnabled specified per channel and global too, with AUTOSAR 4.3.1

Description:

UdpNmNodeIdEnabled should be a per channel parameter and not both global and per channel too.

Requirements:

UdpNm.ASR431.ECUC\_UdpNm\_00008

- ▶ UdpNmNodeDetectionEnabled specified per channel and global too, with AUTOSAR 4.3.1

Description:

UdpNmNodeDetectionEnabled should be a per channel parameter and not both global and per channel too.

Requirements:

UdpNm.ASR431.ECUC\_UdpNm\_00007

- ▶ UdpNmRepeatMsgIndEnabled specified per channel and global too, with AUTOSAR 4.3.1

Description:

UdpNmRepeatMsgIndEnabled should be a per channel parameter and not both global and per channel too.

Requirements:

UdpNm.ASR431.ECUC\_UdpNm\_00015

- ▶ UdpNm\_SoAdIfTxConfirmation syntax and service ID not according to ASR4.3.1



Description:

-The `UdpNm_SoAdlfTxConfirmation` existing syntax: void `UdpNm_SoAdlfTxConfirmation(PduldType UdpNmTxPduld)` AUTOSAR V4.3.1 syntax: void `UdpNm_SoAdlfTxConfirmation(PduldType TxPduld, Std_ReturnType result)` -Service ID is 0x0f instead of 0x40. -Function `UdpNm_SoAdlfTxConfirmation` has no parameter result. (UdpNm.ASR431.SWS\_UdpNm\_00316)

Requirements:

`UdpNm.ASR431.SWS_UdpNm_00228`, `UdpNm.ASR431.SWS_UdpNm_00316`, `UdpNm.ASR431.SWS_UdpNm_00379`

- ▶ `UdpNm_SoAdlfRxIndication` Service ID changed with ASR4.3.1

Description:

Service ID changed from value 0x10 to 0x42.

Requirements:

`UdpNm.ASR431.SWS_UdpNm_00231`

- ▶ The transmission of NM PDUs not start within the next NM main function

Description:

The transmission of NM PDUs is not started within the next NM main function according to ASR 4.3.1. The `UdpNm` module starts the `UdpNm` Message Cycle Timer with `UDPNM_MSG_CYCLE_OFFSET` in order to start transmission of Network Management PDUs.

Requirements:

`UdpNm.ASR431.SWS_UdpNm_00178`

- ▶ The NULL Pointer check of input parameters is not done for `UdpNm_Init`

Description:

If detection of development errors is enabled by `UDPNM_DEV_ERROR_DETECT` (configuration parameter), validity checks for all input parameters shall be performed for each UDP NM API service call according to ASR 4.3.1. The NULL Pointer check of input parameters is not done for `UdpNm_Init`.

Requirements:

`UdpNm.ASR431.SWS_UdpNm_00196`

- ▶ `UdpNm_Transmit` feature is not available

Description:



Functionality regarding UdpNm\_Transmit is not implemented.

Requirements:

UdpNm.ASR431.SWS\_UdpNm\_00464, UdpNm.ASR431.SWS\_UdpNm\_00313, UdpNm.ASR431.SWS\_-  
UdpNm\_00315

- ▶ UdpNmCarWakeUpBytePosition range diffrent than specified in AUTOSAR

Description:

The range for parameter UdpNmCarWakeUpBytePosition is 0..4294967294 and not 0..7 as specified by AUTOSAR

Requirements:

ECUC\_UdpNm\_00086, UdpNm.ASR431.ECUC\_UdpNm\_00086

- ▶ UdpNmUserDataLength is not supported in this release

Description:

UdpNmUserDataLength is not supported in this release. The length of user data is defined as RxPdu Length Node - (Node Id + CBV)

Requirements:

ECUC\_UdpNm\_00027

- ▶ Link Time support

Description:

Link Time configuration of parameters is not supported.

Requirements:

SWS\_UdpNm\_00081, UdpNm.ASR431.SWS\_UdpNm\_00081, UdpNm.ASR431.ECUC\_UdpNm\_00058,  
UdpNm.ASR431.ECUC\_UdpNm\_00091, UdpNm.ASR431.ECUC\_UdpNm\_00090

- ▶ UdpNmActiveWakeupBitEnabled

Description:

Parameter UdpNmActiveWakeupBitEnabled should be a per channel but it's a global one.

Requirements:

ECUC\_UdpNm\_00074, UdpNm.ASR431.ECUC\_UdpNm\_00074

- ▶ UdpNmMainFunctionPeriod



Description:

Parameter UdpNmMainFunctionPeriod should be a per channel but it's a global one. (reference to product description: ASCPD-274)

Requirements:

ECUC\_UdpNm\_00032

- ▶ UdpNmDemEventParameterRefs is not supported.

Description:

Parameters - UdpNmDemEventParameterRefs - UdpNmDemEventParameterRefs/UDPNM\_E\_NET-WORK\_TIMEOUT - UdpNmDemEventParameterRefs/UDPNM\_E\_TCPIP\_TRANSMIT\_ERROR are not supported in this release.

Requirements:

ECUC\_UdpNm\_00050, ECUC\_UdpNm\_00052, ECUC\_UdpNm\_00053

- ▶ UdpNmCoordinatorId is not supported.

Description:

Parameter UdpNmCoordinatorId is not supported in this release.

Requirements:

ECUC\_UdpNm\_00041, UdpNm.ASR431.ECUC\_UdpNm\_00041

- ▶ UdpNmCoordinatorEnabled is not supported.

Description:

UdpNmCoordinatorEnabled is not supported in this release.

Requirements:

ECUC\_UdpNm\_00040, UdpNm.ASR431.ECUC\_UdpNm\_00040

- ▶ Nm\_TxTimeoutException is not supported

Description:

The Nm\_TxTimeoutException API and UdpNmMsgTimeoutTime parameters are not supported in this release.

Requirements:

ECUC\_UdpNm\_00030, UdpNm.ASR431.ECUC\_UdpNm\_00030, UdpNm.ASR431.SWS\_UdpNm\_00379



- ▶ DEM is not supported

Description:

The DEM functionality is not supported in this release.

Requirements:

SWS\_UdpNm\_00190, UdpNm.ASR431.SWS\_UdpNm\_00190, ECUC\_UdpNm\_00193, ECUC\_UdpNm\_00194, ECUC\_UdpNm\_00050, ECUC\_UdpNm\_00053, ECUC\_UdpNm\_00052

- ▶ UdpNm\_SetCoordBits API is not supported

Description:

UdpNm\_SetCoordBits API is not supported in this release

Requirements:

SWS\_UdpNm\_00222, ECUC\_UdpNm\_00040, ECUC\_UdpNm\_00041, UdpNm.ASR431.ECUC\_UdpNm\_00041

- ▶ Coordinator Synchronization Support

Description:

Coordinator Synchronization Support is not supported in this release

Requirements:

SWS\_UdpNm\_00320, UdpNm.ASR431.SWS\_UdpNm\_00320, SWS\_UdpNm\_00364, UdpNm.ASR431.-SWS\_UdpNm\_00364, SWS\_UdpNm\_00321, UdpNm.ASR431.SWS\_UdpNm\_00321, SWS\_UdpNm\_00322, UdpNm.ASR431.SWS\_UdpNm\_00322, ECUC\_UdpNm\_00059, UdpNm.ASR431.ECUC\_UdpNm\_00059, SWS\_UdpNm\_00324, UdpNm.ASR431.SWS\_UdpNm\_00324

- ▶ Initialization check in main function

Description:

If the main function is called while the module is not yet initialized the main function returns immediately without performing any functionality and without raising any Det error. This initialization check is always performed independent of the development error detection setting.

Rationale:

The SchM module may schedule the modules main function before the module is initialized. This would result in lots of Det errors during start up. Therefore the module's main function does not throw a Det error if the module is not yet initialized and simply returns in this case.

Requirements:



SWS\_UdpNm\_00234, UdpNm.ASR431.SWS\_UdpNm\_00234

► Initialization of user data

Description:

The requirement SWS\_UdpNm\_00025 describes that during initialization the UdpNm module shall set each byte of the user data to 0xFF.

In contrast to this, only those user data bytes which are not part of partial networking information are initialized to 0xFF. The bytes allocated for partial networking information are initialized to 0.

Rationale:

SWS\_UdpNm\_00348 states if a NM-PDU is send by the UdpNm, the UdpNm module shall set every PN request bit in the EIRA to 1 that has been requested by the PN request bits in the transmitted NM-PDU. Therefore, requirement SWS\_UdpNm\_00025 and SWS\_UdpNm\_00348 together will lead to a behavior such that all partial networks are requested even if no partial network has ever been requested.

Additional information:

[http://www.autosar.org/bugzilla/show\\_bug.cgi?id=53631](http://www.autosar.org/bugzilla/show_bug.cgi?id=53631)

Requirements:

SWS\_UdpNm\_00025, UdpNm.ASR431.SWS\_UdpNm\_00025

► Dependency between UdpNmTimeoutTime and UdpNmMsgCycleTime

Description:

The requirement ECUC\_UdpNm\_00020 states that the configuration parameter UdpNmTimeoutTime must be a multiple of the value of the configuration parameter UdpNmMsgCycleTime.

This dependency shall be removed since this dependency does not exist and therefore is wrong.

Rationale:

The system requirements of at least one customer requires timing settings where UdpNmTimeoutTime is not a multiple of UdpNmMsgCycleTime.

Additional information:

[http://www.autosar.org/bugzilla/show\\_bug.cgi?id=54115](http://www.autosar.org/bugzilla/show_bug.cgi?id=54115)

Requirements:

ECUC\_UdpNm\_00020, UdpNm.ASR431.ECUC\_UdpNm\_00020

► Changes in Symbolic Name References



Description:

In order to create distinct symbolic name references as specified within the ECU configuration [ecuc\_sws\_2108], the parameter `UdpNmRxPduId` is referred as `UdpNmConf_UdpNmChannelConfig_<CHANNELNAME>_UdpNmRxPdu`, `UdpNmTxConfirmationPduId` as `UdpNmConf_UdpNmChannelConfig_<CHANNELNAME>_UdpNmTxPdu` and `UdpNmTxUserDataPduId` as `UdpNmConf_UdpNmChannelConfig_<CHANNELNAME>_UdpNmUserDataTxPdu`.

Rationale:

It is not possible to create or change short name values of containers with the multiplicity of one within the EB tresos Studio. Therefore the symbolic names generated as specified in [ecuc\_sws\_2108] will not be unique.

- ▶ Changes regarding `UdpNmMsgCycleOffset`

Description:

Parameter `UdpNmMsgCycleOffset` is treated as post-build selectable parameter.

Requirements:

`ECUC_UdpNm_00029`, `UdpNm.ASR431.ECUC_UdpNm_00029`

### 3.3.8.5. 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.

### 3.3.8.6. Open-source software

`UdpNm` does not use open-source software.



## 4. ACG8 IP Stack user guide

### 4.1. Overview

The ACG8 IP Stack user guide provides information about the concepts of the IP stack in the AUTOSAR context and about the configuration of the IP Stack modules. Before you read this user guide, read the general concepts about communication stacks in AUTOSAR that are described in the EB tresos AutoCore Generic documentation.

- ▶ [Section 4.2, “Background information”](#) describes the concept of IP communication in the AUTOSAR context.
- ▶ [Section 4.3, “Configuring the ACG8 IP Stack”](#) advises on the configuration of IP Stack features that involve more than one IP Stack module.
- ▶ [Section 4.4, “DoIP module user guide”](#)
- ▶ [Section 4.5, “Sd module user guide”](#)
- ▶ [Section 4.6, “SoAd module user guide”](#)
- ▶ [Section 4.7, “Tcplp module user guide”](#)
- ▶ [Section 4.8, “QoS Support user guide”](#)

### 4.2. Background information

This chapter provides general information about the IP Stack communication concepts in the AUTOSAR context. If you are not familiar with the general concepts of communication in AUTOSAR, read the general information provided in the EB tresos AutoCore Generic documentation first.

#### 4.2.1. Modules of the AUTOSAR IP Stack

The IP Stack comprises the following modules:

- ▶ SoAd
- ▶ TcpIp
- ▶ EthIf
- ▶ EthSM



- ▶ SomeIpTp
- ▶ UdpNm
- ▶ Sd
- ▶ DoIP

The following sections provide further details about the `SoAd`, `TcpIp` `EthIf`, `EthSM`, and `UdpNm` modules. You find general information about network management and state management modules in the EB tresos AutoCore Generic documentation chapter [Network management and state management stack](#).

The AUTOSAR IP stack provides an Ethernet/IP communication stack including the TCP/IP protocol family, which is fully integrated into the AUTOSAR communication architecture. Thus, the IP stack facilitates the communication between software components and diagnostic communication via an Ethernet/IP network. In the following, the individual modules of the AUTOSAR IP stack are described in detail.

#### Socket Adaptor (`SoAd`)

The Socket Adaptor (`SoAd`) module is located above the `TcpIp` module in the AUTOSAR layered model. The `SoAd` maps AUTOSAR PDUs (which are identified by a unique PDU identifier) to the network endpoints of TCP connections and/or to the UDP datagrams (identified by a 4-tuple of local/remote IP address and port number) by means of static configuration tables. In doing so, the `SoAd` provides an abstraction from TCP specific functionality, e.g., from the different methods for local IP address assignment, i.e. stateless address auto-configuration according to IETF RFC 3927, DHCP according to IETF RFC 2131, or simply by means of static configuration.

It also provides an abstraction from the details regarding the connection setup and tear-down in order to provide an AUTOSAR PDU-based interface to the `PduR`. Similar to the packing of multiple PDUs into a single FlexRay frame in the `FrIf`, the `SoAd` allows for packing multiple PDUs into a single UDP datagram or TCP segment in order to achieve a decent header/payload ratio. This reduces the overhead for connection setup and tear-down.

Additionally, the `SoAd` implements the services specified by the ISO DoIP standard (see Section 4.3) and presents itself as a conventional AUTOSAR transport protocol (i.e. like CanTp or FrTp) to the `PduR` by providing the respective API. This enables the exchange and relaying of diagnostic data with an external tester device, i.e. the Diagnostics over IP (`DoIp`) module.

#### TCP/IP (`TcpIp`)

For the Ethernet communication network, AUTOSAR decided to re-use already well-proven protocols. These are the internet protocol (IPv4 and IPv6), the internet control message protocol (ICMP), the address resolution protocol (ARP) for IPv4 address resolution and the neighbor discovery protocol (NDP) for IPv6 address resolution, the user datagram protocol (UDP) for unreliable connectionless communication, the transmission control protocol (TCP) for reliable connection-oriented communication, and the dynamic host configuration protocol (DHCP) for automated IP address assignment. The functionality of these protocols is implemented in the `TcpIp` module.

The EB tresos AutoCore Generic 8 Security Extensions additionally supports Internet Protocol Security (IPsec) and Internet Key Exchange Protocol Version 2 (IKEv2) to realize interoperable, high quality, crypto-



graphically-based security for IPv4 and IPv6 in an AUTOSAR context. A detailed description can be found in the EB tresos AutoCore Generic 8 Security Extensions product documentation.

#### Ethernet Interface (`EthIf`)

Using the frame-based services provided by the Ethernet driver module (`Eth`), the `EthIf` facilitates the sending and the reception of protocol data units (PDUs). The `EthIf` additionally provides support for virtual local area networks (VLANs) by taking care of the handling of the VLAN tags, i.e. the tag protocol identifier (TPID) and the tag control information (TCI). Thus, the `EthIf` provides an abstraction to upper layers that hides the difference between a VLAN and "normal" LANs.

#### Ethernet State Manager (`EthSm`)

The Ethernet State Manager module (`EthSM`) facilitates the state management of the Ethernet communication controller and Ethernet transceiver with respect to Ethernet-specific startup and shutdown features like link state detection and IP address assignment and provides a common state machine API to Communication Manager (`ComM`). This API consists of functions for requesting the communication modes FULL COMMUNICATION and NO COMMUNICATION.

#### SOME/IP Transport Protocol (`SomeIpTp`)

The SOME/IP Transport Protocol module (`SomeIpTp`) deals with UDP messages that do not fit in a single UDP packet. On transmission, the message is disassembled to smaller segments. Each can fit in one UDP packet. These segments are sent sequentially (one each `SomeIpTp` main function). On reception, `SomeIpTp` reassembles segments that belong to the same message in the correct sequence. If a segment is dropped or duplicated, the reception is cancelled.

#### UDP Network Management (`UdpNm`)

The UDP Network Management module (`UdpNm`) coordinates the transitions between normal operation and a low-power (= sleep) state of the Ethernet network. This coordination takes place by exchanging network management PDUs that the `UdpNm` sends and receives via the `SoAd`.

Similar to the interface API of other networks, I-PDUs that fit into a single frame of the underlying Ethernet network can be transmitted connectionless over the UDP protocol.

Large data, i.e. I-PDUs that do not fit into a single frame of the underlying Ethernet network, can be transmitted as a segmented data stream over the TCP protocol.

## 4.2.2. IP stack dependencies

In addition to the dependencies described in the EB tresos AutoCore Generic documentation, the Ethernet Transceiver Driver (`EthTrcv`) module depends on the Ethernet Driver (`Eth`) module. The `EthTrcv` uses the Media Independent Interface (MII) of the `Eth` module to gain access to registers in the transceiver hardware.



### 4.2.3. Network management in AUTOSAR IP stack

The network and state management is described in the EB tresos AutoCore Generic documentation concept chapter "Network management and state management stack". You find information about the concepts of the network and state management in AUTOSAR. You also learn how to configure the stack.

## 4.3. Configuring the ACG8 IP Stack

### 4.3.1. Multiple provided service instances

Within the topology of the ECU network in a car, there can be services that are similar. These services may be provided on different endpoints but should be processed by the same ECU.

Different instances of the same service may have different semantic meanings, for example:

- ▶ Instance 1: front left headlight
- ▶ Instance 2: front right headlight

These two instances are `Sd` servers that provide the exact same service and are received over the same local IP address and port at `Sd` client side. To distinguish each instance at SWC level, the service for each instance needs to be mapped to different signals/PDUs. Because the SOME/IP protocol does not include the instance ID in its SOME/IP header, it is required that `Sd` defines the routing of multiple provided service instances based on `Sd` control information. This is done by enabling individual routing groups for each connected server instance.

To enable the `Sd` client ECU to interact with multiple `Sd` server instances that provide the same service, you must also configure the `SoAd` module.

From the Socket Adaptor point of view, the different service instances are configured as different PDUs. These PDUs are configured for different `SoAd` socket route destinations, which are bundled under one `SoAd` socket route with one header ID representing the same service for all destinations. Corresponding to that, there must be one `SoAd` socket connection group that has one socket connection for each socket route destination. The relation of socket connection to socket route destination is dynamically set by `Sd` through its routing group reference.



#### 4.3.1.1. Configuring the SoAd module



Configuring the SoAd

##### Step 1

In SoAd/SoAdConfig/SoAdRoutingGroup, set up a SoAd routing group for each server service instance.

##### Step 2

In SoAd/SoAdConfig/SoAdSocketConnectionGroup, set up a SoAd socket connection group with a socket connection for each server service instance.

##### Step 3

In SoAd/SoAdConfig/SoAdSocketRoute, set up a SoAd socket route with a socket route destination for each server service instance.

##### Step 4

In SoAd/SoAdConfig/SoAdSocketRoute/SoAdSocketRouteDest/SoAdRxRoutingGroupRef, reference the SoAd Rx routing group according to the routing groups configured in [Section 4.3.1.2, “Configuring the Sd module”](#).

#### 4.3.1.2. Configuring the Sd module



Configuring the Sd

##### Step 1

In Sd/SdConfig/SdInstance/SdClientService, set up a client service instance as a counterpart to each server service instance.

##### Step 2

Each client service instance must have the same SdClientServiceId in Sd/SdConfig/SdInstance/SdClientService/SdClientServiceId.

##### Step 3

In Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup, set up an Sd consumed event group for each client service.

##### Step 4

In Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup/SdConsumedEventGroupUdpActivationRef, refer to an existing routing group reference from SoAd (see [Section 4.3.1.1, “Configuring the SoAd module”](#)).



## 4.3.2. Debug support

EB tresos AutoCore provides general mechanisms that can help you with debugging a problem you encounter during configuration, such as function entry-exit macros and the Development Error Tracer. For details, see chapter *Debugging support* in the EB tresos AutoCore Generic documentation.

In addition, the ACG8 IP Stack provides the GetAndResetMeasurementData service that you can use with any ACG8 IP Stack module. This service is described in the following section. For module-specific debug support, the `TcpIp` provides the IPsec error logger. For details, see [Section 4.7.2.5, “Debug support in Tcplp”](#).

### 4.3.2.1. GetAndResetMeasurementData service

This is a service for counting specified events like dropped UDP frames. The function is called when the specific condition occurs. Every call increments the counter. The counter can also be reset via a function call. You can configure different counters for different conditions. In that case, each counter must have an own ID.

The main disadvantage of the service is that a general counter is used for all occurrences of the same event. This makes the service useless if a finer granularity is required, e.g. in order to identify the socket connections that cause these frame drops.



Configuring the GetAndResetMeasurementData service

#### Step 1

For each required module, set the configuration parameter

`<module>GetAndResetMeasurementDataApi` to TRUE.

This enables the respective API `<module>GetAndResetMeasurementData()` to read and optionally reset the counter. The API can be called from integration code.

## 4.4. DoIP module user guide

### 4.4.1. Overview

This user guide describes the `DoIP` module. From this user guide you will learn more about the basic functionality of the `DoIP`. You will also learn which related modules are necessary to configure the `DoIP` module. The `DoIP` module reference provides further information on configuring the `DoIP` itself.

This user guide is intended for readers who have good knowledge of AUTOSAR and about the purpose of the `DoIP`. The information provided here should help you to integrate the `DoIP` in your AUTOSAR project.

- ▶ Section [Section 4.4.2, “Background Information”](#) provides an overview of the basic functionality of the DoIP.
- ▶ Section [Section 4.4.3, “Configuring DoIP”](#) provides information on related modules that are needed in order to configure the DoIP.
- ▶ For details on configuring the DoIP itself, refer to the parameter descriptions provided in the DoIP module reference [Chapter 5, “ACG8 IP Stack module references”](#).

## 4.4.2. Background Information

The DoIP module provided by Elektrobit (EB) implements the AUTOSAR basic software module Diagnostics over IP (DoIP).

The DoIP module provides diagnostic access to vehicle external testing devices via Ethernet/IP. It facilitates communication between these external testing devices and diagnostic components inside the vehicle network.

The DoIP module (highlighted in yellow) is located between the AUTOSAR PDU Router (PduR) module and the AUTOSAR Socket Adaptor (SoAd) module as shown in [Figure 4.1, “The AUTOSAR DoIP module”](#).

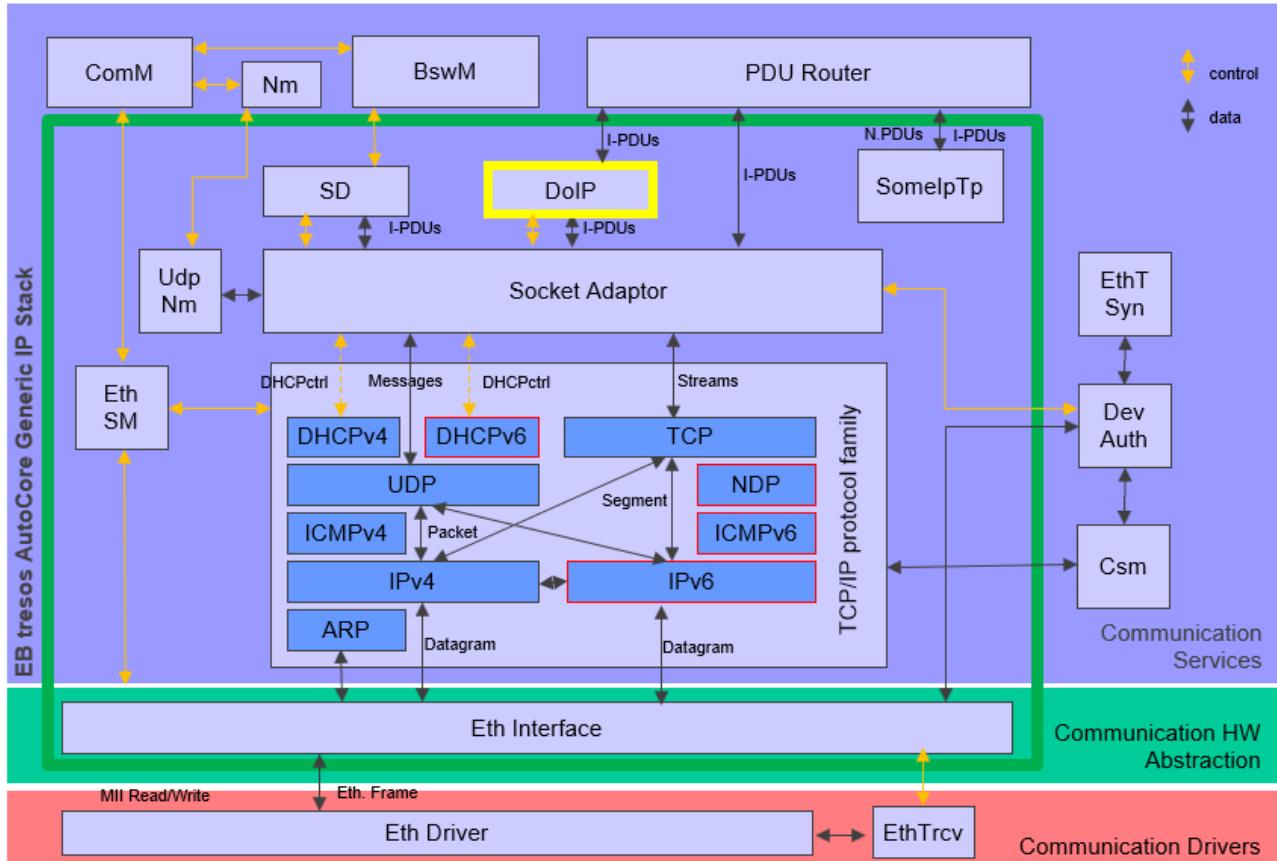


Figure 4.1. The AUTOSAR DoIP module



### 4.4.3. Configuring DoIP

To configure the `DoIP` module, add the module to your project using EB tresos Studio. Parameter descriptions are provided to guide the configuration. You find these in the module references section of this document. You also find these in the parameter description in EB tresos Studio.

To use the `DoIP` module, you must configure additional modules as outlined below:

- ▶ IP Stack: The `DoIP` module communicates with the `SoAd` and the `TcpIp` modules that are part of the IP Stack.
- ▶ COM Services: The `DoIP` module communicates with the `PduR` module which is part of COM Services.
- ▶ Diagnostic Stack: The `DoIP` module communicates with the `Dcm` module which is part of the Diagnostic Stack.

## 4.5. Sd module user guide

### 4.5.1. Overview

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

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

### 4.5.2. Background information

The `Sd` module provided by Elektrobit (EB) implements the AUTOSAR basic software module Service Discovery (`Sd`).

The `Sd` module offers functionality to manage the availability of, and the subscription to, functional entities called *services* in the in-vehicle network by exchanging SOME/IP-SD messages. A service can contain *methods* that can be called from other ECUs. A service also contains *events* to which other ECU services can subscribe. To ensure that event messages produced by these services are only sent to receivers subscribed to these services, the `Sd` module controls the transmission path of the event messages provided by a service during run-time.



The `Sd` module (highlighted in yellow) is located between the AUTOSAR Basic Software Mode Manager (BswM) module and the AUTOSAR Socket Adaptor (SoAd) module as shown in [Figure 4.2, “The AUTOSAR Sd module”](#).

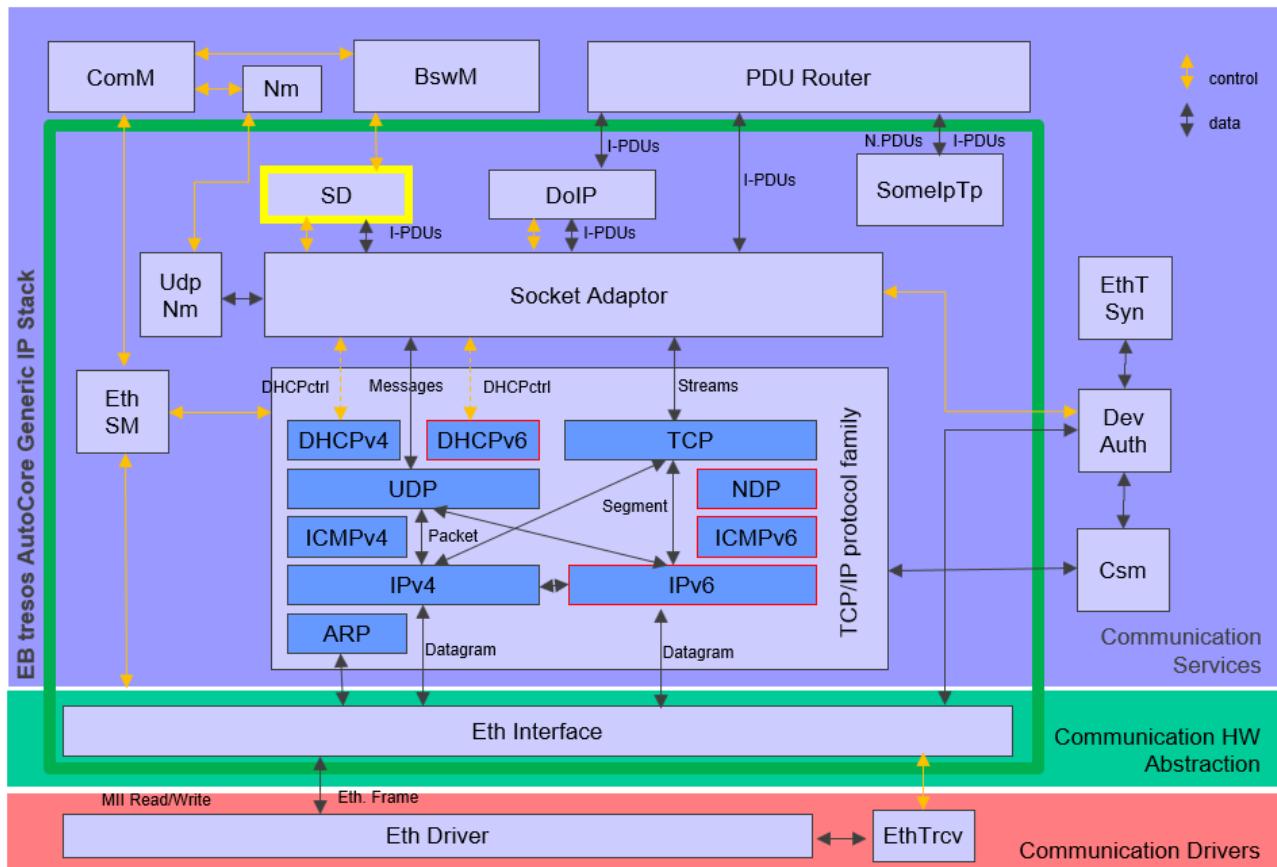


Figure 4.2. The AUTOSAR Sd module

### 4.5.3. Configuring the Sd module

To configure the `Sd` module, add the module to your project using EB tresos Studio. Parameter descriptions are provided to guide the configuration. You find these in the module references section of this document. You also find these in the parameter description in EB tresos Studio.

To use the `Sd` module, you must configure additional modules as listed below:

- `SoAd` module: The `Sd` module communicates directly with the `SoAd` module that is part of the IP Stack.



## 4.6. SoAd module user guide

### 4.6.1. Overview

This chapter provides you with SoAd-specific information:

- ▶ [Section 4.6.2, “Background information”](#) explains the concepts of the SoAd module.
- ▶ [Section 4.6.3, “Configuring the SoAd module”](#) provides instructions on how to configure the SoAd module.

### 4.6.2. Background information

#### 4.6.2.1. N-PDU buffer pooling

The Socket Adaptor can collect multiple PDUs to save bandwidth. This is called the N-PDU feature in AU-TOSAR. These PDUs are stored in Tx buffers. By default, there is one fixed buffer assigned to each SoAdSocketConnection. To save RAM, it is possible to share the buffer within a SoAdSocketConnectionGroup. This is called N-PDU buffer pooling.

If the N-PDU buffer pooling is used, the SoAdSocketUdpRetryEnabled feature also uses the buffers of the buffer pooling.

The buffer pooling only makes sense if the number of SoAdSocketConnections that can use the buffers is greater than the number of buffers reserved for this SocketConnectionGroup. Therefore, if the number entered for the buffer pool size equals the number of SoAdSocketConnections for this SoAdSocketConnectionGroup, fixed buffers are generated.

For information on how to configure the N-PDU buffer pooling, see [Section 4.6.3.1, “Configuring the N-PDU buffer pooling”](#).

##### 4.6.2.1.1. Temporary unavailable buffer

If no buffer is available during a transmit request, the resulting behavior differs for the three possible scenarios as described in the following table.

Transmit request scenario	Behavior
With the UDP socket unused	Indirect sending is applied.



Transmit request scenario	Behavior
▶ With the UDP socket in use AND ▶ Same SocketConnectionGroup AND ▶ Different SocketConnection	The PDU is dropped.
▶ With the UDP socket in use already without a buffer AND ▶ Same SocketConnection AND ▶ Different PDU	The PDU is dropped.

Table 4.1. Buffer unavailability and resulting behavior

#### 4.6.2.1.2. Debug hooks

During the development phase, you can overwrite the debug hooks to create statistics on how often UDP PDU drops occur due to buffer misses in relation to UDP PDUs sent:

- ▶ SOAD\_BUFFERPOOLING\_UDPPDUSENT (SoConGroupId) : Debug hook that gets called whenever a UDP frame is sent.
- ▶ SOAD\_BUFFERPOOLING\_UDPPDUDROP (SoConGroupId) : Debug hook that gets called whenever a UDP frame is dropped due to a missed buffer allocation.
- ▶ SOAD\_BUFFERPOOLING\_CONVERT\_SOCONGROUPID\_TO\_SOCONFIRSTID (SoConGroupId, SoConFirstIdPtr) : Function-like macro that converts the system internal value SoConGroupId to the config parameter of the first SoAdSocketId.

### 4.6.3. Configuring the SoAd module

#### 4.6.3.1. Configuring the N-PDU buffer pooling



Configuring the SoAd

The buffer pooling is organized on a SoAdSocketConnectionGroup level. The number of buffers in the buffer pool for a SoAdSocketConnectionGroup is configured with the SoAdTxBufferSize configuration parameter. This parameter triggers the generation of `SoAdSocketnPduUdpTxBufferMin X SoAdTxBufferSize` buffers for this SoAdSocketConnectionGroup. These buffers are shared among the SoAdSocketConnections.



Additionally, one shadow buffer per SoAdSocketConnectionGroup can be generated to store data during the time that a Tx buffer gets transmitted to avoid PDU drops. This is done by setting the `SoAdEnableShadowBufferSize` parameter.

#### Step 1

Make sure that `SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp/SoAdSocketnPduUdpTxBufferMin` is set.

#### Step 2

In `SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp/SoAdTxBufferSize`, configure a number of Tx buffers that should be used for the pooling within this `SoAdSocketConnectionGroup`.

#### Step 3

In `SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp/SoAdEnableShadowBufferSize`, configure a size for the shadow buffer. This buffer is only used during sending and only for the `SocketConnection` that is already active.

## 4.7. TcpIp module user guide

### 4.7.1. Overview

This user guide provides you with `TcpIp`-specific information:

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

For details on individual `TcpIp` configuration parameters and APIs, see the descriptions provided in the `TcpIp` module reference [Chapter 5, “ACG8 IP Stack module references”](#).

### 4.7.2. Background Information

#### 4.7.2.1. Software hardening support

The `TcpIp` stack provides the following software hardening features:

- ▶ [TCP SYN cookies](#)



► [Unpredictable sequence numbers](#)

#### 4.7.2.1.1. TCP SYN cookies

TCP SYN cookies provide greater protection against sequence number guessing and SYN flooding attacks. They supplement the role of the sequence numbers when the remote host initiates a TCP connection. Upon receiving a SYN segment, `TcpIp` transmits a SYN ACK. The SYN ACK includes the sequence numbers encoded with the connection state using two secret keys along with the time of creation, thus making the number harder to guess.

In order to stop processing potential fake SYNs, the feature also provides a time-out after the SYN queue is overflowed. In addition, a more flexible socket state management is provided. When the cache is full and a valid ACK segment is received while not having a matching entry saved, it can replace any older cache entry that is not established.

For information on how to configure the TCP SYN cookies feature, see [Task “Configuring SYN cookies”](#).

#### 4.7.2.1.2. Unpredictable sequence numbers

TCP unpredictable sequence numbers provide greater protection against sequence number guessing. They supplement the role of the sequence numbers when the local host initiates a TCP connection. Upon initiating a connection, `TcpIp` transmits a TCP SYN segment in which the sequence number is encoded with a secret key so that it is harder to guess.

For information on how to configure the unpredictable sequence numbers feature, see [Task “Configuring the unpredictable sequence numbers”](#).

#### 4.7.2.2. Storage of the IP address to NVM RAM

Each time an AUTOIP address is assigned, a new address should be randomly generated. In order to remember an assigned address during initial ECU start-up, permanent storage is required. Thus, in any subsequent ECU start-up, the address is not generated but read from the permanent storage.

For information on how to configure the storing to NVM RAM, see [Section 4.7.3.2, “Configuring the permanent IP address storage to NVM RAM”](#).

#### 4.7.2.3. IPv4 Duplicate Address Detection (ARP probing)

IPv4 Duplicate Address Detection (DAD) is a part of the Address Resolution Protocol (ARP). It is used to determine if an IPv4 address is unique in a network or already in use on some other interface. It is also called ARP probing.



The uniqueness is checked by transmitting several ARP probes to check if some other node uses the IP address. If there is no reply, the address is unique and can be used. Otherwise, there is a conflict and another address must be used.

The uniqueness of the address is checked before the address assignment. IPv4 Duplicate Address Detection (ARP probing) can be done for LINKLOCAL, LINKLOCAL\_DOIP, and DHCP assignments.

For information on how to configure the IPv4 ARP probing, see [Section 4.7.3.3, “Configuring the IPv4 Duplicate Address Detection \(ARP probing\)”](#).

#### 4.7.2.4. IPv6 Duplicate Address Detection

IPv6 Duplicate Address Detection (DAD) is a part of the Neighbor Discovery Protocol (NDP). It is used to determine if an IPv6 address is unique in a network or already in use on some other interface.

The uniqueness is checked by transmitting several NDP solicitations to check if some other node uses the IP address. If there is no reply, the address is unique and can be used. Otherwise, there is a conflict and another address must be used.

IPv6 DAD supports two modes of operation: optimistic and tentative. In an optimistic operation, `TcpIp` assigns and uses the address before checking for uniqueness. In a tentative operation, the uniqueness is checked before the assignment. IPv6 Duplicate Address Detection can be done for STATIC, LINKLOCAL, DHCP, and IPV6\_ROUTER assignments.

For information on how to configure the IPv6 DAD, see [Section 4.7.3.4, “Configuring the IPv6 Duplicate Address Detection”](#).

#### 4.7.2.5. Debug support in `Tcplp`

With the IPsec error logger, the `TcpIp` provides a module-specific mechanism that can help you with debugging a problem that occurs during configuration. For general debug support mechanisms provided in the ACG8 IP Stack, see [Section 4.3.2, “Debug support”](#).

The IPsec error logger is a means for detecting error conditions and calling a callout function with additional information, e.g. the local and remote IP address, or an error code. More specifically, the callout function is called together with an error code if particular error conditions are met.

The following table shows the error codes that are available in IKEv2 and IPsec.

Protocol	Error code	Value
IKEv2	TCPIP_IKEV2_E_OK	0U



Protocol	Error code	Value
	TCPIP_IKEV2_E_NOT_OK	201U
	TCPIP_IKEV2_E_NO_PROPOSAL_CHOSEN	203U
	TCPIP_IKEV2_E_INVALID_SYNTAX	204U
	TCPIP_IKEV2_E_AUTHENTICATION_FAILED	205U
	TCPIP_IKEV2_E_TS_UNACCEPTABLE	206U
	TCPIP_IKEV2_E_INVALID_KE_PAYLOAD	207U
	TCPIP_IKEV2_E_PAYLOAD_MISSING	210U
	TCPIP_IKEV2_E_SIG_HASH_NOT_SUPPORTED	211U
	TCPIP_IKEV2_E_UNSUPPORTED_CRITICAL_PAYLOAD	213U
	TCPIP_IKEV2_E_INVALID_MESSAGE_ID	214U
IPsec	TCPIP_MEAS_DROP_INVALID_POLICY	0x84U



### Configuring the IPsec error logger

#### Step 1

Add the container `TcpIpConfig/TcpIpIpSecConfig/TcpIpIpSecReportErrorHandler`.

#### Step 2

In `TcpIpIpSecReportErrorHandlerName`, enter the name of the callout function that is called if certain error conditions occur.

#### Step 3

In `TcpIpIpSecReportErrorHandlerFileName`, specify the header file that provides the function declaration of the callout function.

## 4.7.3. Configuring the TcpIp module

### 4.7.3.1. Configuring the software hardening features

- ▶ [Task “Configuring SYN cookies”](#)
- ▶ [Task “Configuring the unpredictable sequence numbers”](#)




### Configuring SYN cookies

#### Prerequisite:

- The `TcpIp`, `Csm`, `Crypto`, `StbM`, and `CryIf` modules are added and configured in EB tresos Studio. For configuration guidance, see the parameter descriptions in EB tresos Studio or in the `TcpIp` module references in [Chapter 5. “ACG8 IP Stack module references”](#).
- In `TcpIp`, the following is configured:
  - ▶ `TcpIpTcpEnabled` is enabled.
  - ▶ `TcpIpTcpSocketMax` AND/OR `TcpIpTcpV6SocketMax` are set to the desired number (minimum of 2 - listen and connection socket).
  - ▶ `TcpIpIpConfig/TcpIpTcpConfig` is enabled.
- In `Csm`, the following is configured:
  - ▶ The `Csm/CsmJob/CsmJob` list contains two separate jobs meant to be used for the SYN cookies only.
  - ▶ The `Csm/KeyJob/KeyJob` list contains two separate keys meant to be used for the SYN cookies only, which are linked to the job IDs.



The `Csm` parameters are used to generate a pseudo-random value using the `Csm_MacGenerate()` function and to periodically reset the keys using the `Csm_KeyGenerate()` function. The choice of the algorithm for creating the pseudo-random value is up to the integrator, because it is used only by the local host when generating and evaluating received SYN cookies. `TcpIpCustomCsmInterfaceHeaderFile` can be enabled to provide a header file to override the default `Csm` functions.

- The Crypto Stack must be initialized before `TcpIp`.

#### Step 1

Enable the parameter `TcpIp/TcpIpConfig/TcpIpTcpConfig/TcpIpTcpSynCookies`.

#### Step 2

In `TcpIpTcpSynCookiesKey1GenerateJobId`, reference the first job ID configured in `Csm`.

#### Step 3

In `TcpIpTcpSynCookiesKey2GenerateJobId`, reference the second job ID configured in `Csm`.

`TcpIpTcpSynCookiesKey2GenerateJobId` must be different from

`TcpIpTcpSynCookiesKey1GenerateJobId`. The keys referenced by both jobs must also be different.

#### Step 4

In `TcpIpTcpSynCookiesTimeResetKeys`, set the time in seconds after which new keys are generated.

The timer starts after the keys were used for the first time.

#### Step 5

In `TcpIpTcpSynCookiesAcceptAckOverflowTime`, set the time in seconds that must elapse before new SYN messages are accepted after an overflow.

#### Step 6

In `TcpIpTcpSynCookiesTimebaseRef`, reference a `StbMSynchronizedTimeBase` from `StbM`.



Configuring the unpredictable sequence numbers

Prerequisite:

- The `TcpIp`, `Csm`, `Crypto`, and `CryIf` modules are added and configured in EB tresos Studio. For configuration guidance, see the parameter descriptions in EB tresos Studio or in the `TcpIp` module references in [Chapter 5, “ACG8 IP Stack module references”](#).
- In `TcpIp`, the following is configured:
  - ▶ `TcpIpEnabled` is enabled.
  - ▶ `TcpIpIpConfig/TcpIpTcpConfig` is enabled.
- In `Csm`, the following is configured:
  - ▶ `Csm/CsmJob/CsmJob` list contains one job meant to be used for the unpredictable sequence numbers only.



- ▶ `Csm/KeyJob/KeyJob list` contains one key meant to be used for the unpredictable sequence numbers only, which is linked to the job ID.

The `Csm` parameters are used to generate a pseudo-random value using the `Csm_MacGenerate()` function and to periodically reset the keys using the `Csm_KeyGenerate()` function. The choice of the algorithm for creating the pseudo-random value is up to the integrator, because it is used only by the local host when generating initial sequence numbers. The secret key length should be at least 128 bit. The `TcpIp-CustomCsmInterfaceHeaderFile` can be enabled to provide a header file to override the default `Csm` functions.

- The Crypto Stack must be initialized before `TcpIp`.

#### Step 1

Enable the parameter `TcpIp/TcpIpConfig/TcpIpTcpConfig/TcpIpTcpUnpredictableSeqNumbers`.

#### Step 2

In `TcpIpTcpUnpredictableSeqNumbersKeyGenerateJobId`, reference the job ID configured in `Csm`.

#### Step 3

In `TcpIpTcpUnpredictableSeqNumbersKeyResetTime`, set the time in seconds after which a new key is generated. The timer starts after the key was used for the first time.



#### 4.7.3.2. Configuring the permanent IP address storage to NVM RAM

**TcpIp**

Name\*

General EB Published Informa TcpIpCtrl TcpIpDhcpServerConfi TcpIpIpConfig TcpIpLocalAddr TcpIpSocketOwnerConf TcpIpConfig TcpIpMemoryConfig TcpIpIpSecConfig »1

**TcpIpConfig**

Name\*

**TcpIpNvmBlock**

Name

TcpIpNvmBlockDescriptorRef

TcpIpNvmBlockSize (1 -> 65535)

**TcpIpPhysAddrConfig**

**TcpIpAddrAssignment**

Name

General

TcpIpAssignmentLifetime

TcpIpAssignmentMethod

TcpIpAssignmentPriority (1 -> 4)

TcpIpAssignmentTrigger

TcpIpUseSimpleDhcpClient



Storing the permanent IP address to NVM RAM

##### Prerequisite:

- The `TcpIp`, `NvM`, and `BswM` modules are added and configured in EB tresos Studio. For configuration guidance, see the parameter descriptions in EB tresos Studio or in the `TcpIp` module references in [Chapter 5, “ACG8 IP Stack module references”](#)
- In `TcpIp`, the following is configured:
  - ▶ `TcpIpIpV4Enabled` is enabled.
  - ▶ `TcpIpAutoIpEnabled` is enabled.
- In `NvM`, the following is configured:
  - ▶ The `NvM/NvMBlockDescriptor` list contains a block that is used for storing IP addresses.
  - ▶ `ref(NvM/NvMBlockDescriptor)/NvMSelectBlockForReadAll` is enabled.



- ▶ `ref (NvM/NvMBlockDescriptor) /NvMSelectBlockForWriteAll` is enabled.
- ▶ `ref (NvM/NvMBlockDescriptor) /NvMRamBlockDataAddress` is set to `Tcplp_Memory_NvM_Ip_Memory`.
- ▶ `ref (NvM/NvMBlockDescriptor) /NvMNvBlockLength` is set to `4 * number of addresses that need to be stored`.
- ▶ `NvM/NvMCommon/NvMUserHeader` list contains an entry `Tcplp.h`.

The `NvM` parameters are used to inform `NvM` to permanently store IP addresses at shutdown using `NvM_SetRamBlockStatus()` and to check if the address was correctly read from permanent storage to `TcpIp_Memory_NvM_Ip_Memory` using `NvM_GetErrorStatus()`.

#### Step 1

Enable the parameter `TcpIpConfig/TcpIpNvmBlock`.

#### Step 2

In `TcpIpNvmBlockDescriptorRef`, reference the block configured in `NvM`.

#### Step 3

In `TcpIpNvmBlockSize`, specify the size of `TcpIp_Memory_NvM_Ip_Memory` in units of four bytes.

#### Step 4

In `TcpIp/TcpIpConfig/TcpIpLocalAddr/*/TcpIpAddrAssignment/*/TcpIpAssignmentLifetime`, specify the address that needs to be permanently stored.

The number of addresses configured for permanent storage must be less or equal to `TcpIpNvmBlockSize`.

Note: Permanent storage can be used only for IPv4 AUTOIP LINKLOCAL and LINKLOCAL\_DOIP addresses.



#### 4.7.3.3. Configuring the IPv4 Duplicate Address Detection (ARP probing)

The screenshot shows the configuration interface for the TCP/IP stack. It includes sections for **TcpIpIpv4General** and **TcpIpDhcpConfig**.

**TcpIpIpv4General** settings:

- Name: TcpIpIpv4General
- TcpIpArpEnabled: Enabled (checked)
- TcpIpAutoIpEnabled: Enabled (checked)

**TcpIpDhcpConfig** settings:

- Name: TcpIpDhcpConfig
- General tab selected.
- TcpIpDhcpIpv4EntriesMax: 5
- TcpIpDhcpInitDelay: 1
- TcpIpIpv4DhcpAddrDefenseMechanism: TCPIP\_DEFEND\_ADDR
- TcpIpDhcpFQDNOptionEnabled: Enabled (checked)
- TcpIpDhcpIpv4DomainNameMaxSize: 11
- TcpIpDhcpConfigurableOptionsEnabled: Enabled (checked)
- TcpIpDhcpConfigurableOptionsEntriesMax: 7
- TcpIpDhcpConfigurableOptionsDataSizeMax: 23
- TcpIpDhcpArpProbingEnabled: Enabled (checked)
- TcpIpDhcpArpProbingType: PROBING\_DEFAULT



#### Configuring IPv4 Duplicate Address Detection (ARP probing)

Prerequisite:

- The `TcpIp` module is added and configured in EB tresos Studio. For configuration guidance, see the parameter descriptions in EB tresos Studio or in the `TcpIp` module references in [Chapter 5, “ACG8 IP Stack module references”](#).
- In `TcpIp`, the following is configured:
  - `TcpIpGeneral/TcpIpIpv4General/TcpIpIpv4Enabled` is enabled.
  - At least one UNICAST IPv4 address exists in the address list.
  - `TcpIpConfig/TcpIpIpConfig/TcpIpIpv4Config` is enabled.



- ▶ The `TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpArpConfig` list contains one entry.

#### Step 1

To use IPv4 Duplicate Address Detection (ARP probing) for all LINKLOCAL and/or LINKLOCAL\_DOIP addresses, enable the parameter `TcpIpGeneral/TcpIpIpV4General/TcpIpAutoIpEnabled`.

#### Step 2

Add one entry in the list `TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpAutoIpConfig`.

#### Step 3

Set at least one LINKLOCAL or LINKLOCAL\_DOIP address. The number of ARP probes and their timings are defined by predefined values that depend on the assignment method, i.e. LINKLOCAL or LINKLOCAL\_DOIP.

#### Step 4

To enable DHCP client support, enable the parameter `TcpIpGeneral/TcpIpIpV4General/TcpIpDhcpClientEnabled`. If enabled, the configuration is updated to use IPv4 Duplicate Address Detection (ARP probing) for all DHCP addresses.

#### Step 5

Add one entry in the list `TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpDhcpConfig`.

#### Step 6

Set at least one DHCP address.

#### Step 7

Enable the parameter `TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpDhcpConfig/TcpIpDhcpArpProbingEnabled` to enable DHCPv4 ARP probing (Duplicate Address Detection according to IETF RFC 2131 and RFC 5227).

#### Step 8

In `TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpDhcpConfig/TcpIpDhcpArpProbingType`, set the probing type for DHCP to either `PROBING_DEFAULT` or `PROBING_DOIP`.

The number of ARP probes and their timings are defined by predefined values that depend on `TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpDhcpConfig/TcpIpDhcpArpProbingType` and are the same as for LINKLOCAL and LINKLOCAL\_DOIP.



#### 4.7.3.4. Configuring the IPv6 Duplicate Address Detection

The screenshot shows two configuration panels in EB tresos Studio:

- TcpIpNdpSlaacConfig**:
  - Name: TcpIpNdpSlaacConfig
  - Configuration fields:
    - TcpIpNdpSlaacDadNumberOfTransmissions (0 -> 254): Value 2
    - TcpIpNdpSlaacDadRetransmissionDelay (0 -> 10): Value 1.0
    - TcpIpNdpSlaacDelayEnabled: Enabled (checkbox checked)
    - TcpIpNdpSlaacOptimisticDadEnabled: Enabled (checkbox checked)
- TcpIpDhcpV6Config**:
  - Name: TcpIpDhcpV6Config\_0
  - General Tab (selected):
    - Configuration fields:
      - TcpIpDhcpV6CnfDelayMax (0 -> 100): Value 1.0
      - TcpIpDhcpV6CnfDelayMin (0 -> 100): Value 0.0
      - TcpIpDhcpV6InfDelayMax (0 -> 100): Value 1.0
      - TcpIpDhcpV6InfDelayMin (0 -> 100): Value 0.0
      - TcpIpDhcpV6SolDelayMax (0 -> 100): Value 1.0
      - TcpIpDhcpV6SolDelayMin (0 -> 100): Value 0.0
      - TcpIpDhcpIpv6EntriesMax (0 -> 255): Value 1
      - TcpIpDhcpIpv6ServerDuidMaxSize (2 -> 128): Value 12
      - TcpIpDhcpIpv6DomainNameMaxSize (5 -> 255): Value 5
      - TcpIpDhcpV6FQDNOptionEnabled: Enabled (checkbox checked)
      - TcpIpDhcpV6ConfigurableOptionsEntriesMax (1 -> 255): Value 3
      - TcpIpDhcpV6ConfigurableOptionsDataSizeMax (1 -> 65535): Value 10
      - TcpIpDhcpV6SlaacDadEnabled: Enabled (checkbox checked)



#### Configuring IPv6 Duplicate Address Detection

##### Prerequisite:

- The `TcpIp` module is added and configured in EB tresos Studio. For configuration guidance, see the parameter descriptions in EB tresos Studio or in the `TcpIp` module references in [Chapter 5, “ACG8 IP Stack module references”](#).
- In `TcpIp`, the following is configured:
  - ▶ `TcpIpGeneral/TcpIpIpv6General/TcpIpIpv6Enabled` is enabled.
  - ▶ At least one UNICAST IPv6 address exists in address list.



- ▶ `TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config` is enabled.
- ▶ The `TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig` list contains one entry.

#### Step 1

To use IPv6 Duplicate Address Detection for all addresses, enable the parameter `TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpSlaacConfig`.

#### Step 2

In `TcpIpNdpSlaacDadNumberOfTransmissions`, set the number of Neighbor Solicitations that have to be unanswered in order to set an autoconfigured address to PREFERRED (usable) state.

#### Step 3

In `TcpIpNdpSlaacDadRetransmissionDelay`, set the maximum value for the address configuration delay(s).

#### Step 4

In `TcpIpNdpSlaacDelayEnabled`, define if the transmission of the first DAD Neighbor Solicitation is delayed by a random value from [0...MAX\_DAD\_DELAY].

#### Step 5

To enable optimistic DAD according to IETF RFC 4429 for all configured addresses, enable `TcpIpNdpSlaacOptimisticDadEnabled`.

#### Step 6

To enable DAD for all configured DHCPv6 addresses, enable the parameter `TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpDhcpV6Config/TcpIpDhcpV6SlaacDadEnabled`.

## 4.8. QoS Support user guide

### 4.8.1. Overview

The ACG8 IP Stack QoS Support user guide provides information about the Quality of Service (QoS) concept of the IP stack in the AUTOSAR context. Before you read this user guide, read the general concept of communication stacks in AUTOSAR that are described in the EB tresos AutoCore Generic documentation.

- ▶ [Section 4.8.2, “Background information”](#) describes the basic concept of QoS guarantees with regard to latency for time-sensitive Ethernet communication.
- ▶ [Section 4.8.3, “Configuring QoS Support”](#) provides instructions on how to configure the QoS Support features in the ACG8 IP Stack modules.



## 4.8.2. Background information

Distributed real-time systems require that operations within a single network node are executed in time. The timeliness of communication between different network nodes is equally important. There must be a guarantee with regard to the end-to-end response time from a stimulus that occurs at one network node to a response at another network node. A late delivery of the exchanged data at the receiving network node impacts the end-to-end response time. Therefore, the end-to-end communication latency via the network is important to achieve the desired end-to-end response time.

Distributed applications that are sensitive to changes in the temporal behavior are called *time-sensitive applications*. The network infrastructure together with the corresponding algorithms and protocols, which prevent such changes in the temporal behavior and which provide a guaranteed *quality of service* (QoS) with regard to latency and jitter limits are called *time-sensitive networking*.

The switched network topology of Ethernet provides very high flexibility. Ethernet switches can decouple simultaneous receptions by storing and forwarding one frame after the other. However, this mechanism affects the timing and requires appropriate buffer resources within the switch. Without further measures, all frames are treated equally. As a consequence, the transmission of a certain frame can either be delayed significantly by interfering traffic or, in case of heavy congestions, the transmission can even be dropped completely if the switch runs out of buffer resources. Congestions can occur when multiple nodes transmit several frames in a short period of time which results in a burst of transmissions.

QoS and transmission latency are particularly important for *data streams*. A data stream refers to a sequence of single frames which form a unidirectional data flow, e.g. audio, video, or control data. The data stream originates at a providing network node and runs to one or more consuming network nodes. The providing network node is called a talker. The consuming network node is called a listener.

To ensure the timely delivery of certain data streams, traffic must be separated into traffic classes. The assignment of frames to a traffic class can be based on the priority denoted by the Priority Code Point (PCP) field of the VLAN tag [1]. Network nodes and switches can handle these traffic classes differently, i.e. store related frames in dedicated frame buffer queues and process them individually. This allows to favor traffic of higher classes and to confine the effects of congestions only to the respective traffic class and all lower traffic classes. Designated portions of the available bandwidth within a given observation interval can be reserved upfront for time-sensitive data streams. This makes the communication predictable and allows to allocate adequate buffer resources.

Every network node and switch shapes the traffic of higher priority classes by introducing small gaps between frames. The gaps avoid bursts and allow the processing of traffic of lower traffic classes during these gaps. This way, the risk of frame losses is eliminated. Also, worst case latencies for the transmission of frames of a certain traffic class can be guaranteed. An example of such a traffic shaper is the *Credit Based Shaper* (CBS) defined by the IEEE Standard for Virtual Bridged LANs [1]. The configuration of traffic shapers is derived from the bandwidth reservations related to the respective port. A significant advantage of this approach in contrast to time-triggered networks is that unused bandwidth can still be utilized by lower-priority traffic.



The ACG8 IP Stack includes the following additional features for network nodes to provide QoS guarantees with regard to latency for time-sensitive communication:

**Prioritized time-sensitive frame transmission:** Prioritized sending and receiving of data in the modules of the ACG8 IP Stack, e.g. priority-based queue selection for sending and receiving data or extended APIs for handling the priority-based queue on the Priority Code Point (PCP) field of the VLAN tag [1].

**Traffic shaping:** Provision of a dispatch unit to perform traffic shaping at the following different levels to ensure that each talker operates in conformance with the shaper algorithms, i.e. CBS algorithm as specified in the IEEE Standard for Virtual Bridged LANs [1]. This applies to individual data streams as well as the overall communication pattern of egress ports.

- ▶ **SR class streams level:** SR (stream reservation) class streams consist of all frames that belong to the same traffic class, e.g. denoted by the Priority Code Point (PCP).
- ▶ **AVB stream level:** All frames that share the same destination are considered to be part of mutual AVB streams.
- ▶ **Control streams level:** Control streams describe the flow of data of a single source, e.g. a control application.

**Sharing of an AVB stream:** Provision of a dispatch unit to enable sharing of AVB streams among multiple control streams, see [Figure 4.3, “Sharing of an AVB stream”](#).

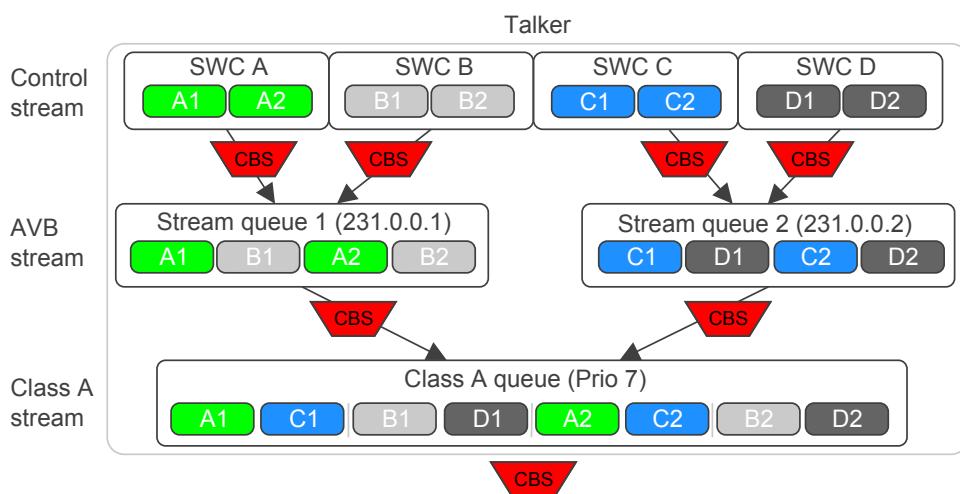


Figure 4.3. Sharing of an AVB stream

The bandwidth reservation for data streams is based on predefined observation intervals and not on the transmit patterns of individual talkers. To avoid over-reservations, the sharing of AVB streams allows to utilize the allocated bandwidth for multiple control streams. By supplying dedicated buffer resources to each individual control stream, it is ensured that every talker adheres to the traffic arrangements and cannot be disturbed by other talker applications. The hierarchical overview depicted in [Figure 4.3, “Sharing of an AVB stream”](#) does



not imply multiple sequential stages of buffer queues but depicts the arrangement of frames on the different levels of streams.

### 4.8.3. Configuring QoS Support

This section provides detailed information on how to configure the different EB tresos AutoCore Generic IP Stack modules as well as EB tresos AutoCore Generic RTE, EB tresos AutoCore MCAL Eth, and EB tresos AutoCore OS to provide QoS guarantees with regard to latency for time-sensitive communication.

#### 4.8.3.1. Configuring the Ethernet Driver Module (Eth)



##### Configuring the Ethernet Driver Module (Eth)

Prerequisite:

- A valid product license for ACM8 Eth QoS Support exists.

##### Step 1

To activate QoS Support, enable the `/Eth/EthGeneral/VendorSpecific/EthQoSSupport` parameter.

##### Step 2

Set up a dedicated reception queue for each traffic class in `EthCtrlConfigIngress`.

##### Step 3

Set up a dedicated transmission queue for each traffic class in `EthCtrlConfigEgress`.

##### Step 4

Configure the CBS parameter `idleSlope` (`EthCtrlConfigShaperIdleSlope`) according to your requirements with regard to the reserved bandwidth for the particular traffic class.



---

**TIP**



**Configuration example**

See the recommended configuration *Ethernet configuration with QoS* (`Eth_QoS`) or *Ethernet configuration with QoS and control stream* (`Eth_QoSCtrlStrm`).

---

**NOTE**



**Selection of the correct queue**

The selection of the correct queue takes place in the context of the `Eth_ProvideTxBuffer()` API and is determined by the `Priority` parameter.

#### 4.8.3.2. Configuring the Ethernet Interface Module (`EthIf`)



Configuring the Ethernet Interface Module (`EthIf`)

Step 1

To activate QoS Support, set the parameter `/EthIf/EthIfGeneral/EthIfSupportEthAPI` to ASR430 or higher.

Step 2

To configure an additional `EthIf_MainFunctionRx_<name>()` for prioritized reception of certain traffic, configure one instance of the `EthIfPhysCtrlRxMainFunctionPriorityProcessing` container for each additional `EthIf_MainFunctionRx_<name>()`. Hereby, `<name>` is the short name of the instance of the `EthIfPhysCtrlRxMainFunctionPriorityProcessing` container.

Step 3

In the `Os` and `Rte` configurations, map the execution of the `EthIf_MainFunctionRx_<name>()` instances to `OsTasks` with an adequate priority (see [Section 4.8.3.5, “Configuring the Operating System Module \(`Os`\)”](#) and [Section 4.8.3.6, “Configuring the Run-Time Environment Module \(`Rte`\)”](#)).

#### 4.8.3.3. Configuring the TCP/IP Module (`TcpIp`)



Configuring the TCP/IP Module (`TcpIp`)

Step 1

Define the default priority for a virtual network with the parameter `/TcpIp/TcpIpConfig/TcpIpCtrl/TcpIpIpFramePrioDefault`.

This assigns all traffic to a given Ethernet driver (`Eth`) queue (see [Section 4.8.3.1, “Configuring the Ethernet Driver Module \(`Eth`\)”](#)). This default priority is used if no specific priorities for a particular socket connec-



tion group are configured in the Socket Adaptor (SoAd) module (see [Section 4.8.3.4, “Configuring the Socket Adaptor Module \(SoAd\)”](#)).

#### 4.8.3.4. Configuring the Socket Adaptor Module (SoAd)

Although the transmission control protocol (TCP) provides features like reliable delivery of information in correct order, TCP is not well-suited for time-sensitive communication. TCP is optimized for accurate delivery rather than timely delivery and can incur relatively long delays (in the order of seconds) while waiting for out-of-order segments or re-transmissions of lost segments. This jeopardizes any guarantee for end-to-end transmission latencies.

The user datagram protocol (UDP) is a better choice for guaranteed end-to-end transmission latencies.



##### Configuring the Socket Adaptor Module (SoAd)

###### Step 1

Set the parameter `/SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol` of the involved socket connection groups to `SoAdSocketUdp`.

###### Step 2

Set the parameter `/SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest/SoAdTxUdpTriggerMode` of the involved `SoAdPduRoute` to `TRIGGER_ALWAYS`.

###### Step 3

To assign all frames of a `SoAdSocketConnectionGroup` to a given Ethernet driver (`Eth`) queue, enable the parameter `/SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketFramePriority` and set the priority appropriately.

#### 4.8.3.5. Configuring the Operating System Module (Os)

The `OsTask` provides the execution context (see [Section 4.8.3.6, “Configuring the Run-Time Environment Module \(Rte\)”](#)) for the `EthIf_MainFunctionRx_<name>()` instances. Use the `/Os/OsTask/OsTaskPriority` parameter to configure the prioritized processing of Ethernet frames that carry time-sensitive data.

**NOTE****Impact of a non-preemptible OsTask**

Any lower priority OsTask with /Os/OsTask/OsTaskSchedule configured to NON (i.e. non-preemptible OsTask) may still influence the execution of a higher priority OsTask and thus interfere with the timely execution of the EthIf\_MainFunctionRx\_<name>() instances mapped to the higher priority OsTask.

#### 4.8.3.6. Configuring the Run-Time Environment Module (Rte)



##### Configuring the Run-Time Environment Module (Rte)

**Step 1**

To facilitate the execution of the EthIf\_MainFunctionRx\_<name>() instances in the context of the proper OsTask, map the corresponding BSW timing events (TimingEvent\_MainFunctionRx\_<name>) to the desired OsTask in the RTE editor. Thereby ensure that you map the BSW timing events of EthIf\_MainFunctionRx\_<name>() instances that are intended for servicing higher priority frames to OsTasks with a higher priority.

**NOTE****Important integration requirement**

In both the Rte and Os configurations, make sure that the different EthIf\_MainFunctionRx\_<name>() instances and the EthIf\_MainFunctionRx() do not preempt each other.

If multiple EthIf\_MainFunctionRx\_<name>() instances are mapped to the same OsTask, ensure that the EthIf\_MainFunctionRx\_<name>() instances intended for servicing higher priority frames are placed before EthIf\_MainFunctionRx\_<name>() instances intended for servicing lower priority frames.



# 5. ACG8 IP Stack module references

## 5.1. Overview

This chapter provides module references for the ACG8 IP Stack 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 IP Stack 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. DoIP

### 5.2.1. 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="#">DoIPConfigSet</a>	1..1	This container contains the configuration parameters and sub containers of the AUTOSAR DoIP module.
<a href="#">DoIPGeneral</a>	1..1	This container specifies the general configuration parameters of the DoIP module.
<a href="#">DoIPDefensiveProgramming</a>	1..1	<b>Label:</b> Defensive Programming Options Parameters for defensive programming
<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

Parameter Name	<b>IMPLEMENTATION_CONFIG_VARIANT</b>
Label	Config Variant
Multiplicity	1..1
Type	ENUMERATION
Default value	VariantPostBuild
Range	VariantPostBuild



Configuration class	VariantPostBuild:	VariantPostBuild
---------------------	-------------------	------------------

### 5.2.1.1. CommonPublishedInformation

Parameters included	
Parameter name	Multiplicity
<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	<b>ArMajorVersion</b>
Label	AUTOSAR Major Version
Description	Major version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	4
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	<b>ArMinorVersion</b>
Label	AUTOSAR Minor Version
Description	Minor version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	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>	3
<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>	1
<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>	22
<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>	173
<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



### 5.2.1.2. DolPConfigSet

Containers included		
Container name	Multiplicity	Description
<a href="#">DolPChannel</a>	1..65536	Configuration of one DolPChannel.
<a href="#">DolPCustomChannel</a>	1..254	Configuration of one DolPCustomChannel.
<a href="#">DolPConnections</a>	1..1	Container contains all lower layer connection specific information, i.e. the single PDU References and Handle IDs to the SoAd.
<a href="#">DolPRoutingActivation</a>	0..255	This container describes the routing activation possibilities by representing for each container a possible routing activation request message to the DolP entity and the according references to the activated diagnostic messages.
<a href="#">DolPTester</a>	1..255	This container describes the properties of the possible connectable Tester for the DolP entity.

Parameters included	
Parameter name	Multiplicity
<a href="#">DolPEid</a>	0..1
<a href="#">DolPGid</a>	0..1
<a href="#">DolPLLogicalAddress</a>	1..1

Parameter Name	DolPEid
Description	Configured EID (Entity ID of) for vehicle identification/vehicle announcement. Only necessary if DolPUseMacAddressForIdentification is set to FALSE.
Multiplicity	0..1
Type	INTEGER
Range	<=281474976710655 >=0
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

Parameter Name	DolPGid
Description	Configured GID (Group ID of) for vehicle identification/vehicle announcement.
Multiplicity	0..1
Type	INTEGER



<b>Range</b>	<=281474976710655 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>DolPLogicalAddress</b>
<b>Description</b>	Describes the logical address of the DolP entity, i.e. the LA that will route diagnostic requests to the Dcm of the DolP entity.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Range</b>	<=65535 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.2.1.3. DolPChannel

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">DolPPduRRxPdu</a>	1..1	This container contains the Rx PDUs to connect with the Rx PDUs of the PduR.
<a href="#">DolPPduRTxPdu</a>	0..1	This container contains the Tx PDUs to connect with the Tx PDUs of the PduR. If the parameter is not configured the channel is for functional addressing.

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

<b>Parameter Name</b>	<b>DolPChannelSARef</b>
<b>Description</b>	Reference to the DolPTester.
<b>Multiplicity</b>	1..1



Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DoIPChannelTARef	
Description	Reference to the target address.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DoIPRoutingActivationType	
Description	Type of routing activation: DOIP_ROUTING_ACTIVATION_MSG - Activation with routing activation message. DOIP_ROUTING_ACTIVATION_AUTOMATIC - No need to send routing activation message prior receiving diagnostic message.	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	DOIP_ROUTING_ACTIVATION_MSG	
Range	DOIP_ROUTING_ACTIVATION_MSG DOIP_ROUTING_ACTIVATION_AUTOMATIC	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

#### 5.2.1.4. DoIPduRRxPdu

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

Parameter Name	DoIPduRRxPduld
Description	The DoIPduRRxPduld is required by the API call DoIP_CancelReceive.



<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Range</b>	<=65535 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	DolPPduRRxPduRef
<b>Description</b>	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.2.1.5. DolPPduRTxPdu

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

<b>Parameter Name</b>	DolPPduRTxPduld
<b>Description</b>	The DolPPduRTxPduld is required by DolP_TpTransmit and DolP_CancelTransmit.
<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>	DolPPduRTxPduRef
<b>Description</b>	Reference to the "global" PDU structure to allow harmonization of handle IDs in the COM-Stack.
<b>Multiplicity</b>	1..1



Type	REFERENCE	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.6. DolPCustomChannel

Containers included		
Container name	Multiplicity	Description
<a href="#">DolPPduRRxPdu</a>	1..1	This container contains the Rx PDUs to connect with the Rx PDUs of the PduR.
<a href="#">DolPPduRTxPdu</a>	1..1	This container contains the Tx PDUs to connect with the Tx PDUs of the PduR.

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

Parameter Name	DolPTcpConnectionRef
Description	Reference to Tcp connection used for transmission of custom diagnostic messages.
Multiplicity	1..1
Type	REFERENCE
Configuration class	<b>VariantPostBuild:</b> VariantPostBuild
Origin	Elektrobit Automotive GmbH

### 5.2.1.7. DolPPduRRxPdu

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

Parameter Name	DolPPduRRxPduld
	Values should be consecutive after DolPPduRRxPdulds from DolPChannel



<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Range</b>	<p>&lt;=65535</p> <p>&gt;=num:i(count(as:modconf('DoIP')[1]/DoIPConfigSet/*[1]/DoIPChannel/*/DoIP-PduRRxPdu/DoIPPduRRxPduld))</p>
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	DolPPduRTxPduRef
<b>Description</b>	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.2.1.8. DolPPduRTxPdu

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">DolPPduRTxPduld</a>	1..1
<a href="#">DolPPduRTxPduRef</a>	1..1
<b>Parameter Name</b>	<b>DolPPduRTxPduld</b>
<b>Description</b>	Values should be consecutive after DolPPduRTxPdulds from DolPChannel
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Range</b>	<p>&lt;=65535</p> <p>&gt;=num:i(count(as:modconf('DoIP')[1]/DoIPConfigSet/*[1]/DoIPChannel/*/DoIP-PduRTxPdu/DoIPPduRTxPduld))</p>
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH
<b>Parameter Name</b>	<b>DolPPduRTxPduRef</b>



<b>Description</b>	Reference to the "global" PDU structure to allow harmonization of handle IDs in the COM-Stack.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.2.1.9. DolPConnections

Containers included		
Container name	Multiplicity	Description
<a href="#">DolPTargetAddress</a>	1..65535	This container describes a possible TargetAddress that is supported by DolP.
<a href="#">DolPTcpConnection</a>	2..255	This container describes a Tcp connection to the lower layer SoAd module.
<a href="#">DolPUdpConnection</a>	1..255	This Container describes a UDP connection to the lower layer SoAd module.
<a href="#">DolPUdpVehicleAnnounce-ment</a>	1..255	This container provides PDUs for UDP multicast vehicle announcements.

### 5.2.1.10. DolPTargetAddress

Parameters included		
Parameter name	Multiplicity	
<a href="#">DolPTargetAddressValue</a>	1..1	
<b>Parameter Name</b>	<b>DolPTargetAddressValue</b>	
<b>Description</b>	Valid Target Address of a DolP target address.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Range</b>	<=65535 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	



### 5.2.1.11. DolPTcpConnection

Containers included		
Container name	Multiplicity	Description
<a href="#">DolPSoAdRxPdu</a>	1..1	This container contains the Rx PDUs received by DolP.
<a href="#">DolPSoAdTxPdu</a>	1..1	This container describes the Tx PDU sent via the SoAd.

Parameters included	
Parameter name	Multiplicity
<a href="#">DolPTcpConnectionSecurityRequired</a>	1..1
<a href="#">DolPRequestAddressAssignment</a>	1..1
<a href="#">DolPUpdateDhcpHostNameOption</a>	1..1

Parameter Name	<b>DolPTcpConnectionSecurityRequired</b>	
Description	Indicates if the associated TCP socket uses a secure connection (e.g. TLS).	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	<a href="#">VariantPostBuild:</a>	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>DolPRequestAddressAssignment</b>	
Description	When set DolP module will request IP address assignment by calling SoAd_RequestIpAddrAssignment() for the TcplpLocalAddr related to this connection.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	<a href="#">VariantPostBuild:</a>	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>DolPUpdateDhcpHostNameOption</b>	
Description	When set DolP module will update DHCP host name option related to this connection.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.2.1.12. DoIPSoAdRxPdu

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

<b>Parameter Name</b>	<b>DoIPSoAdRxPduld</b>
<b>Description</b>	The DoIPSoAdRxPduld is required by the API call DoIP_SoAdTpRxIndication to receive I-PDUs from the SoAd.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Range</b>	<=65535 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>DoIPSoAdRxPduRef</b>
<b>Description</b>	Reference to the "global" PDU structure to allow harmonization of handle IDs in the COM-Stack.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	AUTOSAR_ECUC

### 5.2.1.13. DoIPSoAdTxPdu

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



#### Parameters included

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

<b>Parameter Name</b>	<b>DoIPSoAdTxPduld</b>
<b>Description</b>	The DoIPSoAdTxPduld is required by the API call DoIP_SoAdTpTxConfirmation that is called by the SoAd to confirm that the IPdu has been transmitted successfully.
<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>DoIPSoAdTxPduRef</b>
<b>Description</b>	Reference to the "global" PDU structure to allow harmonization of handle IDs in the COM-Stack.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.2.1.14. DoIPUdpConnection

##### Containers included

Container name	Multiplicity	Description
<a href="#">DoIPSoAdRxPdu</a>	1..1	This container contains the Rx PDUs received by DoIP.
<a href="#">DoIPSoAdTxPdu</a>	1..1	This container describes the Tx PDU sent via the SoAd.

##### Parameters included

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

<b>Parameter Name</b>	<b>DoIPRequestAddressAssignment</b>
<b>Description</b>	When set DoIP module will request IP address assignment by calling SoAd_RequestIpAddrAssignment() for the TcplpLocalAddr related to this connection.
<b>Multiplicity</b>	1..1



Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	DolPUpdateDhcpHostNameOption
Description	When set DolP module will update DHCP host name option related to this connection.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

### 5.2.1.15. DolPSoAdRxPdu

Parameters included	
Parameter name	Multiplicity
<a href="#">DolPSoAdRxPduld</a>	1..1
<a href="#">DolPSoAdRxPduRef</a>	1..1
Parameter Name	DolPSoAdRxPduld
Description	The DolPSoAdRxPduld is required by the API call DolP_SoAdTpRxIndication to receive I-PDUs from the SoAd.
Multiplicity	1..1
Type	INTEGER
Range	<=65535 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC
Parameter Name	DolPSoAdRxPduRef
Description	Reference to the "global" PDU structure to allow harmonization of handle IDs in the COM-Stack.



<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.2.1.16. DolPSoAdTxPdu

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

<b>Parameter Name</b>	<b>DolPSoAdTxPduld</b>
<b>Description</b>	The DolPSoAdTxPduld is required by the API call DolP_SoAdTpTxConfirmation that is called by the SoAd to confirm that the I-PDU has been transmitted successfully.
<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>DolPSoAdTxPduRef</b>
<b>Description</b>	Reference to the "global" PDU structure to allow harmonization of handle IDs in the COM-Stack.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.2.1.17. DolPUdpVehicleAnnouncement

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>



#### Containers included

<a href="#">DoIPSoAdTxPdu</a>	1..1	This container describes the Tx PDU sent via the SoAd.
-------------------------------	------	--

#### Parameters included

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

<b>Parameter Name</b>	<b>DoIPRequestAddressAssignment</b>	
<b>Description</b>	When set DoIP module will request IP address assignment by calling SoAd_RequestIpAddrAssignment() for the TcplpLocalAddr related to this connection.	
<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>DoIPUpdateDhcpHostNameOption</b>	
<b>Description</b>	When set DoIP module will update DHCP host name option related to this connection (starting with 'DoIP-').	
<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	

#### 5.2.1.18. DoIPSoAdTxPdu

##### Parameters included

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

<b>Parameter Name</b>	<b>DoIPSoAdTxPduld</b>
-----------------------	------------------------



<b>Description</b>	The DolPSoAdTxPduld is required by the API call DolP_SoAdTpTxConfirmation that is called by the SoAd to confirm that the IPdu has been transmitted successfully.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Range</b>	<=65535 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>DolPSoAdTxPduRef</b>	
<b>Description</b>	Reference to the "global" PDU structure to allow harmonization of handle IDs in the COM-Stack.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild	
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.2.1.19. DolPRoutingActivation

Containers included		
Container name	Multiplicity	Description
<a href="#">DolPRoutingActivationAuthenticationCallback</a>	0..1	Container describes the Callback function to call on a Routing Activation Request for Authentication. If this container is configured but the DolPRoutingActivationAuthenticationFunc parameter is not present, the DolP module will use an RPort of ServiceInterface <RoutingActivation>_RoutingActivation with the name "CB<RoutingActivation>RoutingActivation". <RoutingActivation> is the ShortName of the DolPRoutingActivation container.
<a href="#">DolPRoutingActivationConfirmationCallback</a>	0..1	Container describes the Callback function to call on a Routing Activation Request for Confirmation. If this container is configured but the DolPRoutingActivationConfirmationFunc parameter is not present the DolP module will use an RPort of ServiceInterface <RoutingActivation>_RoutingActivation with the name "CB<RoutingActivation>RoutingActivation".



#### Containers included

		<RoutingActivation> is the ShortName of the DoIPRoutingActivation container.
--	--	--

#### Parameters included

Parameter name	Multiplicity
<a href="#">DoIPRoutingActivationNumber</a>	1..1
<a href="#">DoIPRoutingActivationSecurityRequired</a>	1..1
<a href="#">DoIPTargetAddressRef</a>	0..65535

Parameter Name	<b>DoIPRoutingActivationNumber</b>	
Description	Identifies the Routing activation Number which is received for a DoIP routing activation request message.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=255 >=0	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>DoIPRoutingActivationSecurityRequired</b>	
Description	Indicates if a routing activation requires a secure TCP connection.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>DoIPTargetAddressRef</b>	
Description	Reference to all DoIPTargetAddress which are activated on this routing activation.	
Multiplicity	0..65535	
Type	REFERENCE	
Configuration class	<b>PostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	



### 5.2.1.20. DolPRoutingActivationAuthenticationCallback

Parameters included	
Parameter name	Multiplicity
<a href="#">DolPRoutingActivationAuthenticationFunc</a>	0..1
<a href="#">DolPRoutingActivationAuthenticationReqLength</a>	1..1
<a href="#">DolPRoutingActivationAuthenticationResLength</a>	1..1

Parameter Name	DolPRoutingActivationAuthenticationFunc
Description	Direct C Callback function to trigger the authentication function for routing activation. If the DolPRoutingActivationAuthenticationFunc parameter is present, the DolP module will not use an RPort of ServiceInterface <RoutingActivation>_RoutingActivation but call the configured function.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPRoutingActivationAuthenticationReqLength
Description	Describes the amount of bytes used to handle to the authentication function on routing activation. If 0 is configured as length the parameter AuthenticationReq-Data will not be handled to the API.
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<=4 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPRoutingActivationAuthenticationResLength
Description	Describes the amount of bytes used to read by the authentication function on routing activation. If 0 is configured as length the parameter AuthenticationRes-Data will not be fetched via the API.
Multiplicity	1..1
Type	INTEGER



<b>Default value</b>	0
<b>Range</b>	<=4 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.2.1.21. DolPRoutingActivationConfirmationCallback

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">DolPRoutingActivationConfirmationFunc</a>	0..1
<a href="#">DolPRoutingActivationConfirmationReqLength</a>	1..1
<a href="#">DolPRoutingActivationConfirmationResLength</a>	1..1

<b>Parameter Name</b>	<b>DolPRoutingActivationConfirmationFunc</b>
<b>Description</b>	Direct C Callback function to trigger the confirmation function for routing activation. If the DolPRoutingActivationConfirmationFunc parameter is present the DolP module will not use an RPort of ServiceInterface <RoutingActivation>_RoutingActivation but call the configured function.
<b>Multiplicity</b>	0..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>DolPRoutingActivationConfirmationReqLength</b>
<b>Description</b>	Describes the amount of bytes used to handle to the confirmation function on routing activation. If 0 is configured as length the parameter ConfirmedReqData will not be handled to the API.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Range</b>	<=4 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild



<b>Origin</b>	AUTOSAR_ECUC
<b>Parameter Name</b>	<b>DolPRoutingActivationConfirmationResLength</b>
<b>Description</b>	Describes the amount of bytes used to read by the confirmation function on routing activation. If 0 is configured as length the parameter ConfirmedResData will not be fetched via the API.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Range</b>	<=4 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.2.1.22. DolPTester

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">DolPNumByteDiagAckNack</a>	1..1
<a href="#">DolPTesterSA</a>	1..1
<a href="#">DolPRoutingActivationRef</a>	0..255
<b>Parameter Name</b>	<b>DolPNumByteDiagAckNack</b>
<b>Description</b>	Specifies the number of original Diagnostic request bytes the DolP entity responses on a NACK of a diagnostic response message to the Tester.  The maximum size is limited by /DolPGeneral/DolPMaxNumByteDiagAckNack.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Range</b>	<=4294967295 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC
<b>Parameter Name</b>	<b>DolPTesterSA</b>



<b>Description</b>	Source Address of the Tester sent via routing activation or diagnostic message.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Range</b>	<=65535 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>DolPRoutingActivationRef</b>	
<b>Description</b>	Reference to a DolPRoutingActivation describing the possible routing activations of the DolPTester.	
<b>Multiplicity</b>	0..255	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.23. DolPGeneral

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">DolPGetGid</a>	0..1	This container describes the Callback function to get the GID.
<a href="#">DolPPowerModeCallback</a>	1..1	This container describes the Callback function to get the Power Mode. This container shall always be present.
<a href="#">DolPTriggerGidSynchronization</a>	0..1	This container describes the Callback function to trigger the GID synchronisation.  <b>Note: Feature not supported.</b>
<a href="#">DolPDynamicGIDMasterSelectionCallback</a>	0..1	This container describes the Callback function to obtain DolP entity GID master/slave status.  If this container is not configured no Callback function will be used and Dynamic GID Synchronization feature will be disabled.
<a href="#">DolPVIDRequestReceived-Callback</a>	0..1	This container provides the functionality to call a user defined function.

**Containers included**

		If this container is not configured this functionality is disabled.
<a href="#">DoIPGetVIN</a>	0..1	This container describes the Callback function to get the VIN.  If this function is not configured DoIP module uses the configured DoIPVinInvalidityPattern.
<a href="#">DoIPTcpSoConModeChgCallback</a>	0..1	This container provides the functionality to call a user defined function.  If this container is not configured this functionality is disabled.

**Parameters included**

Parameter name	Multiplicity
<a href="#">DoIPAliveCheckResponseTimeout</a>	1..1
<a href="#">DoIPDevelopmentErrorDetect</a>	1..1
<a href="#">DoIPDhcpOptionVinUse</a>	1..1
<a href="#">DoIPEntityStatusMaxByteFieldUse</a>	1..1
<a href="#">DoIPGIDInvalidityPattern</a>	1..1
<a href="#">DoIPGeneralInactivityTime</a>	1..1
<a href="#">DoIPHeaderFileInclusion</a>	0..n
<a href="#">DoIPHostNameSizeMax</a>	1..1
<a href="#">DoIPInitialInactivityTime</a>	1..1
<a href="#">DoIPInitialVehicleAnnouncementTime</a>	1..1
<a href="#">DoIPMainFunctionPeriod</a>	1..1
<a href="#">DoIPMaxRequestBytes</a>	1..1
<a href="#">DoIPMaxTesterConnections</a>	1..1
<a href="#">DoIPMaxUDPSocketConnections</a>	1..1
<a href="#">DoIPMaxUDPRequestPerMessage</a>	1..1
<a href="#">DoIPNodeType</a>	1..1
<a href="#">DoIPUseEIDasGID</a>	1..1
<a href="#">DoIPUseMacAddressForIdentification</a>	1..1
<a href="#">DoIPUseVehicleIdentificationSyncStatus</a>	1..1
<a href="#">DoIPVehicleAnnouncementInterval</a>	1..1
<a href="#">DoIPVehicleAnnouncementRepetition</a>	1..1
<a href="#">DoIPVersionInfoApi</a>	1..1



#### Parameters included

<a href="#">DoIPVinGidMaster</a>	1..1
<a href="#">DoIPVinInvalidityPattern</a>	1..1
<a href="#">DoIPEnableMainFunctionTx</a>	1..1
<a href="#">DoIPMaxChannels</a>	1..1
<a href="#">DoIPMaxNumByteDiagAckNack</a>	1..1
<a href="#">DoIPCustomPayloadTypeEnabled</a>	1..1
<a href="#">DoIPGetAndResetMeasurementDataApi</a>	1..1
<a href="#">DoIPDhcpHostNamePrefix</a>	1..1
<a href="#">DoIPEnableTcpClosureWithFIN</a>	1..1
<a href="#">DoIPResponseBeforeRoutingActivation</a>	1..1
<a href="#">DoIPMaxVehicleAnnouncementCon</a>	1..1
<a href="#">DoIPRteUsage</a>	1..1
<a href="#">DoIPRelocatablePbcfgEnable</a>	1..1
<a href="#">DoIPRoutingActivationCallbackList</a>	0..255

#### Parameter Name

#### DoIPAliveCheckResponseTimeout

<b>Description</b>	Timeout in [s] for waiting for a response to an Alive Check request before the connection is considered to be disconnected. Represents parameter T_TCP_-AliveCheck of ISO 13400-2:2012.
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.5
<b>Range</b>	<=Infinity >=0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### Parameter Name

#### DoIPDevelopmentErrorDetect

<b>Description</b>	Pre-processor switch for enabling development error detection support.
<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>	DolPDhcpOptionVinUse
<b>Description</b>	If DolPDhcpOptionVinUse is set to true the DolP module will add the VIN to the Dhcp host name if no valid Dhcp host name is already set.
<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>	DolPEntityStatusMaxByteFieldUse
<b>Description</b>	This parameter is used to distinguish the optional support of the Max data size element of a diagnostic entity status response.
<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>	DolPGIDInvalidityPattern
<b>Description</b>	Specifies the Byte pattern that is used for response messages if no valid GID could be retrieved. Only the value '0' or '255' is allowed.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Range</b>	0 255
<b>Configuration class</b>	VariantPostBuild: VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	DolPGeneralInactivityTime
<b>Description</b>	Timeout in [s] for maximum inactivity of a TCP socket connection before the DolP module will close the according socket connection. Represents parameter T_TCP_General_Inactivity of ISO 13400-2:2012.
<b>Multiplicity</b>	1..1



<b>Type</b>	FLOAT
<b>Default value</b>	1.0
<b>Range</b>	<=Infinity >=0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>DolPHeaderFileInclusion</b>
<b>Description</b>	Name of the header file(s) to be included by the DolP module containing the used C-callback declarations.
<b>Multiplicity</b>	0..n
<b>Type</b>	STRING
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>DolPHostNameSizeMax</b>
<b>Description</b>	Maximum Size of the DHCP HostName in ASCII. This parameter is necessary to reserve the correct amount of bytes for working with the DHCP HostName option. Minimum range is 5 because Dhcp Host Name should be at least "DolP-" on any configuration.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	5
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>DolPInitialInactivityTime</b>
<b>Description</b>	Timeout in [s] used for initial inactivity of a connected TCP socket connection directly after socket connection. Represents parameter T_TCP_Initial_Inactivity of ISO 13400-2:2012.
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	1.0
<b>Range</b>	<=Infinity



	>=0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>DolPInitialVehicleAnnouncementTime</b>	
<b>Description</b>	Time to wait in [s] for sending first vehicle announcement message after IP address assignment. Represents parameter A_DolP_Announce_Wait of ISO 13400-2:2012.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.005	
<b>Range</b>	<=Infinity >=0.0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>DolPMainFunctionPeriod</b>	
<b>Description</b>	Determines the frequency at which the DolP_MainFunction() is called in [s].	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.001	
<b>Range</b>	<=Infinity >=0.0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>DolPMaxRequestBytes</b>	
<b>Description</b>	Specifies the maximum allowed bytes of a DolP message request without the DolP header.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	10	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	



<b>Parameter Name</b>	<b>DolPMaxTesterConnections</b>	
<b>Description</b>	Maximum amount of tester connections that shall be maintained at one time before alive check is performed.	
<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>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>DolPMaxUDPSocketConnections</b>	
<b>Description</b>	Maximum number of UDP socket connections.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>DolPMaxUDPRequestPerMessage</b>	
<b>Description</b>	This parameter captures the maximum amount of UDP Requests necessary to handle parallel within a single UDP connection.  <b>Note:</b> This configuration parameter is not used.	
<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>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>DolPNodeType</b>	
<b>Description</b>	Describes the Type of the DolP node.	
<b>Multiplicity</b>	1..1	



<b>Type</b>	ENUMERATION	
<b>Default value</b>	DOIP_NODE	
<b>Range</b>	DOIP_GATEWAY DOIP_NODE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>DolPUseEIDasGID</b>	
<b>Description</b>	Specifies if the DolP entity shall use its EID if it is the Master for vehicle identification/gid on the vehicle identification/vehicle announcement.	
<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>DolPUseMacAddressForIdentification</b>	
<b>Description</b>	Provides the information if a configured EID at vehicle identification response/vehicle announcement is used or the MAC address.	
<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>DolPUseVehicleIdentificationSyncStatus</b>	
<b>Description</b>	Defines if the optional VIN/GID synchronization status is used additionally in the vehicle identification/announcement.	
<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>DolPVehicleAnnouncementInterval</b>	
-----------------------	--	--



<b>Description</b>	Time to wait in [s] for sending subsequent vehicle announcement messages. Represents parameter A_DoIP_Announce_Interval of ISO 13400-2:2012.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.005	
<b>Range</b>	<=Infinity >=0.0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>DolPVehicleAnnouncementRepetition</b>	
<b>Description</b>	Amount of repetitions of the vehicle announcement message on IP address assignment. Represents parameter A_DoIP_Announce_Num of ISO 13400-2:2012.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	3	
<b>Range</b>	<=255 >=1	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>DolPVersionInfoApi</b>	
<b>Description</b>	Activates the DoIP_GetVersionInfo() API.	
<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>DolPVinGidMaster</b>	
<b>Description</b>	Specifies if the DoIP entity is the Vehicle Identification Master for the GID (Group ID).  <b>Note: Feature not supported.</b>	
<b>Multiplicity</b>	1..1	



Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPVinInvalidityPattern
Description	Specifies the Byte pattern that is used for response messages if no valid VIN could be retrieved.
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<=255 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPEnableMainFunctionTx
Description	Pre-processor switch for enabling calling the tx part of the main function separately.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	DolPMaxChannels
Description	Specifies the maximum number of configured channels (including custom and proxy channels if enabled).
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<=65535 >=1
Configuration class	VariantPostBuild: VariantPostBuild



<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

<b>Parameter Name</b>	<b>DoIPMaxNumByteDiagAckNack</b>
<b>Description</b>	Specifies maximum number of bytes of the original Diagnostic message that will be copied into the ACK/NACK response message to the Tester.  If this parameter is set to 0, the feature of sending a part of previous diagnostic message in acknowledge message is disabled. <b>This results in reduced RAM and ROM consumption.</b>
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	128
<b>Range</b>	<=65535 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>DoIPCustomPayloadTypeEnabled</b>
<b>Description</b>	Enables custom diagnostic message payload type feature - transmitting diagnostic messages with reserved payload types in the range 0xF000..0xFFFF for manufacturer specific use.  The feature implements IF-TP bridging between PduR and SoAd - PduR communicates over IF, and SoAd over TP Api.
<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>DoIPGetAndResetMeasurementDataApi</b>
<b>Description</b>	Pre-processor switch for enabling DoIP_GetAndResetMeasurementData API used for reading out and resetting counter for dropped messages due to generic header errors and counter for dropped diagnostic messages.
<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>DolPDhcpHostNamePrefix</b>	
<b>Description</b>	If parameter DolPDhcpHostNamePrefix is not an empty string it will be added to the DHCP host name after DolP- and prior to VIN when DolPDhcpOptionVinUse parameter is enabled. According to SWS this parameter shall be filled with "VIN", but it can also be filled with vendor specific value.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	STRING	
<b>Default value</b>		
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>DolPEnableTcpClosureWithFIN</b>	
<b>Description</b>	Enables Tcp connection closure with <b>FIN</b> flag set when performed together with NACK transmission. In this case <code>SoAd_CloseSoCon()</code> will be called with Abort parameter set to FALSE. In other cases when DolP needs to close Tcp connection, e.g. due to Inactivity or Alive check timeout, Abort parameter will be set to TRUE. This will perform a closure with <b>RST</b> flag set.  When this parameter is disabled, <code>SoAd_CloseSoCon()</code> will always be called with Abort set to TRUE.	
<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>DolPResponseBeforeRoutingActivation</b>	
<b>Description</b>	Specifies if diagnostic negative acknowledge messages will be sent prior to Routing activation:  DOIP_ENABLE_ALL_DIAG_NACK - Diag Nack messages will always be sent. DOIP_DISABLE_DIAG_NACK_0x06 - Diag Nack code 0x06 will not be sent before Routing Activation.	
<b>Multiplicity</b>	1..1	



<b>Type</b>	ENUMERATION	
<b>Default value</b>	DOIP_ENABLE_ALL_DIAG_NACK	
<b>Range</b>	DOIP_ENABLE_ALL_DIAG_NACK DOIP_DISABLE_DIAG_NACK_0x06	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>DolPMaxVehicleAnnouncementCon</b>	
<b>Description</b>	Specifies maximum number of vehicle announcement connections that can be configured.	
<b>Multiplicity</b>	1..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>DolPRteUsage</b>	
<b>Description</b>	Enables RTE Usage.  If enabled, the DolP will generate an SWCD and supply the specified software component interfaces.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>PreCompile:</b>	
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>DolPRelocatablePbcfgEnable</b>	
<b>Description</b>	Enables/disable support for relocatable postbuild configuration.  ▶ True: Postbuild configuration relocatable in memory. ▶ False: Postbuild configuration not relocatable in memory.	
<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>DolPRoutingActivationCallbackList</b>
<b>Description</b>	<p>List of user defined callback functions to be called after routing activation was successfully performed on a TCP connection.</p> <p>Syntax: (void) User_RoutingActivationCallback(SoAd_SoConId- Type SoConId, uint16 SourceAddr, uint8 ActivationType, const uint8* OemSpecificPtr);</p> <p>with User_RoutingActivationCallback as a placeholder for a user defined call- back function name provided with this configuration field.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>SoConId - SoConId of a Tcp connection that routing activation message was re- ceived on.</li> <li>SourceAddr - Source address field from routing activation message.</li> <li>ActivationType - Activation type field from routing activation message.</li> <li>OemSpecificPtr - Pointer to OEM specific field from routing activation mes- sage (4 bytes). NULL_PTR if OEM specific field was omitted.</li> </ul> <p><b>Note:</b> User defined header files shall be added to the configuration container DolPHeaderFileInclusion.</p>
<b>Multiplicity</b>	0..255
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>PreCompile:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.2.1.24. DolPGetGid

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">DolPGetGID</a>	0..1
<b>Parameter Name</b>	<b>DolPGetGID</b>



<b>Description</b>	Direct C Callback function to get the GID of the DoIP entity.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FUNCTION-NAME	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.2.1.25. DoIPPowerModeCallback

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">DoIPPowerMode</a>	0..1
<b>Parameter Name</b>	
<b>DoIPPowerMode</b>	
<b>Description</b>	Direct C Callback function to get the Power Mode of the DoIP entity. If the DoIP-PowerMode parameter is present, the DoIP module will not use an RPort of ServiceInterface CallbackGetPowerMode but will call the configured function.
<b>Multiplicity</b>	0..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>VariantPostBuild:</b>
	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.2.1.26. DoIPTTriggerGidSynchronization

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">DoIPTTriggerGidSynchronization</a>	0..1
<b>Parameter Name</b>	
<b>DoIPTTriggerGidSynchronization</b>	
<b>Description</b>	Direct C Callback function to trigger the synchronization of the GID. If the DoIP-TriggerGidSynchronization parameter is present, the DoIP module will not use an RPort of ServiceInterface CallbackTriggerGidSynchronization but call the configured function.
	<b>Note: Feature not supported.</b>
<b>Multiplicity</b>	0..1



Type	FUNCTION-NAME	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.2.1.27. DoIPDynamicGIDMasterSelectionCallback

Parameters included	
Parameter name	Multiplicity
<a href="#">DoIPDynamicGIDMasterSelection</a>	0..1

Parameter Name	<b>DoIPDynamicGIDMasterSelection</b>	
Description	Direct C Callback function to get the DoIP entity GID master/slave status.	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	<b>Link:</b>	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

### 5.2.1.28. DoIPVIDRequestReceivedCallback

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

Parameter Name	<b>DoIPVIDRequestReceived</b>
Description	User defined callback function to be called for every valid received VID request.  Syntax: (void) UserDefinedFunction(uint8 DoIPInterfaceId)  with UserDefinedFunction as placeholder for a user defined function name provided with this configuration field.  <b>Note:</b> User defined header files can be added to configuration container DoIP-HeaderFileInclusion.
Multiplicity	1..1
Type	FUNCTION-NAME



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.2.1.29. DoIPGetVIN

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

<b>Parameter Name</b>	<b>DoIPGetVin</b>
<b>Description</b>	The function should store VIN into specified location.
<b>Multiplicity</b>	0..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>Link:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.2.1.30. DoIPTcpSoConModeChgCallback

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

<b>Parameter Name</b>	<b>DoIPTcpSoConModeChg</b>
<b>Description</b>	User defined callback function to be called to provide information about the TCP connection status.  Syntax: (void) UserDefinedFunction(SoAd_SoConIdType SoConId, SoAd_SoConModeType Status)  with UserDefinedFunction as placeholder for a user defined function name provided with this configuration field.  <b>Note:</b> User defined header files can be added to configuration container DoIP-HeaderFileInclusion.
<b>Multiplicity</b>	1..1
<b>Type</b>	FUNCTION-NAME



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.2.1.31. DoIPDefensiveProgramming

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">DoIPDefProgEnabled</a>	1..1
<a href="#">DoIPPrecondAssertEnabled</a>	1..1
<a href="#">DoIPPostcondAssertEnabled</a>	1..1
<a href="#">DoIPStaticAssertEnabled</a>	1..1
<a href="#">DoIPUnreachAssertEnabled</a>	1..1
<a href="#">DoPIInvariantAssertEnabled</a>	1..1

<b>Parameter Name</b>	<b>DoIPDefProgEnabled</b>
<b>Label</b>	Enable Defensive Programming
<b>Description</b>	<p>Enables or disables the defensive programming feature for the module DoIP.</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>	<b>VariantPostBuild:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>DoIPPrecondAssertEnabled</b>
<b>Label</b>	Enable Precondition Assertions
<b>Description</b>	<p>Enables handling of precondition assertion checks reported from the module DoIP.</p> <p>Dependency on parameter(s):</p>



	<ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>DoIPDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>DoIPDefProgEnabled</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>DolPPostcondAssertEnabled</b>
<b>Label</b>	Enable Postcondition Assertions
<b>Description</b>	<p>Enables handling of postcondition assertion checks reported from the module DolP.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>DoIPDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>DoIPDefProgEnabled</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>DolPStaticAssertEnabled</b>
<b>Label</b>	Enable Static Assertions
<b>Description</b>	<p>Enables handling of static assertion checks reported from the module DolP.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>DoIPDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>DoIPDefProgEnabled</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>DolPUnreachAssertEnabled</b>
<b>Label</b>	Enable Unreachable Code Assertions
<b>Description</b>	<p>Enables handling of unreachable code assertion checks reported from the module DolP.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>DoIPDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>DoIPDefProgEnabled</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>DolPIInvariantAssertEnabled</b>
<b>Label</b>	Enable Invariant Assertions
<b>Description</b>	<p>Enables handling of invariant assertion checks reported from functions of the module DolP.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>DoIPDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>DoIPDefProgEnabled</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



Origin	Elektrobit Automotive GmbH
--------	----------------------------

### 5.2.1.32. PublishedInformation

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

Parameter Name	<b>PbcfgMSupport</b>
Label	PbcfgM support
Description	Specifies whether or not the DoIP can use the PbcfgM module for post-build support.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	<b>PublishedInformation:</b>
Origin	Elektrobit Automotive GmbH

## 5.2.2. Application programming interface (API)

### 5.2.2.1. Macro constants

#### 5.2.2.1.1. DOIP\_E\_INVALID\_PARAMETER

Purpose	
Value	0x04U
Description	DET error code: Invalid parameter

#### 5.2.2.1.2. DOIP\_E\_INVALID\_PDU\_SDU\_ID

Purpose	
---------	--



<b>Value</b>	0x03U
<b>Description</b>	DET error code: Invalid ID

#### 5.2.2.1.3. DOIP\_E\_PARAM\_POINTER

<b>Purpose</b>	
<b>Value</b>	0x02U
<b>Description</b>	DET error code: Invalid pointer (NULL_PTR)

#### 5.2.2.1.4. DOIP\_E\_UNINIT

<b>Purpose</b>	
<b>Value</b>	0x01U
<b>Description</b>	DET error code: Module not initialized

#### 5.2.2.1.5. DOIP\_INSTANCE\_ID

<b>Purpose</b>	Module instance ID.
<b>Value</b>	0U
<b>Description</b>	Defines the instance number of this module. Since multiple instances are not supported this ID is always zero.

#### 5.2.2.1.6. DOIP\_SID\_ACTIVATIONLINESWITCH

<b>Purpose</b>	
<b>Value</b>	0x0FU
<b>Description</b>	SID for DoIP_ActivationLineSwitch()

#### 5.2.2.1.7. DOIP\_SID\_CANCELRECEIVE

<b>Purpose</b>	
<b>Value</b>	0x05U
<b>Description</b>	SID for <a href="#">DoIP_CancelReceive()</a>



#### 5.2.2.1.8. DOIP\_SID\_CANCELTRANSMIT

<b>Purpose</b>	
<b>Value</b>	0x04U
<b>Description</b>	SID for <a href="#">DoIP_CancelTransmit()</a>

#### 5.2.2.1.9. DOIP\_SID\_GETANDRESETMEASUREMENTDATA

<b>Purpose</b>	Defines API id of function <a href="#">DoIP_GetAndResetMeasurementData()</a> .
<b>Value</b>	0x50U

#### 5.2.2.1.10. DOIP\_SID\_GETVERSIONINFO

<b>Purpose</b>	
<b>Value</b>	0x00U
<b>Description</b>	SID for <a href="#">DoIP_GetVersionInfo()</a>

#### 5.2.2.1.11. DOIP\_SID\_IFTRANSMIT

<b>Purpose</b>	
<b>Value</b>	0x49U
<b>Description</b>	SID for <a href="#">DoIP_IfTransmit()</a>

#### 5.2.2.1.12. DOIP\_SID\_INIT

<b>Purpose</b>	
<b>Value</b>	0x01U
<b>Description</b>	SID for <a href="#">DoIP_Init()</a>

#### 5.2.2.1.13. DOIP\_SID\_LOCALIPADDRASSIGNMENTCHG

<b>Purpose</b>	
----------------	--



<b>Value</b>	0x0CU
<b>Description</b>	SID for <a href="#">DoIP_LocallpAddrAssignmentChg()</a>

#### 5.2.2.1.14. DOIP\_SID\_MAINFUNCTION

<b>Purpose</b>	
<b>Value</b>	0x02U
<b>Description</b>	SID for <a href="#">DoIP_MainFunction()</a>

#### 5.2.2.1.15. DOIP\_SID\_MAINFUNCTION\_TX

<b>Purpose</b>	
<b>Value</b>	0xF1U
<b>Description</b>	SID for <a href="#">DoIP_MainFunctionTx()</a>

#### 5.2.2.1.16. DOIP\_SID\_SOADIFRXINDICATION

<b>Purpose</b>	
<b>Value</b>	0x42U
<b>Description</b>	SID for <a href="#">DoIP_SoAdIfRxIndication()</a>

#### 5.2.2.1.17. DOIP\_SID\_SOADIFTXCONFIRMATION

<b>Purpose</b>	
<b>Value</b>	0x40U
<b>Description</b>	SID for <a href="#">DoIP_SoAdIfTxConfirmation()</a>

#### 5.2.2.1.18. DOIP\_SID\_SOADTPCOPYRXDATA

<b>Purpose</b>	
<b>Value</b>	0x08U
<b>Description</b>	SID for <a href="#">DoIP_SoAdTpCopyRxData()</a>



#### 5.2.2.1.19. DOIP\_SID\_SOADTPCOPYTXDATA

<b>Purpose</b>	
<b>Value</b>	0x06U
<b>Description</b>	SID for <a href="#">DoIP_SoAdTpCopyTxData()</a>

#### 5.2.2.1.20. DOIP\_SID\_SOADTPRXINDICATION

<b>Purpose</b>	
<b>Value</b>	0x0AU
<b>Description</b>	SID for <a href="#">DoIP_SoAdTpRxIndication()</a>

#### 5.2.2.1.21. DOIP\_SID\_SOADTPSTARTOFRECEPTION

<b>Purpose</b>	
<b>Value</b>	0x09U
<b>Description</b>	SID for <a href="#">DoIP_SoAdTpStartOfReception()</a>

#### 5.2.2.1.22. DOIP\_SID\_SOADTPTXCONFIRMATION

<b>Purpose</b>	
<b>Value</b>	0x07U
<b>Description</b>	SID for <a href="#">DoIP_SoAdTpTxConfirmation()</a>

#### 5.2.2.1.23. DOIP\_SID\_SOCONMODECHG

<b>Purpose</b>	
<b>Value</b>	0x0BU
<b>Description</b>	SID for <a href="#">DoIP_SoConModeChg()</a>

#### 5.2.2.1.24. DOIP\_SID\_TPTRANSMIT

<b>Purpose</b>	
----------------	--



<b>Value</b>	0x49U
<b>Description</b>	SID for <a href="#">DoIP_TpTransmit()</a>

### 5.2.2.2. Functions

#### 5.2.2.2.1. DoIP\_ActivationLineSwitchActive

<b>Purpose</b>	API to enable connections.
<b>Synopsis</b>	<code>void DoIP_ActivationLineSwitchActive ( void );</code>
<b>Service ID</b>	0x0F
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non reentrant
<b>Description</b>	This function opens all configured socket connections.

#### 5.2.2.2.2. DoIP\_ActivationLineSwitchInactive

<b>Purpose</b>	API to disable connections.
<b>Synopsis</b>	<code>void DoIP_ActivationLineSwitchInactive ( void );</code>
<b>Service ID</b>	0x0F
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non reentrant
<b>Description</b>	This function closes all configured socket connections.

#### 5.2.2.2.3. DoIP\_CancelReceive

<b>Purpose</b>	API to cancel a reception from SoAd to PduR.
<b>Synopsis</b>	<code>Std_ReturnType DoIP_CancelReceive ( PduIdType DoIPPduRRxId );</code>
<b>Service ID</b>	0x05
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Reentrant for different Pdulds. Non reentrant for the same Pduld



<b>Parameters (in)</b>	DoIP_PduRRxId	- DoIP handle ID to be used for DoIP API calls from PduR.
<b>Return Value</b>	Std_ReturnType	
	E_OK	- Request is accepted.
	E_NOT_OK	- Request is rejected.

#### 5.2.2.2.4. DoIP\_CancelTransmit

<b>Purpose</b>	API to cancel a transmission from PduR to SoAd.	
<b>Synopsis</b>	Std_ReturnType <b>DoIP_CancelTransmit</b> ( PduIdType DoIP_PduRTxId );	
<b>Service ID</b>	0x04	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different Pdulds. Non reentrant for the same Pduld	
<b>Parameters (in)</b>	DoIP_PduRTxId	- DoIP handle ID to be used for DoIP API calls from PduR.
<b>Return Value</b>	Std_ReturnType	
	E_OK	- Request is accepted.
	E_NOT_OK	- Request is rejected.

#### 5.2.2.2.5. DoIP\_GetAndResetMeasurementData

<b>Purpose</b>	API to read and reset measurement data.	
<b>Synopsis</b>	Std_ReturnType <b>DoIP_GetAndResetMeasurementData</b> ( DoIP_MeasurementIdxType MeasurementIdx , boolean MeasurementResetNeeded , uint32 * MeasurementDataPtr );	
<b>Parameters (in)</b>	MeasurementIdx	Index to select specific measurement data: DOIP_MEAS_DROP_GENHDR (0x01) - Measurement index of dropped messages caused by invalid generic header. DOIP_MEAS_DROP_DIAGMSG (0x02) - Measurement index of dropped diag messages. DOIP_MEAS_ALL (0xFF) - Represents all measurement indexes.
	MeasurementResetNeeded	Flag to trigger a reset of the measurement data.



<b>Parameters (out)</b>	MeasurementDataPtr	Pointer to data buffer, where to copy measurement data.
<b>Return Value</b>	Std_ReturnType	
	E_OK	The function has been successfully executed.
	E_NOT_OK	The function could not be successfully executed.
<b>Description</b>	This service allows to read and reset detailed measurement data for diagnostic purposes. Get all MeasurementIdx's at once is not supported. DOIP_MEAS_ALL shall only be used to reset all MeasurementIdx's at once. A NULL_PTR shall be provided for MeasurementDataPtr in this case.	

#### 5.2.2.2.6. DoIP\_GetVersionInfo

<b>Purpose</b>	API to get the module version information.	
<b>Synopsis</b>	<code>void DoIP_GetVersionInfo ( Std_VersionInfoType * VersionInfo );</code>	
<b>Service ID</b>	0x00	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (out)</b>	VersionInfo	- Pointer to return the module version information.
<b>Description</b>	This service returns the version information of this module.	

#### 5.2.2.2.7. DoIP\_Init

<b>Purpose</b>	API to initialize the module.	
<b>Synopsis</b>	<code>void DoIP_Init ( const DoIP_ConfigType * DoIP_ConfigPtr );</code>	
<b>Service ID</b>	0x01	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	DoIP_ConfigPtr	- Pointer to the module post build configuration.
<b>Description</b>	This service initializes the modules shared variables.	



### 5.2.2.2.8. DoIP\_IsValidConfig

<b>Purpose</b>	Checks compatibility of the post-build-time configuration.	
<b>Synopsis</b>	<code>Std_ReturnType DoIP_IsValidConfig ( const void * voidConfigPtr ) ;</code>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	<code>voidConfigPtr</code>	Pointer to the configuration data of the DoIP module.
<b>Return Value</b>	<code>E_OK</code>	Provided configuration is compatible.
<code>E_NOT_OK</code>	Provided configuration is not compatible.	
<b>Description</b>	This service checks the compatibility of the post-build-time configuration	

### 5.2.2.2.9. DoIP\_LocalIpAddrAssignmentChg

<b>Purpose</b>	API to indicate local IP address assignment changes.	
<b>Synopsis</b>	<code>void DoIP_LocalIpAddrAssignmentChg ( SoAd_SoConIdType SoConId , TcpIp_IpAddrStateType State ) ;</code>	
<b>Service ID</b>	0x0C	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SoConIds. Non reentrant for the same SoConId	
<b>Parameters (in)</b>	<code>SoConId</code>	- Socket connection index.
	<code>State</code>	- Returns information if IP address is assigned or not.

### 5.2.2.2.10. DoIP\_MainFunction

<b>Purpose</b>	Main function API.
<b>Synopsis</b>	<code>void DoIP_MainFunction ( void ) ;</code>
<b>Service ID</b>	0x02
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Description</b>	This function executes module tasks periodically.  fixed



#### 5.2.2.2.11. DoIP\_MainFunctionTx

<b>Purpose</b>	Transmit part of the main function if enabled to be external.
<b>Synopsis</b>	<code>void DoIP_MainFunctionTx ( void );</code>
<b>Service ID</b>	0xF1

#### 5.2.2.2.12. DoIP\_SoAdIfRxIndication

<b>Purpose</b>	API to indicate a UDP reception.	
<b>Synopsis</b>	<code>void DoIP_SoAdIfRxIndication ( PduIdType RxPduId , PduInfoType * PduInfoPtr );</code>	
<b>Service ID</b>	0x42	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different Pdulds. Non reentrant for the same Pduld	
<b>Parameters (in)</b>	RxPduId	- DoIP handle ID to be used for DoIP APIs to be called from SoAd.
	PduInfoPtr	- Pointer to received data.

#### 5.2.2.2.13. DoIP\_SoAdIfTxConfirmation

<b>Purpose</b>	API to confirm a UDP transmission.	
<b>Synopsis</b>	<code>void DoIP_SoAdIfTxConfirmation ( PduIdType TxPduId );</code>	
<b>Service ID</b>	0x40	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non reentrant	
<b>Parameters (in)</b>	TxPduId	- DoIP handle ID to be used for DoIP APIs to be called from SoAd.

#### 5.2.2.2.14. DoIP\_SoAdTpCopyRxData

<b>Purpose</b>	API to provide received data.
<b>Synopsis</b>	<code>BufReq_ReturnType DoIP_SoAdTpCopyRxData ( PduIdType RxPduId , const PduInfoType * PduInfoPtr , PduLengthType * BufferSizePtr ) ;</code>
<b>Service ID</b>	0x08



<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different Pduls. Non reentrant for the same Pdul	
<b>Parameters (in)</b>	RxPduId	- DoIP handle ID to be used for DoIP APIs to be called from SoAd.
	PduInfoPtr	- Pointer providing received data and data length.
<b>Parameters (out)</b>	BufferSizePtr	- Amount of data which shall be provided next call.
<b>Return Value</b>	BufReq_ReturnType	
	BUFREQ_OK	- Data is copied.
	BUFREQ_E_BUSY	- Request postponed. No data is copied.
	BUFREQ_E_NOT_OK	- Request failed.

#### 5.2.2.2.15. DoIP\_SoAdTpCopyTxData

<b>Purpose</b>	API to request data to transmit.	
<b>Synopsis</b>	<pre>BufReq_ReturnType DoIP_SoAdTpCopyTxData ( PduIdType TxPduId , PduInfoType * PduInfoPtr , RetryInfoType * Retry , Pdu- LengthType * AvailableDataPtr );</pre>	
<b>Service ID</b>	0x06	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non reentrant	
<b>Parameters (in)</b>	TxPduId	- DoIP handle ID to be used for DoIP APIs to be called from SoAd.
	PduInfoPtr	- Pointer providing a buffer and length to copy the Tx data.
	Retry	- This parameter is expected to be a NULL_PTR as retry is not supported by the DoIP.
<b>Parameters (out)</b>	AvailableDataPtr	- Pointer which returns remaining number of bytes to be copied.
<b>Return Value</b>	BufReq_ReturnType	
	BUFREQ_OK	- Data is copied.
	BUFREQ_E_BUSY	- Request postponed. No data is copied.
	BUFREQ_E_NOT_OK	- Request failed.



### 5.2.2.2.16. DoIP\_SoAdTpRxIndication

<b>Purpose</b>	API to indicate that all TCP receptions from this tester are finished.	
<b>Synopsis</b>	<pre>void DoIP_SoAdTpRxIndication ( PduIdType RxPduId , NotifResultType Result );</pre>	
<b>Service ID</b>	0x0A	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different Pdulds. Non reentrant for the same Pduld	
<b>Parameters (in)</b>	RxPduId	- DoIP handle ID to be used for DoIP APIs to be called from SoAd.
	Result	- Result of the finished reception.

### 5.2.2.2.17. DoIP\_SoAdTpStartOfReception

<b>Purpose</b>	API to start a reception.	
<b>Synopsis</b>	<pre>BufReq_ReturnType DoIP_SoAdTpStartOfReception ( PduIdType RxPduId , PduLengthType TpSduLength , PduLengthType * BufferSizePtr );</pre>	
<b>Service ID</b>	0x09	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different Pdulds. Non reentrant for the same Pduld	
<b>Parameters (in)</b>	RxPduId	- DoIP handle ID to be used for DoIP APIs to be called from SoAd.
	TpSduLength	- Message length shall always be zero for DoIP.
<b>Parameters (out)</b>	BufferSizePtr	Available Rx buffer in the DoIP module.
<b>Return Value</b>	BufReq_ReturnType	
	BUFREQ_OK	- Reception request has been accepted. RxBufferSizePtr indicates the available receive buffer.
	BUFREQ_E_OVFL	- No Buffer of the required length can be provided.
	BUFREQ_E_NOT_OK	- Reception request has been rejected. RxBufferSizePtr remains unchanged.
<b>Description</b>	This function is called once by SoAd if a TCP connection to a tester is established.	



#### 5.2.2.2.18. DoIP\_SoAdTpTxConfirmation

<b>Purpose</b>	API to confirm a TCP transmission.	
<b>Synopsis</b>	<pre>void DoIP_SoAdTpTxConfirmation ( PduIdType TxPduId , NotifResultType Result );</pre>	
<b>Service ID</b>	0x07	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non reentrant	
<b>Parameters (in)</b>	TxPduId	- DoIP handle ID to be used for DoIP APIs to be called from SoAd.
	Result	- Parameter indicates the result of the transmission.

#### 5.2.2.2.19. DoIP\_SoConModeChg

<b>Purpose</b>	API to indicate SoAd socket connection state change.	
<b>Synopsis</b>	<pre>void DoIP_SoConModeChg ( SoAd_SoConIdType SoConId , SoAd_SoConModeType Mode );</pre>	
<b>Service ID</b>	0x0B	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SoConIds. Non reentrant for the same SoConId	
<b>Parameters (in)</b>	SoConId	- Socket connection index.
	Mode	- New connection mode.

#### 5.2.2.2.20. DoIP\_TpTransmit

<b>Purpose</b>	API to request Tp data transfer from the PduR to the SoAd.	
<b>Synopsis</b>	<pre>Std_ReturnType DoIP_TpTransmit ( PduIdType DoIPPduRTxId , const PduInfoType * DoIPPduRTxInfoPtr );</pre>	
<b>Service ID</b>	0x03	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant for different Pduids. Non reentrant for the same Pduid	
<b>Parameters (in)</b>	DoIPPduRTxId	- DoIP handle ID to be used for DoIP API calls from PduR.

	DoIPduRTxInfoPtr	- Pointer to Tx PDU information structure which contains the length of the message to transmit.
<b>Return Value</b>	Std_ReturnType	
	E_OK	- Request accepted
	E_NOT_OK	- Request not accepted

## 5.2.3. Integration notes

### 5.2.3.1. Exclusive areas

This section describes the exclusive areas used by the DoIP module.

#### 5.2.3.1.1. SCHM\_DOIP\_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 Mapping exclusive areas in the basic software modules in the Integration notes section for details.

### 5.2.3.2. Production errors

Production errors are not reported by the DoIP module.

### 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:



Memory section
CODE
CONFIG_DATA_UNSPECIFIED
VAR_INIT_8
VAR_INIT_32
VAR_INIT_UNSPECIFIED
VAR_CLEARED_8
VAR_CLEARED_16
VAR_CLEARED_32
VAR_CLEARED_UNSPECIFIED
CONST_8
CONST_32
CODE

#### 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. DoIP.EB\_INTREQ\_DoIP\_0001

<b>Description</b>	The reinitialization process shall not interrupt other module functions. If reinitialization of the module is required, the call of DoIP_Init() shall not interrupt other module functions.
<b>Rationale</b>	The reinitialization process resets all internal variables. Continuing an interrupted module function after reinitialization can lead to undefined module behavior.

##### 5.2.3.4.2. DoIP.EB\_INTREQ\_DoIP\_0002

<b>Description</b>	DoIP_Init() shall not be preempted by any other module API calls. It needs to be ensured that the function call DoIP_Init() is not preempted by any other module API calls.
--------------------	---



<b>Rationale</b>	During the call of DoIP_Init() global variables and pointers get initialized. It is easy for the integrator to avoid this preemption, thus no data protection mechanism has been implemented for function DoIP_Init().
------------------	--

#### 5.2.3.4.3. DoIP.EB\_INTREQ\_DoIP\_0003

<b>Description</b>	The following parameters have to be manually configured: <ul style="list-style-type: none"> <li>▶ DoIPRoutingActivation</li> <li>▶ DoIP_Eid, DoIP_Gid, DoIP_LogicalAddress</li> <li>▶ DoIPTester</li> </ul>
<b>Rationale</b>	A Com transformer for DoIP is available which creates a local ECU configuration based on the system description. Only those parameters have to be manually configured.

#### 5.2.3.4.4. DoIP.EB\_INTREQ\_DoIP\_0004

<b>Description</b>	In case configuration parameter DoIPGeneral/DoIPRteUsage is disabled, user provided power mode callback function include file shall contain definition of DoIP_PowerStateType according to [SWS_DoIP_00266], i.e. it needs to contain the following code: <pre>#define DOIP_NOT_READY 0x00U #define DOIP_READY 0x01U #define DOIP_NOT_SUPPORTED 0x02U typedef uint8 DoIP_PowerStateType;</pre> In case Rte usage is enabled, RTE will generate above mentioned definitions.
--------------------	--

#### 5.2.3.4.5. DoIP.EB\_INTREQ\_DoIP\_0005

<b>Description</b>	The integrator shall ensure that DoIP lower layer functions and DoIP main functions do not preempt each other. List of DoIP lower layer: DoIP_SoAdIfRxIndication(), DoIP_SoAdIfTxConfirmation(), DoIP_SoAdTpStartOfReception(), DoIP_SoAdTpCopyRxData(), DoIP_SoAdTpRxIndication(), DoIP_SoAdTpCopyTxData(), DoIP_SoAdTpTxConfirmation(), DoIP_SoConModeChg(), DoIP_LocallIpAddrAssignmentChg() List of DoIP main functions: DoIP_MainFunction() DoIP_MainFunctionTx() In the classic AUTOSAR environment this can be achieved by setting EthIf and Eth driver in polling mode (e.g. disabling of EthIfEnableRxInterrupt, EthCtrlEnableRxInterrupt, EthIfEnableTxInterrupt and EthCtrlEnableTxInterrupt) to enforce that lower layer API are called only in context of main functions. In addition all Eth stack main functions (e.g. EthIf_MainFunctionRx(), EthIf_MainFunctionTx(), EthIf_MainFunctionState(), Tcplp_-
--------------------	--



	MainFunction(), Tcplp_MainFunctionTx(), EthSM_MainFunction(), SoAd_MainFunctionTx(), SoAd_MainFunction()) need to be in the same Os task or have the same task priority as the module main functions to eliminate preemption. This integration requirement also applies for possible other not listed DoIP lower layer functions.
<b>Rationale</b>	This limitation reduces code size and execution time by eliminating the need for extensive use of exclusive areas.

#### 5.2.3.4.6. DoIP.EB\_INTREQ\_DoIP\_0006

<b>Description</b>	In case configuration parameter DoIPTcpConnection/DoIPTcpConnectionSecurityRequired is set to TRUE, corresponding TCP connection shall be secure (e.g. TLS).
--------------------	--

#### 5.2.3.4.7. DoIP.EB\_INTREQ\_DoIP\_0007

<b>Description</b>	If Diagnostic message is received before the route is activated, on a channel with DoIPRoutingActivationType DOIP_ROUTING_ACTIVATION_AUTOMATIC, then this SourceAddress is already registered on another Tcp connection and Tester can send a Routing Activation on this Tcp connection in order to free this SourceAddress.
--------------------	--

## 5.3. EthIf

### 5.3.1. 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="#">EthIfConfigSet</a>	1..1	Collecting container for all parameters with post-build configuration classes.
<a href="#">EthIfGeneral</a>	1..1	This container contains the general configuration parameters of the Ethernet Interface.



#### Containers included

<a href="#">EthIfDefensiveProgramming</a>	1..1	<b>Label:</b> Defensive Programming Options  Parameters for defensive programming
<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

#### 5.3.1.1. CommonPublishedInformation

Parameters included	
Parameter name	Multiplicity
<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	<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>	3
<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>	9
<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>	19
<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>	65
<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 AUTOSAR 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

### 5.3.1.2. EthIfConfigSet

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">EthIfController</a>	1..n	This container is a subcontainer of EthIfConfigSet and specifies the configuration parameters for a EthIf controller.
<a href="#">EthIfFrameOwnerConfig</a>	1..n	Configuration of Ethernet frame owner.  <b>Note:</b> Every Ethernet frame type which shall be received needs to be listed in this container. A recommended configuration is available which sets common Ethernet frame types together with its upper layer callback functions.
<a href="#">EthIfPhysController</a>	1..255	This container contains the configuration of EthIfPhysController.
<a href="#">EthIfRxIndicationConfig</a>	1..254	Configuration of receive callback functions.
<a href="#">EthIfSwitch</a>	0..n	This container contains the configuration of EthIfSwitches.
<a href="#">EthIfSwitchMgmtInfoIndicationConfig</a>	0..n	Configuration of Switch Management callback function.



<b>Containers included</b>		
		<b>Note:</b> This configuration container is not used.
<a href="#">EthIfSwitchPortGroup</a>	0..n	This container contains the configuration of EthIfSwitchPortGroups.
<a href="#">EthIfSwitchTimeStampIndicationConfig</a>	0..n	Configuration of Switch timestamp indications.  <b>Note:</b> This configuration container is not used.
<a href="#">EthIfTransceiver</a>	1..255	This container contains the configuration of EthIfTransceiver. The configuration flag EthIfTrcvSupportEnable enables this container.
<a href="#">EthIfTrcvLinkStateChgConfig</a>	1..254	Specifies link state change callback function.  If a link state change occurs, all listed upper layer callback functions are called.
<a href="#">EthIfTxConfirmationConfig</a>	1..254	Configuration of transmit confirmation callback functions.  In case of a transmission confirmation, all listed upper layer callback functions are called.
<a href="#">EthIfEthControllerType</a>	0..255	
<a href="#">EthIfEthTrcvType</a>	0..255	
<a href="#">EthIfEthSwtType</a>	0..1	

### 5.3.1.3. EthIfController

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EthIfCtrlIdx</a>	1..1
<a href="#">EthIfCtrlMtu</a>	1..1
<a href="#">EthIfMaxTxBufsTotal</a>	1..1
<a href="#">EthIfVlanId</a>	0..1
<a href="#">EthIfEthTrcvRef</a>	0..1
<a href="#">EthIfPhysControllerRef</a>	1..1
<a href="#">EthIfSwitchRefOrPortGroupRef</a>	0..1

<b>Parameter Name</b>	<b>EthIfCtrlIdx</b>
<b>Label</b>	EthIf controller index



<b>Description</b>	EthIf controller index.	
<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>EthIfCtrlMtu</b>	
<b>Label</b>	Maximum transmission unit	
<b>Description</b>	Specifies the maximum transmission unit (MTU) of the EthIfController in [bytes]. Note: in case a VLAN tag is used for the EthIfController, the MTU is 4 bytes smaller than the maximum payload size of an Ethernet frame which can be transmitted on the network.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	1500	
<b>Range</b>	<=9000 >=64	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>EthIfMaxTxBufsTotal</b>	
<b>Description</b>	Limits the total number of transmit buffers.  <b>This configuration parameter is not used</b>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	1	
<b>Range</b>	<=4294967295 >=1	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	



<b>Parameter Name</b>	<b>EthIfVlanId</b>	
<b>Label</b>	VLAN identifier	
<b>Description</b>	12 bit VLAN Identifier which is part of the 4-byte VLAN header as specified by IEEE 802.1Q. Hexadecimal values of 0x000 and 0xFFFF are reserved. All other values may be used as VLAN identifiers, allowing up to 4,094 VLANs.	
<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>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>EthIfEthTrcvRef</b>	
<b>Label</b>	Ethernet interface transceiver reference	
<b>Description</b>	Reference to the Ethernet transceiver on which this connection will be transmitted/received. Connections are specified in the Socket Adapter [9].	
<b>Multiplicity</b>	0..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>EthIfPhysControllerRef</b>	
<b>Label</b>	Ethernet interface physical controller reference	
<b>Description</b>	Reference to the Physical controller reference in Ethernet Driver on which this connection will be transmitted / received. Connections are specified in the Socket Adapter [9].	
<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>EthIfSwitchRefOrPortGroupRef</b>	
<b>Description</b>	The choice reference allows to configure either the EthIfController references an EthIfSwitch or an EthIfSwitchPortGroup.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	CHOICE-REFERENCE	



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.3.1.4. EthIfFrameOwnerConfig

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

<b>Parameter Name</b>	<b>EthIfFrameType</b>
<b>Description</b>	Selects the Ethernet frame type.  <b>Note:</b> Typical values are 0x0800 for IPv4 or 0x0806 for ARP. A recommended configuration is available which sets common Ethernet frame types.  <b>Note:</b> The VLAN frame type (0x8100) does not need to be configured since EthIf processes all VLAN frames and uses the encapsulated frame type to identify the frame owner.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EthIfOwner</b>
<b>Description</b>	Selects the owner of an Ethernet frame type. The owner is a zero based index into the callback function configuration 'EthIfRxIndicationConfig'. I.e. an Ethernet frame of type IPv4 (0x800) at index 0 will call the first callback function configured in 'EthIfRxIndicationConfig'.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Range</b>	<=254 >=0



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.3.1.5. EthIfPhysController

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">EthIfPhysCtrlRxMainFunctionPriorityProcessing</a>	0..255	This container is a subcontainer of EthIfPhysController and specifies the configuration parameters for the priority processing.

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EthIfPhysControllerIdx</a>	1..1
<a href="#">EthIfEthCtrlRef</a>	1..1
<a href="#">EthIfWEthCtrlRef</a>	0..1

<b>Parameter Name</b>	<b>EthIfPhysControllerIdx</b>
<b>Description</b>	This parameter provides a zero-based consecutive index of the physical Ethernet controllers. Upper layer BSW modules and the Ethernet Interface itself use this index to identify a physical Ethernet controller.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Range</b>	<=255 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EthIfEthCtrlRef</b>
<b>Description</b>	Reference to the controller in Ethernet Driver on which this connection will be transmitted / received. Connections are specified in the Socket Adapter [9].
<b>Multiplicity</b>	1..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b>



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

<b>Parameter Name</b>	<b>EthIfWEthCtrlRef</b>	
<b>Multiplicity</b>	0..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.3.1.6. EthIfPhysCtrlRxMainFunctionPriorityProcessing

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

<b>Parameter Name</b>	<b>EthIfPhysCtrlRxIndicationIterations</b>	
<b>Description</b>	Maximum number of Ethernet frames per Ethernet controller polled from the Ethernet driver within EthIf_MainFunctionRxPrio.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	10	
<b>Range</b>	<=65535 >=1	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>EthIfPhysCtrlRxMainFunctionPeriod</b>	
<b>Description</b>	Specifies the period of main function EthIf_MainFunctionRxPrio in seconds.  <b>Note:</b> Ethernet Interface does not require this information but the BSW scheduler.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	



<b>Default value</b>	0.01
<b>Range</b>	<=65535 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EthIfPhysCtrlRxIngressFifoRef</b>
<b>Description</b>	Reference to the ethernet controller ingress FIFO.
<b>Multiplicity</b>	1..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Range</b>	node:when(count(node:ref(..../EthIfEthCtrlRef)/EthCtrlConfigIngress/EthCtrlConfigIngressFifo/*) > 0, node:paths(node:ref(..../EthIfEthCtrlRef)/EthCtrlConfigIngress/EthCtrlConfigIngressFifo/*), "")
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.3.1.7. EthIfRxIndicationConfig

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

<b>Parameter Name</b>	<b>EthIfRxIndicationFunction</b>
<b>Description</b>	Specifies receive indication callback function.  <b>Note:</b> A typical receive indication function entry is Tcplp_RxIndication. A recommended configurations are available which add common receive indication functions.  <b>Note:</b> Enter EthIf_Up_RxIndicationDummy if no receive indication callback function shall be called.
<b>Multiplicity</b>	1..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC



### 5.3.1.8. EthIfSwitch

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

Parameter Name	<b>EthIfSwitchIdx</b>	
Description	This parameter provides a zero-based consecutive index of the Ethernet Interface Switches. Upper layer BSW modules and the EthIf itself use this index to identify a Ethernet Switch.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=255 >=0	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>EthIfSwitchRef</b>	
Description	Reference to a Ethernet Switch, which is handled by a specific Ethernet Switch driver.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.3.1.9. EthIfSwitchMgmtInfoIndicationConfig

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



Type	FUNCTION-NAME	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.3.1.10. EthIfSwitchPortGroup

Parameters included		
Parameter name	Multiplicity	
<a href="#">EthIfSwitchPortGroupIdx</a>	1..1	
<a href="#">EthIfSwitchPortGroupRefSemantics</a>	0..1	
<a href="#">EthIfPortRef</a>	1..n	

Parameter Name	<b>EthIfSwitchPortGroupIdx</b>	
Multiplicity	1..1	
Type	INTEGER	
Range	<=255 >=0	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>EthIfSwitchPortGroupRefSemantics</b>	
Multiplicity	0..1	
Type	ENUMERATION	
Range	ETHIF_SWITCH_PORT_GROUP_CONTROL ETHIF_SWITCH_PORT_GROUP_LINK_INFO	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>EthIfPortRef</b>	
Multiplicity	1..n	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild



Origin	AUTOSAR_ECUC
--------	--------------

### 5.3.1.11. EthIfSwitchTimeStampIndicationConfig

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

Parameter Name	<b>EthIfSwitchEgressTimeStampIndicationFunction</b>	
Multiplicity	1..1	
Type	FUNCTION-NAME	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>EthIfSwitchIngressTimeStampIndicationFunction</b>	
Multiplicity	1..1	
Type	FUNCTION-NAME	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.3.1.12. EthIfTransceiver

Parameters included	
Parameter name	Multiplicity
<a href="#">EthIfTransceiverIdx</a>	1..1
<a href="#">EthIfEthTrcvRef</a>	0..1
<a href="#">EthIfWEthTrcvRef</a>	0..1

Parameter Name	<b>EthIfTransceiverIdx</b>
Description	This parameter provides a zero-based consecutive index of the Ethernet transceivers. Upper layer BSW modules and the Ethernet Interface itself use this index to identify an Ethernet transceiver.



<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Range</b>	<=255 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EthIfEthTrcvRef</b>
<b>Description</b>	Reference to an Ethernet transceiver, which is handled by a specific Ethernet transceiver driver.
<b>Multiplicity</b>	0..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>EthIfWEthTrcvRef</b>
<b>Multiplicity</b>	0..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.3.1.13. EthIfTrcvLinkStateChgConfig

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

<b>Parameter Name</b>	<b>EthIfTrcvLinkStateChgFunction</b>
<b>Description</b>	Specifies link state change callback function.  <b>Note:</b> Enter EthIf_Up_TrsvLinkStateChgDummy if no link state change function shall be called.
<b>Multiplicity</b>	1..1
<b>Type</b>	FUNCTION-NAME



<b>Default value</b>	EthSM_TrcvLinkStateChg	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.3.1.14. EthIfTxConfirmationConfig

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
EthIfTxConfirmationFunction	1..1

<b>Parameter Name</b>	<b>EthIfTxConfirmationFunction</b>	
<b>Description</b>	Specifies transmit confirmation callback function.  <b>Note:</b> Enter EthIf_Up_TxConfirmationDummy if transmit confirmation function shall be called.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FUNCTION-NAME	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.3.1.15. EthIfEthControllerType

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
EthIfEthControllerBswmdlImplementationRefs	1..1

<b>Parameter Name</b>	<b>EthIfEthControllerBswmdlImplementationRefs</b>	
<b>Description</b>	Reference to the BswImplementation of the underlying Controller which contains the vendorId and vendorApiInfix. To be configured only when support of multiple Eth Controllers is required, or when Eth Controller that contains vendorId and vendorApiInfix is used. Supported for Eth Controller versions 4.3.0. and above.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FOREIGN-REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild



<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

### 5.3.1.16. EthIfEthTrcvType

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

<b>Parameter Name</b>	<b>EthIfEthTrcvBswmdImplementationRefs</b>	
<b>Description</b>	Reference to the BswImplementation of the underlying Transceiver which contains the vendorId and vendorApiInfix. To be configured only when support of multiple Eth Transceivers is required, or when Eth Transceiver that contains vendorId and vendorApiInfix is used. Supported for Eth Transceiver version 4.3.0. and above.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FOREIGN-REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.3.1.17. EthIfEthSwtType

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

<b>Parameter Name</b>	<b>EthIfEthSwtBswmdImplementationRefs</b>	
<b>Description</b>	Reference to the BswImplementation of the underlying Switch which contains the vendorId and vendorApiInfix. To be configured only when support of Eth Switch that contains vendorId and vendorApiInfix is used. Supported for Eth Switch version 4.3.0. and above.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FOREIGN-REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	



### 5.3.1.18. EthIfGeneral

Containers included		
Container name	Multiplicity	Description
<a href="#">ReportToDem</a>	1..1	<b>Label:</b> Production error handling Production error handling

Parameters included	
Parameter name	Multiplicity
<a href="#">EthIfDevErrorDetect</a>	1..1
<a href="#">EthIfEnableRxInterrupt</a>	1..1
<a href="#">EthIfEnableTxInterrupt</a>	1..1
<a href="#">EthIfEnableWEthApi</a>	0..1
<a href="#">EthIfGetAndResetMeasurementDataApi</a>	1..1
<a href="#">EthIfGetBaudRate</a>	1..1
<a href="#">EthIfGetCounterState</a>	1..1
<a href="#">EthIfGetCtrlIdxList</a>	1..1
<a href="#">EthIfGetTransceiverWakeupModeApi</a>	0..1
<a href="#">EthIfGetVlanIdSupport</a>	1..1
<a href="#">EthIfGlobalTimeSupport</a>	1..1
<a href="#">EthIfMainFunctionPeriod</a>	1..1
<a href="#">EthIfMainFunctionStatePeriod</a>	0..1
<a href="#">EthIfMaxTrcvTotal</a>	1..1
<a href="#">EthIfPortStartupActiveTime</a>	0..1
<a href="#">EthIfPublicCddHeaderFile</a>	0..255
<a href="#">EthIfRxIndicationIterations</a>	1..1
<a href="#">EthIfSetForwardingModeApi</a>	1..1
<a href="#">EthIfStartAutoNegotiation</a>	1..1
<a href="#">EthIfSwitchManagementSupport</a>	1..1
<a href="#">EthIfSwitchOffPortTimeDelay</a>	0..1
<a href="#">EthIfTrcvLinkStateChgMainReload</a>	1..1
<a href="#">EthIfVerifyConfigApi</a>	1..1
<a href="#">EthIfVersionInfoApi</a>	1..1
<a href="#">EthIfVersionInfoApiMacro</a>	1..1



### Parameters included

<a href="#">EthIfWakeUpSupport</a>	1..1
<a href="#">EthIfMiiApiEnable</a>	1..1
<a href="#">EthIfSwitchingPortGroupSupport</a>	1..1
<a href="#">EthIfAsyncEthTrcvModeSupport</a>	1..1
<a href="#">EthIfAsyncEthCtrlModeSupport</a>	1..1
<a href="#">EthIfPublicHandleTypeEnum</a>	1..1
<a href="#">EthIfSupportEthAPI</a>	1..1
<a href="#">EthIfMaxCtrl</a>	1..1
<a href="#">EthIfMaxPhyCtrl</a>	1..1
<a href="#">EthIfMaxEthSwitches</a>	1..1
<a href="#">EthIfMaxSwtPorts</a>	1..1
<a href="#">EthIfMaxSwtPortGroups</a>	1..1
<a href="#">EthIfRelocatablePbcfgEnable</a>	1..1
<a href="#">EthIfSetPhysAddrSupportEnable</a>	1..1
<a href="#">EthIfTrcvSupportEnable</a>	1..1
<a href="#">EthIfUpdatePhysAddrFilterSupportEnable</a>	1..1
<a href="#">EthIfVirtualCtrlSupportEnable</a>	1..1
<a href="#">EthIfVLANSupportEnable</a>	1..1
<a href="#">EthIfGetArlTableApi</a>	1..1
<a href="#">EthIfGetBufferLevelApi</a>	1..1
<a href="#">EthIfSwtGetCounterValuesApi</a>	1..1
<a href="#">EthIfGetPortMacAddrApi</a>	1..1
<a href="#">EthIfResetConfigurationApi</a>	1..1
<a href="#">EthIfStoreConfigurationApi</a>	1..1
<a href="#">EthIfSetModeTimeout</a>	0..1
<a href="#">EthIfInitControllersTransceivers</a>	1..1
<a href="#">EthIfSwtPreProcessRxFrame</a>	1..1
<a href="#">EthIfSwtAdpatTxFrame</a>	1..1
<a href="#">EthIfDeviceAuthenticationApiEnable</a>	1..1
<a href="#">EthIfRetransmitApiEnable</a>	1..1

Parameter Name	EthIfDevErrorDetect



<b>Description</b>	Enables/Disables development error detection.	
<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>EthIfEnableRxInterrupt</b>	
<b>Description</b>	Enables / Disables receive interrupt.  <b>This configuration parameter is not used</b>	
<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>EthIfEnableTxInterrupt</b>	
<b>Description</b>	Enables / Disables the transmit interrupt.  <b>This configuration parameter is not used</b>	
<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>EthIfEnableWEthApi</b>	
<b>Description</b>	Enables / Disables API's for WEth / WEthTrcv  <b>This configuration parameter is not used</b>	
<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>EthIfGetAndResetMeasurementDataApi</b>	
<b>Description</b>	Enables / Disables the Get and Reset Measurement Data API	
<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>EthIfGetBaudRate</b>	
<b>Description</b>	Enables / Disables GetBaudRate API.  <b>This configuration parameter is not used</b>	
<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>EthIfGetCounterState</b>	
<b>Description</b>	Enables / Disables GetCounterState API.  <b>This configuration parameter is not used</b>	
<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>EthIfGetCtrlIdxList</b>	
<b>Description</b>	Enables / Disables GetCtrlIdxList API.  <b>This configuration parameter is not used</b>	
<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>EthIfGetTransceiverWakeupModeApi</b>
<b>Description</b>	Enables / Disables EthIf_GetTransceiverWakeupMode API.
<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>EthIfGetVlanIdSupport</b>
<b>Description</b>	Enables / Disables GetVlanId API.
<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>EthIfGlobalTimeSupport</b>
<b>Description</b>	Enables/Disables the Global Time APIs used amongst others by Global Time Synchronization over Ethernet.
<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>EthIfMainFunctionPeriod</b>
<b>Description</b>	Specifies the period of functions EthIf_MainFunctionRx() and EthIf_MainFunctionTx() in seconds.  Ethernet Interface does not require this information but the BSW scheduler.
<b>Multiplicity</b>	1..1



<b>Type</b>	FLOAT
<b>Default value</b>	0.01
<b>Range</b>	<=65535 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EthIfMainFunctionStatePeriod</b>
<b>Description</b>	Specifies the period of main function EthIf_MainFunctionState in seconds. Ethernet Interface does not require this information but the BSW scheduler.
<b>Multiplicity</b>	0..1
<b>Type</b>	FLOAT
<b>Range</b>	<=65535 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EthIfMaxTrcvTotal</b>
<b>Description</b>	Limits the total number of transceivers.
<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>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EthIfPortStartupActiveTime</b>
<b>Description</b>	Time delay after which all ports started with API call EthIf_StartAllPorts() are switched off.
<b>Multiplicity</b>	0..1
<b>Type</b>	FLOAT
<b>Range</b>	<=65.535



	>=0.001
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EthIfPublicCddHeaderFile</b>
<b>Description</b>	Defines header files for callback functions which shall be included in case of CDDs. Range of characters is 1..32.
<b>Multiplicity</b>	0..255
<b>Type</b>	STRING
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EthIfRxIndicationIterations</b>
<b>Description</b>	Maximum number of Ethernet frames per Ethernet controller polled from the Ethernet driver within EthIf_MainFunctionRx.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	10
<b>Range</b>	<=65535 >=1
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EthIfSetForwardingModeApi</b>
<b>Description</b>	Enables /disables EthIf_SetForwardingMode API.  <b>This configuration parameter is not used</b>
<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>EthIfStartAutoNegotiation</b>
-----------------------	----------------------------------



<b>Description</b>	Enables/Disables StartAutoNegotiation API.
<b>This configuration parameter is not used</b>	
<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>EthIfSwitchManagementSupport</b>
<b>Description</b>	Enables/Disables the Switch management APIs to support a Switch-port specific communication attribute access.
<b>This configuration parameter is not used</b>	
<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>EthIfSwitchOffPortTimeDelay</b>
<b>Description</b>	Denote the time delay after the mode "ETHTRCV_MODE_DOWN" of a EthIfSwitchPortGroup will be executed. This is only used for EthIfSwtPortGroups which not referenced by a EthIfController or the reference is of type "link-information".
<b>Multiplicity</b>	0..1
<b>Type</b>	FLOAT
<b>Range</b>	<=65.535 >=0.001
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EthIfTrcvLinkStateChgMainReload</b>
<b>Description</b>	Specifies the frequency of transceiver link state change checks in each period of main function EthIf_MainFunctionTx().
<b>This configuration parameter is not used</b>	



<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>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>EthIfVerifyConfigApi</b>
<b>Description</b>	Enables /disables EthIf_VerifyConfig API.  <b>This configuration parameter is not used</b>
<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>EthIfVersionInfoApi</b>
<b>Description</b>	Enables/Disables version info API.
<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>EthIfVersionInfoApiMacro</b>
<b>Description</b>	Enables/Disables version info API macro implementation.  <b>This configuration parameter is not used</b>
<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>EthIfWakeUpSupport</b>	
<b>Description</b>	Configures if Ethernet Wakeup is supported or not.	
<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>EthIfMiiApiEnable</b>	
<b>Description</b>	Enables / Disables EthIf_ReadMii and EthIf_WriteMii APIs.	
<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>EthIfSwitchingPortGroupSupport</b>	
<b>Description</b>	Enables / Disables switching of switch ports groups feature.	
<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>EthIfAsyncEthTrcvModeSupport</b>	
<b>Description</b>	Switch between sync/asyn behaviour of EthTrcv.  ▶ True: Asynchronous communication with EthTrcv - callback EthIf_TrcvModeIndication - is enabled. ▶ False: Asynchronous communication with EthTrcv - callback EthIf_TrcvModeIndication - is 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>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EthIfAsyncEthCtrlModeSupport</b>
<b>Description</b>	<p>Switch between sync/asyn behaviour of Eth Controllers.</p> <ul style="list-style-type: none"> <li>▶ True: Asynchronous communication with EthCtrl - callback EthIf_CtrlModeIndication - is enabled.</li> <li>▶ False: Asynchronous communication with EthCtrl - callback EthIf_CtrlModeIndication - is 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>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EthIfPublicHandleTypeEnum</b>
<b>Description</b>	EthIfPublicHandleTypeEnum is used to configure the type Eth_BufIdxType. The type Eth_BufIdxType represents the hardware buffer handle of a Eth hardware buffer. This configuration parameter must be configured according to the Eth driver's needs.
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	UINT8
<b>Range</b>	UINT32 UINT8
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EthIfSupportEthAPI</b>
<b>Description</b>	<p>EthIfSupportEthAPI parameter is used to select support of correct version of Eth API, according to the version of AUTOSAR.</p> <p>ASR403: AUTOSAR 4.0 rev. 3</p> <p>ASR422: AUTOSAR 4.2 rev. 2</p>



	ASR430: AUTOSAR 4.3 rev. 0 ASR430_EB: compatible with current 4.3.0 EB Eth drivers
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	ASR430
<b>Range</b>	ASR403 ASR422 ASR430 ASR430_EB
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EthIfMaxCtrl</b>
<b>Description</b>	Limits the total number of controllers.
<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

<b>Parameter Name</b>	<b>EthIfMaxPhyCtrl</b>
<b>Description</b>	Limits the total number of used physical Ethernet controllers (Eth).
<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

<b>Parameter Name</b>	<b>EthIfMaxEthSwitches</b>
-----------------------	----------------------------



<b>Description</b>	Limits the total number of Ethernet switches.	
<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>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EthIfMaxSwtPorts</b>	
<b>Description</b>	Limits the total number of switch ports.	
<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>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EthIfMaxSwtPortGroups</b>	
<b>Description</b>	Limits the total number of switch port groups.	
<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>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>EthIfRelocatablePbcfgEnable</b>	
<b>Description</b>	Enables/disable support for relocatable postbuild configuration. ▶ True: Postbuild configuration relocatable in memory.	



	▶ False: Postbuild configuration not relocatable in memory.
<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>EthIfSetPhysAddrSupportEnable</b>
<b>Description</b>	<p>This parameter allows to enable and disable the usage of Ethernet driver API Eth_SetPhysAddr().</p> <ul style="list-style-type: none"> <li>▶ <b>True:</b> EthIf_SetPhysAddr() will call Ethernet driver API Eth_SetPhysAddr().</li> <li>▶ <b>False:</b> EthIf_SetPhysAddr() will return without any lower layer calls. Eth_SetPhysAddr() is not required.</li> </ul> <p><b>Note:</b> Eth_SetPhysAddr() does not exists in AUTOSAR 4.0.3. Therefore, it is required to disable this configuration parameter for such Ethernet drivers.</p>
<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>EthIfTrcvSupportEnable</b>
<b>Description</b>	<p>Switch to enable / disable EthTrcv support. If disabled, all API functions accessing the EthTrcv are excluded.</p> <ul style="list-style-type: none"> <li>▶ True: The EthIf supports controlling the EthTrcv.</li> <li>▶ False: The EthIf does not support controlling the EthTrcv.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (code):</b> Enabling this parameter reduces the ROM consumption of the module code.</li> <li>▶ <b>Execution time reduction (code):</b> Enabling this parameter reduces the execution time of the module code.</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>EthIfUpdatePhysAddrFilterSupportEnable</b>	
<b>Description</b>	<p>This parameter allows to enable and disable the usage of Ethernet driver API Eth_UpdatePhysAddrFilter().</p> <ul style="list-style-type: none"> <li>▶ <b>True:</b> EthIf_UpdatePhysAddrFilter() will call Ethernet driver API Eth_UpdatePhysAddrFilter().</li> <li>▶ <b>False:</b> EthIf_UpdatePhysAddrFilter() will return without any lower layer calls. Eth_UpdatePhysAddrFilter() is not required.</li> </ul> <p><b>Note:</b> Eth_UpdatePhysAddrFilter() does not exists in AUTOSAR 4.0.3. Therefore, it is required to disable this configuration parameter for such Ethernet drivers.</p>	
<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>EthIfVirtualCtrlSupportEnable</b>	
<b>Description</b>	<p>Switch to enable / disable virtual EthIf Controller support. If enabled, multiple EthIf Controllers can be configured per EthCtrl or EthTrcv respectively. This allows the use of more than one IP address per Ethernet Controller.</p> <ul style="list-style-type: none"> <li>▶ True: The EthIf supports virtual EthIf Controllers.</li> <li>▶ False: The EthIf does not support virtual EthIf Controllers.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</li> <li>▶ <b>Execution time reduction (code):</b> Disabling this parameter reduces the execution time of the module code.</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>EthIfVLANSupportEnable</b>
<b>Description</b>	<p>Switch to enable / disable VLAN support.</p> <ul style="list-style-type: none"> <li>▶ True: The EthIf supports VLAN.</li> <li>▶ False: The EthIf does not support VLAN.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module 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>EthIfGetArlTableApi</b>
<b>Description</b>	Enables / Disables EthIf_GetArlTable API.
<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>EthIfGetBufferLevelApi</b>
<b>Description</b>	Enables / Disables API to fetch the switch buffer utilization.
<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>EthIfSwtGetCounterValuesApi</b>
-----------------------	------------------------------------



<b>Description</b>	Enables / Disables EthIf_SwtGetCounterValues API.	
<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>EthIfGetPortMacAddrApi</b>	
<b>Description</b>	Enables / Disables EthIf_GetPortMacAddr API.	
<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>EthIfResetConfigurationApi</b>	
<b>Description</b>	Enables / Disables EthIf_ResetConfiguration API.	
<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>EthIfStoreConfigurationApi</b>	
<b>Description</b>	Enables / Disables EthIf_StoreConfiguration API.	
<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>EthIfSetModeTimeout</b>	
<b>Description</b>	Specifies in which time duration EthIf module shall repeat (delay) a call to set mode APIs.	



<b>Multiplicity</b>	0..1
<b>Type</b>	FLOAT
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EthIfInitControllersTransceivers</b>
<b>Description</b>	Switch to enable / disable EthIf to initialize configured physical controllers and transceivers. This config parameter shall be disabled for ASR 4.2.2 and higher Eth drivers and transceivers.
<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>EthIfSwtPreProcessRxFrame</b>
<b>Description</b>	<p>Switch to enable / disable the call of EthSwt_EthIfPreProcessRxFrame() to allow the switch driver to preprocess received frames. This is necessary for frames which are created or modified by switches and do not have a proper EtherType which is required for routing in EthIf.</p> <ul style="list-style-type: none"> <li>▶ True: EthSwt_EthIfPreProcessRxFrame() is called.</li> <li>▶ False: No preprocessing is performed.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>Execution time reduction (code):</b> Disabling this parameter reduces the execution time of the module code.</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>EthIfSwtAdpatTxFrame</b>
<b>Description</b>	Switch to enable / disable the functionality to extend outgoing frames with additional switch specific data.



	<ul style="list-style-type: none"> <li>▶ True: EthIf calls EthSwt APIs to modify Tx frames.</li> <li>▶ False: No Tx frame modification is performed.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>Execution time reduction (code):</b> Disabling this parameter reduces the execution time of the module code.</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>EthIfDeviceAuthenticationApiEnable</b>
<b>Description</b>	Enables / Disables EthIf_EnableRelatedEthIfCtrls and EthIf_DisableRelatedEthIfCtrls APIs.
<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>EthIfRetransmitApiEnable</b>
<b>Description</b>	Enables / Disables EthIf_Retransmit API.
<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

### 5.3.1.19. ReportToDem

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>



#### Parameters included

<a href="#">EthIfDemCtrlTestResultReportToDem</a>	1..1
<a href="#">EthIfDemCtrlTestResultReportToDemDetErrorId</a>	1..1

<b>Parameter Name</b>	<b>EthIfDemCtrlTestResultReportToDem</b>	
<b>Label</b>	EthIfDemCtrlTestResult report to	
<b>Description</b>	<p>Selects the handling of the production error EthIfDemCtrlTestResult.</p> <ul style="list-style-type: none"> <li>▶ DEM: The error is reported to the Diagnostics Event Manager (Dem).</li> <li>▶ DET: The error is reported to the Development Error Tracer (Det) if enabled.</li> <li>▶ DISABLE: The error is not reported at all.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (code):</b> Setting this parameter to a value of DISABLE reduces the ROM consumption of the module code.</li> <li>▶ <b>Execution time reduction (code):</b> Setting this parameter to a value of DISABLE reduces the execution time of the module code.</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>EthIfDemCtrlTestResultReportToDemDetErrorId</b>	
<b>Label</b>	EthIfDemCtrlTestResult Dem To Det error ID	
<b>Description</b>	If a production error is reported towards the Det, EthIfDemCtrlTestResultReportToDemDetErrorId defines the error ID which is reported towards the Det.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	145	
<b>Range</b>	<=255 >=0	

<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.3.1.20. EthIfDefensiveProgramming

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EthIfDefProgEnabled</a>	1..1
<a href="#">EthIfPrecondAssertEnabled</a>	1..1
<a href="#">EthIfPostcondAssertEnabled</a>	1..1
<a href="#">EthIfStaticAssertEnabled</a>	1..1
<a href="#">EthIfUnreachAssertEnabled</a>	1..1
<a href="#">EthIfInvariantAssertEnabled</a>	1..1

<b>Parameter Name</b>	<b>EthIfDefProgEnabled</b>
<b>Label</b>	Enable Defensive Programming
<b>Description</b>	<p>Enables or disables the defensive programming feature for the module EthIf.</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>	<b>VariantPostBuild:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EthIfPrecondAssertEnabled</b>
<b>Label</b>	Enable Precondition Assertions
<b>Description</b>	<p>Enables handling of precondition assertion checks reported from the module EthIf.</p> <p>Dependency on parameter(s):</p>



	<ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EthIfDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EthIfDefProgEnabled</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>EthIfPostcondAssertEnabled</b>
<b>Label</b>	Enable Postcondition Assertions
<b>Description</b>	<p>Enables handling of postcondition assertion checks reported from the module EthIf.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EthIfDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EthIfDefProgEnabled</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>EthIfStaticAssertEnabled</b>
<b>Label</b>	Enable Static Assertions
<b>Description</b>	<p>Enables handling of static assertion checks reported from the module EthIf.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EthIfDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EthIfDefProgEnabled</code>): must be enabled</li> </ul>
<b>Multiplicity</b>	1..1



Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>EthIfUnreachAssertEnabled</b>	
Label	Enable Unreachable Code Assertions	
Description	<p>Enables handling of unreachable code assertion checks reported from the module EthIf.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EthIfDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EthIfDefProgEnabled</code>): must be enabled</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>EthIfInvariantAssertEnabled</b>	
Label	Enable Invariant Assertions	
Description	<p>Enables handling of invariant assertion checks reported from functions of the module EthIf.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EthIfDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EthIfDefProgEnabled</code>): must be enabled</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	Elektrobit Automotive GmbH
--------	----------------------------

### 5.3.1.21. PublishedInformation

Parameters included	
Parameter name	<b>Multiplicity</b>
<a href="#">PbcfgMSupport</a>	1..1
Parameter Name	<b>PbcfgMSupport</b>
Label	PbcfgM support
Description	Specifies whether or not the EthIf can use the PbcfgM module for post-build support.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	<b>PublishedInformation:</b>
Origin	Elektrobit Automotive GmbH

## 5.3.2. Recommended configurations

### 5.3.2.1. EthIfRecConfiguration

Containers included	
Container name	<b>Container definition</b>
<a href="#">EthIfConfigSet</a>	<a href="#">EthIfConfigSet</a>
Parameters included	
Parameter name	<b>Value</b>

#### 5.3.2.1.1. EthIfConfigSet

Containers included	
Container name	<b>Container definition</b>

**Containers included**

<a href="#">EthIfController_0</a>	<a href="#">EthIfController</a>
<a href="#">EthIfFrameOwnerConfig_IPv4</a>	<a href="#">EthIfFrameOwnerConfig</a>
<a href="#">EthIfFrameOwnerConfig_IPv6</a>	<a href="#">EthIfFrameOwnerConfig</a>
<a href="#">EthIfFrameOwnerConfig_ARP</a>	<a href="#">EthIfFrameOwnerConfig</a>
<a href="#">EthIfPhysController_0</a>	<a href="#">EthIfPhysController</a>
<a href="#">EthIfRxIndicationConfig_IPv4</a>	<a href="#">EthIfRxIndicationConfig</a>
<a href="#">EthIfRxIndicationConfig_IPv6</a>	<a href="#">EthIfRxIndicationConfig</a>
<a href="#">EthIfRxIndicationConfig_ARP</a>	<a href="#">EthIfRxIndicationConfig</a>
<a href="#">EthIfTrcvLinkStateChgConfig_EthSM</a>	<a href="#">EthIfTrcvLinkStateChgConfig</a>
<a href="#">EthIfTxConfirmationConfig_0</a>	<a href="#">EthIfTxConfirmationConfig</a>

**Parameters included**

Parameter name	Value

**5.3.2.1.2. EthIfController\_0****Parameters included**

Parameter name	Value
<a href="#">EthIfCtrlIdx</a>	0

**5.3.2.1.3. EthIfFrameOwnerConfig\_IPv4****Parameters included**

Parameter name	Value
<a href="#">EthIfFrameType</a>	2048
<a href="#">EthIfOwner</a>	0

**5.3.2.1.4. EthIfFrameOwnerConfig\_IPv6****Parameters included**

Parameter name	Value
<a href="#">EthIfFrameType</a>	34525

**Parameters included**

<a href="#">EthIfOwner</a>	1
----------------------------	---

**5.3.2.1.5. EthIfFrameOwnerConfig\_ARP****Parameters included**

Parameter name	Value
<a href="#">EthIfFrameType</a>	2054
<a href="#">EthIfOwner</a>	2

**5.3.2.1.6. EthIfPhysController\_0****Parameters included**

Parameter name	Value
<a href="#">EthIfPhysControllerIdx</a>	0

**5.3.2.1.7. EthIfRxIndicationConfig\_IPv4****Parameters included**

Parameter name	Value
<a href="#">EthIfRxIndicationFunction</a>	TcpIp_RxIndication

**5.3.2.1.8. EthIfRxIndicationConfig\_IPv6****Parameters included**

Parameter name	Value
<a href="#">EthIfRxIndicationFunction</a>	TcpIp_RxIndication

**5.3.2.1.9. EthIfRxIndicationConfig\_ARP****Parameters included**

Parameter name	Value
----------------	-------



#### Parameters included

<a href="#">EthIfRxIndicationFunction</a>	TcpIp_RxIndication
---	--------------------

#### 5.3.2.1.10. EthIfTrcvLinkStateChgConfig\_EthSM

##### Parameters included

Parameter name	Value
<a href="#">EthIfTrcvLinkStateChgFunction</a>	EthSM_TrsvLinkStateChg

#### 5.3.2.1.11. EthIfTxConfirmationConfig\_0

##### Parameters included

Parameter name	Value
<a href="#">EthIfTxConfirmationFunction</a>	EthIf_Up_TxConfirmationDummy

#### 5.3.2.2. EthIfRecConfigurationEthTSyn

##### Containers included

Container name	Container definition
<a href="#">EthIfConfigSet</a>	<a href="#">EthIfConfigSet</a>
<a href="#">EthIfGeneral</a>	<a href="#">EthIfGeneral</a>

##### Parameters included

Parameter name	Value

#### 5.3.2.2.1. EthIfConfigSet

##### Containers included

Container name	Container definition
<a href="#">EthIfController_0</a>	<a href="#">EthIfController</a>
<a href="#">EthIfFrameOwnerConfig_PTP</a>	<a href="#">EthIfFrameOwnerConfig</a>
<a href="#">EthIfFrameOwnerConfig_IPv4</a>	<a href="#">EthIfFrameOwnerConfig</a>

**Containers included**

<a href="#">EthIfFrameOwnerConfig_IPv6</a>	<a href="#">EthIfFrameOwnerConfig</a>
<a href="#">EthIfFrameOwnerConfig_ARP</a>	<a href="#">EthIfFrameOwnerConfig</a>
<a href="#">EthIfPhysController_0</a>	<a href="#">EthIfPhysController</a>
<a href="#">EthIfRxIndicationConfig_PTP</a>	<a href="#">EthIfRxIndicationConfig</a>
<a href="#">EthIfRxIndicationConfig_IPv4</a>	<a href="#">EthIfRxIndicationConfig</a>
<a href="#">EthIfRxIndicationConfig_IPv6</a>	<a href="#">EthIfRxIndicationConfig</a>
<a href="#">EthIfRxIndicationConfig_ARP</a>	<a href="#">EthIfRxIndicationConfig</a>
<a href="#">EthIfTrcvLinkStateChgConfig_EthSM</a>	<a href="#">EthIfTrcvLinkStateChgConfig</a>
<a href="#">EthIfTrcvLinkStateChgConfig_EthTSyn</a>	<a href="#">EthIfTrcvLinkStateChgConfig</a>
<a href="#">EthIfTxConfirmationConfig_PTP</a>	<a href="#">EthIfTxConfirmationConfig</a>

**Parameters included**

Parameter name	Value

**5.3.2.2.2. EthIfController\_0****Parameters included**

Parameter name	Value
<a href="#">EthIfCtrlIdx</a>	0

**5.3.2.2.3. EthIfFrameOwnerConfig\_PTP****Parameters included**

Parameter name	Value
<a href="#">EthIfFrameType</a>	35063
<a href="#">EthIfOwner</a>	0

**5.3.2.2.4. EthIfFrameOwnerConfig\_IPv4****Parameters included**

Parameter name	Value
<a href="#">EthIfFrameType</a>	2048

**Parameters included**

<a href="#">EthIfOwner</a>	1
----------------------------	---

**5.3.2.2.5. EthIfFrameOwnerConfig\_IPv6****Parameters included**

Parameter name	Value
<a href="#">EthIfFrameType</a>	34525
<a href="#">EthIfOwner</a>	2

**5.3.2.2.6. EthIfFrameOwnerConfig\_ARP****Parameters included**

Parameter name	Value
<a href="#">EthIfFrameType</a>	2054
<a href="#">EthIfOwner</a>	3

**5.3.2.2.7. EthIfPhysController\_0****Parameters included**

Parameter name	Value
<a href="#">EthIfPhysControllerIdx</a>	0

**5.3.2.2.8. EthIfRxIndicationConfig\_PTP****Parameters included**

Parameter name	Value
<a href="#">EthIfRxIndicationFunction</a>	EthTSyn_RxIndication

**5.3.2.2.9. EthIfRxIndicationConfig\_IPv4****Parameters included**

Parameter name	Value

**Parameters included**

<a href="#">EthIfRxIndicationFunction</a>	TcpIp_RxIndication
---	--------------------

**5.3.2.2.10. EthIfRxIndicationConfig\_IPv6****Parameters included**

Parameter name	Value
<a href="#">EthIfRxIndicationFunction</a>	TcpIp_RxIndication

**5.3.2.2.11. EthIfRxIndicationConfig\_ARP****Parameters included**

Parameter name	Value
<a href="#">EthIfRxIndicationFunction</a>	TcpIp_RxIndication

**5.3.2.2.12. EthIfTrcvLinkStateChgConfig\_EthSM****Parameters included**

Parameter name	Value
<a href="#">EthIfTrcvLinkStateChgFunction</a>	EthSM_TrcvLinkStateChg

**5.3.2.2.13. EthIfTrcvLinkStateChgConfig\_EthTSyn****Parameters included**

Parameter name	Value
<a href="#">EthIfTrcvLinkStateChgFunction</a>	EthTSyn_TrcvLinkStateChg

**5.3.2.2.14. EthIfTxConfirmationConfig\_PTP****Parameters included**

Parameter name	Value
<a href="#">EthIfTxConfirmationFunction</a>	EthTSyn_TxConfirmation



### 5.3.2.2.15. EthIfGeneral

Parameters included	
Parameter name	Value

## 5.3.3. Application programming interface (API)

### 5.3.3.1. Macro constants

#### 5.3.3.1.1. ETHIF\_CBK\_RXINDICATION\_SVCID

Purpose	Defines API id of function <a href="#">EthIf_Cbk_RxIndication()</a> .
Value	0x10U

#### 5.3.3.1.2. ETHIF\_CBK\_TXCONFIRMATION\_SVCID

Purpose	Defines API id of function <a href="#">EthIf_Cbk_TxConfirmation()</a> .
Value	0x11U

#### 5.3.3.1.3. ETHIF\_CHECKWAKEUP\_SVCID

Purpose	Defines API id of function <a href="#">EthIf_CheckWakeup()</a> .
Value	0x30U

#### 5.3.3.1.4. ETHIF\_CLEARSWITCHPORTSIGNALQUALITY\_SVCID

Purpose	Defines API id of function <a href="#">EthIf_ClearSwitchPortSignalQuality()</a> .
Value	0x1bU

#### 5.3.3.1.5. ETHIF\_CLEARTRCV SIGNALQUALITY\_SVCID

Purpose	Defines API id of function <a href="#">EthIf_ClearTrcvSignalQuality()</a> .
---------	---



<b>Value</b>	0x19U
--------------	-------

#### 5.3.3.1.6. ETHIF\_CTRLMODEINDICATION\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_CtrlModeIndication()</a> .
<b>Value</b>	0x0EU

#### 5.3.3.1.7. ETHIF\_DISABLERELATEDETHIFCTRLS\_SVCID

<b>Purpose</b>	Defines API ID of function <a href="#">EthIf_DisableRelatedEthIfCtrls()</a> .
<b>Value</b>	0xF1U

#### 5.3.3.1.8. ETHIF\_ENABLEEGRESSTIMESTAMP\_SVCID

<b>Purpose</b>	Defines API ID of function <a href="#">EthIf_EnableEgressTimeStamp()</a> .
<b>Value</b>	0x23U

#### 5.3.3.1.9. ETHIF\_ENABLERELATEDETHIFCTRLS\_SVCID

<b>Purpose</b>	Defines API ID of function <a href="#">EthIf_EnableRelatedEthIfCtrls()</a> .
<b>Value</b>	0xF0U

#### 5.3.3.1.10. ETHIF\_E\_HW\_NOT\_INITIALIZED

<b>Purpose</b>	API requests called with invalid parameter.
<b>Value</b>	0x80U

#### 5.3.3.1.11. ETHIF\_E\_INIT\_FAILED

<b>Purpose</b>	Initialization of EthIf module failed.
<b>Value</b>	0x07U



### 5.3.3.1.12. ETHIF\_E\_INV\_CTRL\_IDX

<b>Purpose</b>	API requests called with invalid controller index.
<b>Value</b>	0x01U

### 5.3.3.1.13. ETHIF\_E\_INV\_PARAM

<b>Purpose</b>	API requests called with invalid parameter.
<b>Value</b>	0x06U

### 5.3.3.1.14. ETHIF\_E\_INV\_PORT\_GROUP\_IDX

<b>Purpose</b>	API requests called with invalid port group index.
<b>Value</b>	0x03U

### 5.3.3.1.15. ETHIF\_E\_INV\_SWT\_IDX

<b>Purpose</b>	API requests called with invalid switch index.
<b>Value</b>	0x90U

### 5.3.3.1.16. ETHIF\_E\_INV\_TRCV\_IDX

<b>Purpose</b>	API requests called with invalid transceiver index.
<b>Value</b>	0x02U

### 5.3.3.1.17. ETHIF\_E\_NOT\_INITIALIZED

<b>Purpose</b>	API requests called before Ethif module is initialized.
<b>Value</b>	0x04U

### 5.3.3.1.18. ETHIF\_E\_PARAM\_POINTER

<b>Purpose</b>	API requests called with invalid pointer in parameter list.
----------------	---



<b>Value</b>	0x05U
--------------	-------

#### 5.3.3.1.19. ETHIF\_GETANDRESETMEASUREMENTDATA\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_GetAndResetMeasurementData()</a> .
<b>Value</b>	0x45U

#### 5.3.3.1.20. ETHIF\_GETARLTABLE\_SVCID

<b>Purpose</b>	Defines API ID of function <a href="#">EthIf_GetArlTable()</a> .
<b>Value</b>	0x29U

#### 5.3.3.1.21. ETHIF\_GETBUFFERLEVEL\_SVCID

<b>Purpose</b>	Defines API ID of function <a href="#">EthIf_GetBufferLevel()</a> .
<b>Value</b>	0x2AU

#### 5.3.3.1.22. ETHIF\_GETCABLEDIAGNOSTICSRESULT\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_GetCableDiagnosticsResult()</a> .
<b>Value</b>	0x14U

#### 5.3.3.1.23. ETHIF\_GETCONTROLLERMODE\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_GetControllerMode()</a> .
<b>Value</b>	0x04U

#### 5.3.3.1.24. ETHIF\_GETCTRLIDXLIST\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_GetCtrlIdxList()</a> .
<b>Value</b>	0x44U



### 5.3.3.1.25. ETHIF\_GETCURRENTTIME\_SVCID

<b>Purpose</b>	Defines API ID of function <a href="#">EthIf_GetCurrentTime()</a> .
<b>Value</b>	0x22U

### 5.3.3.1.26. ETHIF\_GETEGRESSTIMESTAMP\_SVCID

<b>Purpose</b>	Defines API ID of function <a href="#">EthIf_GetEgressTimeStamp()</a> .
<b>Value</b>	0x24U

### 5.3.3.1.27. ETHIF\_GETINGRESSTIMESTAMP\_SVCID

<b>Purpose</b>	Defines API ID of function <a href="#">EthIf_GetIngressTimeStamp()</a> .
<b>Value</b>	0x25U

### 5.3.3.1.28. ETHIF\_GETPHYIDENTIFIER\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_GetPhyIdentifier()</a> .
<b>Value</b>	0x15U

### 5.3.3.1.29. ETHIF\_GETPHYSADDR\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_GetPhysAddr()</a> .
<b>Value</b>	0x08U

### 5.3.3.1.30. ETHIF\_GETPORTMACADDR\_SVCID

<b>Purpose</b>	Defines API ID of function <a href="#">EthIf_GetPortMacAddr()</a> .
<b>Value</b>	0x28U

### 5.3.3.1.31. ETHIF\_GETSWITCHPORTSIGNALQUALITY\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_GetSwitchPortSignalQuality()</a> .
----------------	---



<b>Value</b>	0x1aU
--------------	-------

#### 5.3.3.1.32. ETHIF\_GETTRANSCEIVERWAKEUPMODE\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_GetTransceiverWakeupMode()</a> .
<b>Value</b>	0x2FU

#### 5.3.3.1.33. ETHIF\_GETTRCVSIGNALQUALITY\_SVCID

<b>Purpose</b>	Defines API id of function EthIf_GetTrcvSignalQuality().
<b>Value</b>	0x18U

#### 5.3.3.1.34. ETHIF\_GETVERSIONINFO\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_GetVersionInfo()</a> .
<b>Value</b>	0x0BU

#### 5.3.3.1.35. ETHIF\_GETVLANID\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_GetVlanId()</a> .
<b>Value</b>	0x43U

#### 5.3.3.1.36. ETHIF\_INIT\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_Init()</a> .
<b>Value</b>	0x01U

#### 5.3.3.1.37. ETHIF\_INSTANCE\_ID

<b>Purpose</b>	Defines the instance number of this Ethernet Interface. Since multiple instances of Ethernet Interface are not supported the Instance Id is always zero.
<b>Value</b>	0U



#### 5.3.3.1.38. ETHIF\_MAINFUNCTIONRX\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_MainFunctionRx()</a> .
<b>Value</b>	0x20U

#### 5.3.3.1.39. ETHIF\_MAINFUNCTIONSTATE\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_MainFunctionState()</a> .
<b>Value</b>	0x05U

#### 5.3.3.1.40. ETHIF\_MAINFUNCTIONTX\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_MainFunctionTx()</a> .
<b>Value</b>	0x21U

#### 5.3.3.1.41. ETHIF\_PROVIDETXBUFFER\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_ProvideTxBuffer()</a> .
<b>Value</b>	0x09U

#### 5.3.3.1.42. ETHIF\_READMII\_SVCID

<b>Purpose</b>	Defines API ID of function <a href="#">EthIf_ReadMii()</a> .
<b>Value</b>	0x70U

#### 5.3.3.1.43. ETHIF\_RESETCONFIGURATION\_SVCID

<b>Purpose</b>	Defines API ID of function <a href="#">EthIf_ResetConfiguration()</a> .
<b>Value</b>	0x2DU

#### 5.3.3.1.44. ETHIF\_RETRANSMIT\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_Retransmit()</a> .
----------------	---



<b>Value</b>	0xF2U
--------------	-------

#### 5.3.3.1.45. ETHIF\_SETCONTROLLERMODE\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_SetControllerMode()</a> .
<b>Value</b>	0x03U

#### 5.3.3.1.46. ETHIF\_SETCORRECTIONTIME\_SVCID

<b>Purpose</b>	Defines API ID of function <a href="#">EthIf_SetCorrectionTime()</a> .
<b>Value</b>	0x26U

#### 5.3.3.1.47. ETHIF\_SETFORWARDINGMODE\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_SetForwardingMode()</a> .
<b>Value</b>	0x41U

#### 5.3.3.1.48. ETHIF\_SETGLOBALTIME\_SVCID

<b>Purpose</b>	Defines API ID of function <a href="#">EthIf_SetGlobalTime()</a> .
<b>Value</b>	0x27U

#### 5.3.3.1.49. ETHIF\_SETPHYLOOPBACKMODE\_SVCID

<b>Purpose</b>	Defines API id of function EthIf_SetPhyLoopbackMode().
<b>Value</b>	0x12U

#### 5.3.3.1.50. ETHIF\_SETPHYSADDR\_SVCID

<b>Purpose</b>	Defines API id of function <a href="#">EthIf_SetPhysAddr()</a> .
<b>Value</b>	0x0dU



#### 5.3.3.1.51. ETHIF\_SETPHYTESTMODE\_SVCID

Purpose	Defines API id of function EthIf_SetPhyTestMode().
Value	0x17U

#### 5.3.3.1.52. ETHIF\_SETPHYTXMODE\_SVCID

Purpose	Defines API id of function EthIf_SetPhyTxMode().
Value	0x13U

#### 5.3.3.1.53. ETHIF\_SETTRANSCEIVERWAKEUPMODE\_SVCID

Purpose	Defines API id of function <a href="#">EthIf_SetTransceiverWakeMode()</a> .
Value	0x2EU

#### 5.3.3.1.54. ETHIF\_STARTALLPORTS\_SVCID

Purpose	Defines API id of function <a href="#">EthIf_StartAllPorts()</a> .
Value	0x07U

#### 5.3.3.1.55. ETHIF\_STORECONFIGURATION\_SVCID

Purpose	Defines API ID of function <a href="#">EthIf_StoreConfiguration()</a> .
Value	0x2CU

#### 5.3.3.1.56. ETHIF\_SWITCHMGMTINFOINDICATION\_SVCID

Purpose	Defines API id of function <a href="#">EthIf_SwitchMgmtInfoIndication()</a> .
Value	0x3AU

#### 5.3.3.1.57. ETHIF\_SWITCHPORTGROUPREQUESTMODE\_SVCID

Purpose	Defines API id of function <a href="#">EthIf_SwitchPortGroupRequestMode()</a> .
Value	0x06U



#### 5.3.3.1.58. ETHIF\_SWTGETCOUNTERVALUES\_SVCID

Purpose	Defines API ID of function <a href="#">EthIf_SwtGetCounterValues()</a> .
Value	0x40U

#### 5.3.3.1.59. ETHIF\_TRANSMIT\_SVCID

Purpose	Defines API id of function <a href="#">EthIf_Transmit()</a> .
Value	0x0AU

#### 5.3.3.1.60. ETHIF\_TRCVMODEINDICATION\_SVCID

Purpose	Defines API id of function <a href="#">EthIf_TrcvModeIndication()</a> .
Value	0x0FU

#### 5.3.3.1.61. ETHIF\_UPDATEPHYSADDRFILTER\_SVCID

Purpose	Defines API id of function <a href="#">EthIf_UpdatePhysAddrFilter()</a> .
Value	0x0cU

#### 5.3.3.1.62. ETHIF\_VERIFYCONFIG\_SVCID

Purpose	Defines API id of function <a href="#">EthIf_VerifyConfig()</a> .
Value	0x40U

#### 5.3.3.1.63. ETHIF\_WRITEMII\_SVCID

Purpose	Defines API ID of function <a href="#">EthIf_WriteMii()</a> .
Value	0x71U

#### 5.3.3.1.64. TS\_RELOCATABLE\_CFG\_ENABLE

Purpose	Enable/disable relocateable config.
---------	-------------------------------------



<b>Value</b>	ETHIF_RELOCATABLE_CFG_ENABLE
--------------	------------------------------

### 5.3.3.2. Functions

#### 5.3.3.2.1. EthIf\_Cbk\_RxIndication

<b>Purpose</b>	calls EthIf_RxIndication	
<b>Synopsis</b>	<code>void EthIf_Cbk_RxIndication ( uint8 CtrlIdx , Eth_DataType * DataPtr , uint16 LenByte );</code>	
<b>Service ID</b>	0x10	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the EthIf.
	DataPtr	Address of the received payload (including ethernet header).
	LenByte	Length of the payload contained in the received rx buffer.
<b>Description</b>	This service directly calls EthIf_RxIndication.	

#### 5.3.3.2.2. EthIf\_Cbk\_TxConfirmation

<b>Purpose</b>	Tx-Confirmation callback function.	
<b>Synopsis</b>	<code>void EthIf_Cbk_TxConfirmation ( uint8 CtrlIdx , Eth_BufIdxType BufIdx );</code>	
<b>Service ID</b>	0x11	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the Ethernet controller within the context of the Ethernet Interface.
	BufIdx	Index of the transmitted buffer.
<b>Description</b>	This service Confirms the transmission of an Ethernet frame.	



### 5.3.3.2.3. EthIf\_CheckWakeup

<b>Purpose</b>	Check a wake up source.	
<b>Synopsis</b>	<code>Std_ReturnType EthIf_CheckWakeup ( EcuM_WakeupSourceType WakeupSource );</code>	
<b>Parameters (in)</b>	WakeupSource	Source (transceiver) which initiated the wake up event.
<b>Return Value</b>	Std_ReturnType	
	E_OK	The function has been successfully executed.
	E_NOT_OK	The function could not be successfully executed.
<b>Description</b>	This service is called by integration code to check a wake up source.	

### 5.3.3.2.4. EthIf\_ClearSwitchPortSignalQuality

<b>Purpose</b>	Clears stored signal quality.	
<b>Synopsis</b>	<code>Std_ReturnType EthIf_ClearSwitchPortSignalQuality ( uint8 SwitchIdx , uint8 SwitchPortIdx );</code>	
<b>Service ID</b>	none	
<b>Sync/Async</b>	synchronous	
<b>Reentrancy</b>	Reentrant for different SwitchIdx and SwitchPortIdx. Non reentrant for the same SwitchIdx and SwitchPortIdx.	
<b>Parameters (in)</b>	SwitchIdx	Switch index within the context of the Ethernet Interface.
	SwitchPortIdx	Switch port index within the context of the Ethernet Interface.
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request has been accepted.
	E_NOT_OK	The request has not been accepted.
<b>Description</b>	This function clears the stored signal quality of the link for the indexed switch port.	

### 5.3.3.2.5. EthIf\_CtrlModeIndication

<b>Purpose</b>	Callback function to indicate controller mode change.
----------------	---



<b>Synopsis</b>	<code>void EthIf_CtrlModeIndication ( uint8 CtrlIdx , Eth_ModeType CtrlMode );</code>	
<b>Service ID</b>	0x0E	
<b>Sync/Async</b>	Synchronous	
<b>Parameters (in)</b>	CtrlIdx	Index of the physical Ethernet controller within the context of the Ethernet Interface.
	CtrlMode	Notified Ethernet controller mode: ETH_MODE_DOWN ETH_MODE_ACTIVE.
<b>Description</b>	<p>Called asynchronously when mode has been read out. Triggered by previous Eth_SetControllerMode() call. Can be called directly within the trigger functions.</p> <p>{Non Reentrant for the same CtrlIdx, reentrant for different}</p>	

### 5.3.3.2.6. EthIf\_DisableRelatedEthIfCtrls

<b>Purpose</b>	Within a group of EthIfControllers that reference the same PhyController, disable all but one received as a parameter.	
<b>Synopsis</b>	<code>Std_ReturnType EthIf_DisableRelatedEthIfCtrls ( uint8 CtrlIdx );</code>	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the EthIf.
<b>Return Value</b>	Std_ReturnType	
	E_OK	
	E_NOT_OK	{}

### 5.3.3.2.7. EthIf\_EnableEgressTimeStamp

<b>Purpose</b>	Service to enable egress time stamp.	
<b>Synopsis</b>	<code>void EthIf_EnableEgressTimeStamp ( uint8 CtrlIdx , Eth_BufIdx- Type BufIdx );</code>	
<b>Service ID</b>	0x23	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	CtrlIdx	- EthIf controller index.
	BufIdx	- Index of the used transmit buffer.



### 5.3.3.2.8. EthIf\_EnableRelatedEthIfCtrls

<b>Purpose</b>	Within a group of EthIfControllers that reference the same PhyController, enable all but one received as a parameter.	
<b>Synopsis</b>	<code>Std_ReturnType EthIf_EnableRelatedEthIfCtrls ( uint8 CtrlIdx );</code>	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the EthIf.
<b>Return Value</b>	Std_ReturnType	
	E_OK	
	E_NOT_OK	{}

### 5.3.3.2.9. EthIf\_GetAndResetMeasurementData

<b>Purpose</b>	Reads and resets measurement data.	
<b>Synopsis</b>	<code>Std_ReturnType EthIf_GetAndResetMeasurementData ( EthIf_MeasurementIdxType MeasurementIdx , boolean MeasurementResetNeeded , uint32 * MeasurementDataPtr );</code>	
<b>Parameters (in)</b>	MeasurementIdx	Index to select specific measurement data: ETHIF_MEAS_DROP_CRTLIDX (0x01) - Measurement index of dropped datagrams caused by invalid CtrlIdx/VLAN. ETHIF_MEAS_RESERVED_1 (0x02-0x7F) - Reserved by AUTOSAR. ETHIF_MEAS_RESERVED_2 (0x80-0xEF) - Vendor specific range. ETHIF_MEAS_RESERVED_3 (0xF0-0xFE) - Reserved by AUTOSAR (future use). ETHIF_MEAS_ALL (0xFF) - Represents all measurement indexes.
	MeasurementResetNeeded	Flag to trigger a reset of the measurement data.
<b>Parameters (out)</b>	MeasurementDataPtr	Pointer to data buffer, where to copy measurement data.
<b>Return Value</b>	Std_ReturnType	
	E_OK	The function has been successfully executed.
	E_NOT_OK	The function could not be successfully executed.



<b>Description</b>	This service allows to read and reset detailed measurement data for diagnostic purposes. Get all MeasurementIdx's at once is not supported. ETHIF_MEAS_ALL shall only be used to reset all MeasurementIdx's at once. A NULL_PTR shall be provided for MeasurementDataPtr in this case.
--------------------	--

### 5.3.3.2.10. EthIf\_GetArlTable

<b>Purpose</b>	Service to obtain the address resolution table of a switch.	
<b>Synopsis</b>	Std_ReturnType <b>EthIf_GetArlTable</b> ( uint8 SwitchIdx , uint16 * numberOfElements , Eth_MacVlanType * arlTableListPointer );	
<b>Service ID</b>	0x29	
<b>Parameters (in)</b>	SwitchIdx	- Index of the switch within the context of the Ethernet Switch Driver
	inout]	numberOfElements - in: Maximum number of elements which can be written into the arlTable. out: Number of elements which are currently available in the EthSwitch module.
<b>Parameters (in,out)</b>	inout]	numberOfElements - in: Maximum number of elements which can be written into the arlTable. out: Number of elements which are currently available in the EthSwitch module.
<b>Parameters (out)</b>	ArlTable	- Returns the ARL table of the switch consisting of a list of structs with MAC-address, VLAN-ID and port.
<b>Return Value</b>		

### 5.3.3.2.11. EthIf\_GetBufferLevel

<b>Purpose</b>	Service to read the buffer level of the corresponding switch. Whether this buffer level is one value for the entire switch (shared memory) or one value for each port at a switch is technology dependent.
<b>Synopsis</b>	Std_ReturnType <b>EthIf_GetBufferLevel</b> ( uint8 SwitchIdx , uint32 * SwitchBufferLevelPtr );
<b>Service ID</b>	0x2A



<b>Parameters (in)</b>	SwitchIdx	- Index of the switch within the context of the Ethernet Switch Driver
<b>Parameters (out)</b>	SwitchBufferLevelPtr	- The interpretation of this value is switch dependent
<b>Return Value</b>		

#### 5.3.3.2.12. EthIf\_GetControllerMode

<b>Purpose</b>	Gets the Ethernet controller mode.	
<b>Synopsis</b>	<code>Std_ReturnType EthIf_GetControllerMode ( uint8 CtrlIdx , Eth_-ModeType * CtrlModePtr );</code>	
<b>Service ID</b>	0x04	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the EthIf.
<b>Parameters (out)</b>	CtrlModePtr	Address to set the current controller mode to.
<b>Return Value</b>	<code>E_OK</code> : Service execution was successful. <code>E_NOT_OK</code> : Service execution failed.	
<b>Description</b>	This service gets the Ethernet controller mode .	

#### 5.3.3.2.13. EthIf\_GetCtrlIdxList

<b>Purpose</b>	Gets active Ethernet controllers.	
<b>Synopsis</b>	<code>Std_ReturnType EthIf_GetCtrlIdxList ( uint8 * NumberOfCtrlIdx , uint8 * CtrlIdxListPtr );</code>	
<b>Parameters (in)</b>	inout]	NumberOfCtrlIdx in: Maximum number of controllers in CtrlIdxListPtr, 0 to return the number of controllers but without filling CtrlIdxListPtr. out: Number of active controllers.
<b>Parameters (in,out)</b>	inout]	NumberOfCtrlIdx in: Maximum number of controllers in CtrlIdxListPtr, 0 to return the number of controllers but without filling CtrlIdxListPtr. out: Number of active controllers.



<b>Parameters (out)</b>	CtrlIdxListPtr	Pointer to a list of active controller indexes.
<b>Return Value</b>	Std_ReturnType	
	E_OK	The function has been successfully executed.
	E_NOT_OK	The function could not be successfully executed.
<b>Description</b>	This service returns the number and index of all active Ethernet controllers.	

### 5.3.3.2.14. EthIf\_GetCurrentTime

<b>Purpose</b>	Service to get current hardware time.	
<b>Synopsis</b>	Std_ReturnType <b>EthIf_GetCurrentTime</b> ( uint8 CtrlIdx , Eth_TimeStampQualType * timeQualPtr , Eth_TimeStampType * timeStampPtr );	
<b>Service ID</b>	0x22	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	CtrlIdx	- EthIf controller index.
<b>Parameters (out)</b>	timeQualPtr	- Pointer containing the quality of the hardware time stamp.
	timeStampPtr	- Pointer containing the current time.
<b>Return Value</b>	E_OK - Service execution was successful. E_NOT_OK - Service execution failed.	

### 5.3.3.2.15. EthIf\_GetEgressTimeStamp

<b>Purpose</b>	Service to get egress time stamp.	
<b>Synopsis</b>	Std_ReturnType <b>EthIf_GetEgressTimeStamp</b> ( uint8 CtrlIdx , Eth_BufIdxType BufIdx , Eth_TimeStampQualType * timeQualPtr , Eth_TimeStampType * timeStampPtr );	
<b>Parameters (in)</b>	CtrlIdx	- EthIf controller index.
	BufIdx	- Index of the used transmit buffer.
<b>Parameters (out)</b>	timeQualPtr	- Pointer containing the quality of the hardware time stamp.



	timeStampPtr	- Pointer containing the egress time stamp.
<b>Return Value</b>	Std_ReturnType	
	E_OK	Operation was successful
	E_NOT_OK	Operation was not successful

### 5.3.3.2.16. EthIf\_GetIngressTimeStamp

<b>Purpose</b>	Service to get ingress time stamp.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>EthIf_GetIngressTimeStamp</b> ( uint8 CtrlIdx , Eth_DataType * DataPtr , Eth_TimeStampQualType * timeQualPtr , Eth_TimeStampType * timeStampPtr );</pre>	
<b>Parameters (in)</b>	CtrlIdx	- EthIf controller index.
	DataPtr	- Pointer to Rx data buffer of the requested frame.
<b>Parameters (out)</b>	timeQualPtr	- Pointer containing the quality of the hardware time stamp.
	timeStampPtr	- Pointer containing the ingress time stamp.
<b>Return Value</b>	Std_ReturnType	
	E_OK	Operation was successful
	E_NOT_OK	Operation was not successful

### 5.3.3.2.17. EthIf\_GetPhysAddr

<b>Purpose</b>	Gets the local Ethernet physical address.	
<b>Synopsis</b>	<pre>void <b>EthIf_GetPhysAddr</b> ( uint8 CtrlIdx , uint8 * PhysAddrPtr );</pre>	
<b>Service ID</b>	0x08	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller the transceiver is connected to. within the context of the EthIf.
<b>Parameters (out)</b>	PhysAddrPtr	Address to write the physical address to.
<b>Description</b>	This service gets the local Ethernet physical device address.	



### 5.3.3.2.18. EthIf\_GetPortMacAddr

<b>Purpose</b>	Service to obtain the port over which this MAC-address can be reached.	
<b>Synopsis</b>	<code>Std_ReturnType EthIf_GetPortMacAddr ( const uint8 * MacAddrPtr , uint8 * SwitchIdxPtr , uint8 * PortIdxPtr );</code>	
<b>Service ID</b>	0x28	
<b>Parameters (in)</b>	MacAddrPtr	- MAC-address for which a switch port is searched over which the node with this MAC-address can be reached.
<b>Parameters (out)</b>	SwitchIdxPtr	- Pointer to the switch index
	PortIdxPtr	- Pointer to the port index
<b>Return Value</b>		

### 5.3.3.2.19. EthIf\_GetSwitchPortSignalQuality

<b>Purpose</b>	Obtains the switch port signal quality.	
<b>Synopsis</b>	<code>Std_ReturnType EthIf_GetSwitchPortSignalQuality ( uint8 SwitchIdx , uint8 SwitchPortIdx , EthIf_SignalQualityResultType * ResultPtr );</code>	
<b>Service ID</b>	none	
<b>Sync/Async</b>	synchronous	
<b>Reentrancy</b>	Reentrant for different SwitchIdx and SwitchPortIdx. Non reentrant for the same SwitchIdx and SwitchPortIdx.	
<b>Parameters (in)</b>	SwitchIdx	Switch index within the context of the Ethernet Interface.
	SwitchPortIdx	Switch port index within the context of the Ethernet Interface.
<b>Parameters (out)</b>	ResultPtr	Pointer to the memory where the signal quality shall be stored.
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request has been accepted.
	E_NOT_OK	The request has not been accepted.
<b>Description</b>	This function obtains the signal quality of the link for the indexed switch port.	



### 5.3.3.2.20. EthIf\_GetTransceiverWakeupMode

<b>Purpose</b>	Gets the transceiver wake up mode.	
<b>Synopsis</b>	<code>Std_ReturnType EthIf_GetTransceiverWakeupMode ( uint8 TrcvIdx , EthTrcv_WakeupModeType * TrcvWakeupModePtr );</code>	
<b>Parameters (in)</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Interface.
<b>Parameters (out)</b>	TrcvWakeupModePtr	Pointer where transceiver wake up mode will be written: ETHTRCV_WUM_DISABLE - Transceiver wake up is disabled ETHTRCV_WUM_ENABLE - Transceiver wake up is enabled.
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	Transceiver wake up mode could not be obtained.
<b>Description</b>	This service returns the wake up mode of the indexed transceiver.	

### 5.3.3.2.21. EthIf\_GetVersionInfo

<b>Purpose</b>	Get version information of the Ethernet Interface.	
<b>Synopsis</b>	<code>void EthIf_GetVersionInfo ( Std_VersionInfoType * VersionInfoPtr );</code>	
<b>Service ID</b>	0x0b	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (out)</b>	VersioninfoPtr	Pointer where to store the version information of this module.
<b>Description</b>	This service returns the version information of this module. The version information includes: <ul style="list-style-type: none"> <li>▶ Module Id</li> <li>▶ Vendor Id</li> <li>▶ Vendor specific version numbers</li> </ul>	



### 5.3.3.2.22. EthIf\_GetVlanId

<b>Purpose</b>	Gets Ethernet controller VLAN identifier.	
<b>Synopsis</b>	<code>Std_ReturnType EthIf_GetVlanId ( uint8 CtrlIdx , uint16 * VlanIdPtr );</code>	
<b>Parameters (in)</b>	CtrlIdx	Index of the Ethernet controller within the context of the Ethernet Interface.
<b>Parameters (out)</b>	VlanIdPtr	Pointer to store the VLAN identifier (VID) of the Ethernet controller. 0 if the the Ethernet controller represents no virtual network (VLAN).
<b>Return Value</b>	Std_ReturnType	
	E_OK	The function has been successfully executed.
	E_NOT_OK	The function could not be successfully executed.
<b>Description</b>	This service returns the VLAN identifier of the requested Ethernet controller.	

### 5.3.3.2.23. EthIf\_Init

<b>Purpose</b>	Initializes the EthIf module.	
<b>Synopsis</b>	<code>void EthIf_Init ( const EthIf_ConfigType * CfgPtr );</code>	
<b>Service ID</b>	0x01	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CfgPtr	Address of the post-build configuration data structure.
<b>Description</b>	This service initializes the EthIf module. It shall be the first function of the module which is called.	

### 5.3.3.2.24. EthIf\_IsValidConfig

<b>Purpose</b>	Checks compatibility of the post-build-time configuration.
----------------	--



<b>Synopsis</b>	<code>Std_ReturnType EthIf_IsValidConfig ( const void * voidConfigPtr );</code>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	<code>voidConfigPtr</code>	Pointer to the configuration data of the EthIf module.
<b>Return Value</b>	Result of compatibility check	
	<code>E_OK</code>	Provided configuration is compatible.
	<code>E_NOT_OK</code>	Provided configuration is not compatible.
<b>Description</b>	This service checks the compatibility of the post-build-time configuration against the source code.	

### 5.3.3.2.25. EthIf\_MainFunctionRx

<b>Purpose</b>	Main function for rx-tasks.
<b>Synopsis</b>	<code>void EthIf_MainFunctionRx ( void );</code>
<b>Service ID</b>	0x20
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Description</b>	This service performs the periodic tasks for Ethernet reception (polling the Ethernet driver).

### 5.3.3.2.26. EthIf\_MainFunctionRxPrio

<b>Purpose</b>	Main function for priority receive processing tasks.
<b>Synopsis</b>	<code>void EthIf_MainFunctionRxPrio ( uint8 PCtrlIngrFifoIdx );</code>
<b>Service ID</b>	0xXX
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Description</b>	This service performs the periodic tasks for ethernet reception (polling the ethernet driver) for the priority frames only. This function should be called via the related function <code>EthIf_MainFunctionRx_&lt;EthIfCtr&gt;_Prio()</code> .



### 5.3.3.2.27. EthIf\_MainFunctionState

<b>Purpose</b>	Function to update mode state and transceiver link state of EthIf objects.
<b>Synopsis</b>	<code>void EthIf_MainFunctionState ( void );</code>
<b>Service ID</b>	0x05
<b>Sync/Async</b>	Asynchronous
<b>Reentrancy</b>	Non Reentrant
<b>Description</b>	Function is polling the link state of the used communication hardware (Ethernet transceiver, Ethernet switch ports). For active objects it reads-out / calculates transceiver link state and reports it to upper layers (State Manager or BswM).

### 5.3.3.2.28. EthIf\_MainFunctionTx

<b>Purpose</b>	Main function for tx-tasks.
<b>Synopsis</b>	<code>void EthIf_MainFunctionTx ( void );</code>
<b>Service ID</b>	0x21
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Description</b>	This service contains currently no functionality.

### 5.3.3.2.29. EthIf\_ProvideTxBuffer

<b>Purpose</b>	Provide an Ethernet Tx buffer.	
<b>Synopsis</b>	<code>BufReq_ReturnType EthIf_ProvideTxBuffer ( uint8 CtrlIdx , Eth_FrameType FrameType , uint8 Priority , Eth_BufIdxType * BufIdxPtr , EthIf_Uint8TypePtr * BufPtr , uint16 * LenBytePtr );</code>	
<b>Service ID</b>	0x09	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the EthIf.



	FrameType	Ethernet type of the related payload for the transmission.
	Priority	Priority of the related payload for transmission (used for VLAN only).
<b>Parameters (in,out)</b>	LenBytePtr	Address that holds the requested buffer length and after return of the call it holds the actual size of the returned Tx buffer.
<b>Parameters (out)</b>	BufIdxPtr	Address an index to identify the returned Tx buffer is stored to.
	BufPtr	Address of a pointer where the address of the returned Tx buffer is stored to.
<b>Return Value</b>	BUFREQ_OK: Service execution was successful. BUFREQ_E_NOT_OK: Service execution failed. BUFREQ_E_BUSY: Service currently not available.	
<b>Description</b>	This service returns an Ethernet driver Tx buffer with the required length.	

### 5.3.3.2.30. EthIf\_ReadMii

<b>Purpose</b>	Service to read from an Ethernet transceiver register.	
<b>Synopsis</b>	Std_ReturnType <b>EthIf_ReadMii</b> ( uint8 CtrlIdx , uint8 RegIdx , uint16 * RegValPtr );	
<b>Service ID</b>	0x70	
<b>Parameters (in)</b>	CtrlIdx	- EthIf controller index.
	RegIdx	- Register index of the TrcvIdx assigned to the CtrlIdx.
<b>Parameters (out)</b>	RegValPtr	- Pointer containing the value of the RegIdx .
<b>Return Value</b>		

### 5.3.3.2.31. EthIf\_ResetConfiguration

<b>Purpose</b>	Service to reset the configuration of the learned MAC/Port tables of a switch in a persistent manner and will be used by e.g. CDD. The statically configured entries shall still remain.
<b>Synopsis</b>	Std_ReturnType <b>EthIf_ResetConfiguration</b> ( uint8 SwitchIdx );
<b>Service ID</b>	0x2D



<b>Parameters (in)</b>	SwitchIdx	- Index of the switch within the context of the Ethernet Switch Driver
<b>Return Value</b>		

### 5.3.3.2.32. EthIf\_Retransmit

<b>Purpose</b>	Retransmit a received/transmitted frame.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>EthIf_Retransmit</b> ( uint8 CtrlIdx , uint8 OrigCtrlIdx , Eth_BufIdxType * BufIdxPtr , Eth_DataType * DataPtr , Eth_FrameType FrameType , uint16 LenByte , const Eth_RetransmitInfoType *const RetransmitInfoPtr );</pre>	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller where frame shall be re-transmitted within the context of the EthIf.
	OrigCtrlIdx	Index of the controller of the provided buffer within the context of the EthIf.
	DataPtr	Buffer pointer of retransmitted frame. It points before the VLAN portion of the frame.
	FrameType	Ethernet type of the related payload for the transmission.
	LenByte	Length of the payload contained in the tx buffer to transmit.
	RetransmitInfoPtr	Pointer to additional retransmit info. In case of retransmit of an already transmitted buffer, Priority needs to be set to 0xFF.
<b>Parameters (in,out)</b>	BufIdxPtr	Pointer to the Index of the buffer to transmit.
<b>Return Value</b>	<p>E_OK - re-transmission of the buffer was successful</p> <p>E_NOT_OK - if provided CtrlIdx-es refer to not compatible EthIf controllers (e.g. origin EthIf controller uses VLAN and EthIf controller uses no VLAN or vice versa)</p>	
<b>Description</b>	<p>This function retransmits the current buffer. It can be called to retransmit a received buffer (OrigCtrlIdx and DataPtr of current buffer must be provided) and to retransmit a transmitted buffer (OrigCtrlIdx and BufIdxPtr of current buffer must be provided). The current buffer is not released when the related EthIf_RxIndication function or EthIf_TxConfirmation function returns. In both cases the function returns an buffer index (parameter BufIdxPtr).</p>	



### 5.3.3.2.33. EthIf\_RxIndication

<b>Purpose</b>	Rx-Indication function.	
<b>Synopsis</b>	<pre>void <b>EthIf_RxIndication</b> ( uint8 CtrlIdx , Eth_FrameType FrameType , boolean isBroadcast , const uint8 * PhysAddrPtr , Eth_DataType * DataPtr , uint16 LenByte );</pre>	
<b>Service ID</b>	0x10	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the EthIf.
	FrameType	Ethernet Frame type.
	isBroadcast	Indicates whether ethernet destination address is a broadcast address or not.
	PhysAddrPtr	Ethernet source address.
	DataPtr	Address of the received payload.
	LenByte	Length of the payload contained in the received rx buffer.
<b>Description</b>	This service is called by the ethernet driver in case a reception is indicated.	

### 5.3.3.2.34. EthIf\_SetControllerMode

<b>Purpose</b>	Sets the Ethernet controller mode.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>EthIf_SetControllerMode</b> ( uint8 CtrlIdx , Eth_ModeType CtrlMode );</pre>	
<b>Service ID</b>	0x03	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the EthIf.
	CtrlMode	Mode to set the controller to.
<b>Return Value</b>	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
<b>Description</b>	This service sets the Ethernet controller mode.	



### 5.3.3.2.35. EthIf\_SetCorrectionTime

<b>Purpose</b>	Service to perform time correction time stamp.	
<b>Synopsis</b>	<pre>void <b>EthIf_SetCorrectionTime</b> ( uint8 CtrlIdx , const Eth_ TimeIntDiffType * timeOffsetPtr , const Eth_RateRatioType * rateRatioPtr );</pre>	
<b>Service ID</b>	0x26	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	CtrlIdx	- EthIf controller index.
	timeOffsetPtr	- Pointer to adjust hardware time.
	rateRatioPtr	- Pointer to set rate ratio.

### 5.3.3.2.36. EthIf\_SetForwardingMode

<b>Purpose</b>	Sets the frame forwarding mode.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>EthIf_SetForwardingMode</b> ( uint8 SwitchIdx , boolean mode );</pre>	
<b>Parameters (in)</b>	SwitchIdx	Index of the switch within the context of the Eth. Switch Driver.
	mode	Frame forwarding mode: TRUE - Forwarding enabled. FALSE - Forwarding disabled.
<b>Return Value</b>	Std_ReturnType	
	E_OK	Stopping of frame forwarding succeeded
	E_NOT_OK	stopping of frame forwarding not succeeded.
<b>Description</b>	This service sets the frame forwarding mode.	

### 5.3.3.2.37. EthIf\_SetGlobalTime

<b>Purpose</b>	Service to set the global time.
----------------	---------------------------------



<b>Synopsis</b>	<code>Std_ReturnType EthIf_SetGlobalTime ( uint8 CtrlIdx , const Eth_TimeStampType * timeStampPtr );</code>	
<b>Service ID</b>	0x27	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	CtrlIdx	- Ethif controller index.
	timeStampPtr	- Pointer containing the new time.
<b>Return Value</b>		

### 5.3.3.2.38. EthIf\_SetPhysAddr

<b>Purpose</b>	set physical source address of controller.	
<b>Synopsis</b>	<code>void EthIf_SetPhysAddr ( uint8 CtrlIdx , const uint8 * PhysAddrPtr );</code>	
<b>Service ID</b>	0x80	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the Ethif.
	PhysAddrPtr	Pointer to memory containing the physical source address (MAC address) in network byte order.
<b>Description</b>	This service sets the physical source address used by the indexed controller.	

### 5.3.3.2.39. EthIf\_SetTransceiverWakeupMode

<b>Purpose</b>	Sets the transceiver wake up mode.	
<b>Synopsis</b>	<code>Std_ReturnType EthIf_SetTransceiverWakeupMode ( uint8 TrcvIdx , EthTrcv_WakeupModeType TrcvWakeUpMode );</code>	
<b>Parameters (in)</b>	TrcvIdx	Index of the transceiver within the context of the Ethernet Interface
	TrcvWakeUpMode	Transceiver wake up mode: ETHTRCV_WUM_DISABLE - Disable transceiver wake up ETHTRCV_WUM_ENABLE - Enable transceiver wake up



		CV_WUM_CLEAR - Clears transceiver wake up reason
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	Transceiver wake up could not be changed or wake-up reason could not be cleared.
<b>Description</b>	This service enables / disables the wake up mode or clear the wake-up reason of the specified transceiver.	

#### 5.3.3.2.40. EthIf\_StartAllPorts

<b>Purpose</b>	Requests to start all configured ports.	
<b>Synopsis</b>	Std_ReturnType <b>EthIf_StartAllPorts</b> ( void );	
<b>Return Value</b>	Std_ReturnType	
	E_OK	Success.
	E_NOT_OK	Port mode could not be started.
<b>Description</b>	This service requests to start all configured ports.	

#### 5.3.3.2.41. EthIf\_StoreConfiguration

<b>Purpose</b>	Service to store the configuration of the learned MAC/Port tables of a switch in a persistent manner and will be used by e.g. CDD.	
<b>Synopsis</b>	Std_ReturnType <b>EthIf_StoreConfiguration</b> ( uint8 SwitchIdx );	
<b>Service ID</b>	0x2C	
<b>Parameters (in)</b>	SwitchIdx	- Index of the switch within the context of the Ethernet Switch Driver
<b>Return Value</b>		

#### 5.3.3.2.42. EthIf\_SwitchMgmtInfoIndication

<b>Purpose</b>	Callback function to indicate switch management info.
<b>Synopsis</b>	void <b>EthIf_SwitchMgmtInfoIndication</b> ( uint8 CtrlIdx , Eth_DataType * DataPtr , EthSwt_MgmtInfoType * MgmtInfoPtr );



<b>Service ID</b>	0x3A	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of an Ethernet Interface controller.
	DataPtr	Ethernet data pointer where the management information belongs.
<b>Parameters (out)</b>	MgmtInfoPtr	Pointer to Management information.
<b>Description</b>	Ingress Switch management info indication redirected call to upper layers who registered for the call.	

#### 5.3.3.2.43. EthIf\_SwitchPortGroupRequestMode

<b>Purpose</b>	Requests the mode of EthIfSwtPortGroup.	
<b>Synopsis</b>	Std_ReturnType <b>EthIf_SwitchPortGroupRequestMode</b> ( EthIf_SwitchPortGroupIdxType PortGroupIdx , EthTrcv_ModeType PortMode );	
<b>Parameters (in)</b>	PortGroupIdx	Index of the port group within the context of the Eth. Interface.
	PortMode	Requested port mode: ETHTR-CV_MODE_DOWN - Disables the port group. ETHTRCV_MODE_ACTIVE - Enables the port group.
<b>Return Value</b>	Std_ReturnType	
	E_OK	Success.
	E_NOT_OK	Port group mode could not be changed.
<b>Description</b>	This service request a mode for the EthIfSwtPortGroup. The call is forwarded to EthSwt by calling EthSwt_SetSwitchPortMode for all EthSwtPorts referenced by the port group.	

#### 5.3.3.2.44. EthIf\_SwtGetCounterValues

<b>Purpose</b>	Service to read a list with drop counter values of the corresponding switch. The meaning of these values is switch dependent and can include values like 1.) dropped packets due to buffer overrun, 2.) dropped packets due to CRC errors, etc.
<b>Synopsis</b>	Std_ReturnType <b>EthIf_SwtGetCounterValues</b> ( uint8 SwitchIdx , uint8 SwitchPortIdx , Eth_CounterType * CounterPtr );



<b>Service ID</b>	0x40	
<b>Parameters (in)</b>	SwitchIdx	- Index of the switch within the context of the Ethernet Switch Driver
<b>Return Value</b>		

### 5.3.3.2.45. EthIf\_Transmit

<b>Purpose</b>	Transmit a Tx buffer on Ethernet.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>EthIf_Transmit</b> ( uint8 CtrlIdx , Eth_BufIdxType BufIdx , Eth_FrameType FrameType , boolean TxConfirmation , uint16 LenByte , uint8 * PhysAddrPtr );</pre>	
<b>Service ID</b>	0x0a	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx BufIdx FrameType TxConfirmation LenByte PhysAddrPtr	Index of the controller within the context of the EthIf.
		Index of the buffer to transmit.
		Ethernet type of the related payload for the transmission.
		Activates transmission confirmation.
		Length of the payload contained in the tx buffer to transmit.
		Address which holds the destination address the data shall be transmitted to.
<b>Return Value</b>		
<b>Description</b>	This service unlocks a previously provided Tx buffer for transmission.	

### 5.3.3.2.46. EthIf\_TrcvModeIndication

<b>Purpose</b>	Callback function to indicate transceiver mode change.
<b>Synopsis</b>	<pre>void <b>EthIf_TrcvModeIndication</b> ( uint8 TrcvIdx , EthTrcv_ModeType TrcvMode );</pre>
<b>Service ID</b>	0x0F
<b>Sync/Async</b>	Synchronous



<b>Parameters (in)</b>	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Interface
	TrcvMode	Notified Ethernet transceiver mode: ETHTRCV_MODE_DOWN ETHTRCV_MODE_ACTIVE.
<b>Description</b>	Called asynchronously when mode has been read out. Triggered by previous Eth_SetTransceiverMode() call. Can be called directly within the trigger functions.  {Non Reentrant for the same CtrlIdx, reentrant for different}	

### 5.3.3.2.47. EthIf\_TxConfirmation

<b>Purpose</b>	Tx-Confirmation function.	
<b>Synopsis</b>	<pre>void <b>EthIf_TxConfirmation</b> ( uint8 CtrlIdx , Eth_BufIdxType BufIdx , Std_ReturnType Result );</pre>	
<b>Service ID</b>	0x11	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the Ethernet controller within the context of the Ethernet Interface.
	BufIdx	Index of the transmitted buffer.
	Result	E_OK: The transmission was successful, E_NOT_OK: The transmission failed.
<b>Description</b>	This service confirms frame transmission by the indexed controller.	

### 5.3.3.2.48. EthIf\_UpdatePhysAddrFilter

<b>Purpose</b>	Updates physical address filter.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>EthIf_UpdatePhysAddrFilter</b> ( uint8 CtrlIdx , const uint8 * PhysAddrPtr , Eth_FilterActionType Action );</pre>	
<b>Service ID</b>	0x81	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	CtrlIdx	Index of the controller within the context of the EthIf.



	PhysAddrPtr	Pointer to memory containing the physical source address (MAC address) in network byte order.
	Action	Add or remove the address from the Ethernet controllers filter.
<b>Return Value</b>		Std_ReturnType
	E_OK	Filter was successfully changed.
	E_NOT_OK	Filter could not be changed.
<b>Description</b>	This service updates the physical source address to/from the indexed controller filter. If the Ethernet Controller is not capable to do the filtering, the software has to do this.	

### 5.3.3.2.49. EthIf\_VerifyConfig

<b>Purpose</b>	Verifies the Switch Configuration.	
<b>Synopsis</b>	Std_ReturnType <b>EthIf_VerifyConfig</b> ( uint8 SwitchIdx , boolean * Result );	
<b>Parameters (in)</b>	SwitchIdx	Index of the switch within the context of the Eth. Switch Driver.
<b>Parameters (out)</b>	Result	Pointer to the result of verification: TRUE - Configuration verified ok. FALSE - Configuration values found corrupted.
<b>Return Value</b>		Std_ReturnType
	E_OK	Configuration verification succeeded.
	E_NOT_OK	Configuration verification not succeeded.
<b>Description</b>	This service verifies the Switch Configuration depending on the HW-Architecture, HW-capability and the intended accuracy of this verification. Forwarded to EthSwt_VerifyConfig.	

### 5.3.3.2.50. EthIf\_WriteMii

<b>Purpose</b>	Service to write into an Ethernet transceiver register.
<b>Synopsis</b>	Std_ReturnType <b>EthIf_WriteMii</b> ( uint8 CtrlIdx , uint8 RegIdx , uint16 RegVal );
<b>Service ID</b>	0x71

<b>Parameters (in)</b>	CtrlIdx	- EthIf controller index.
	RegIdx	- Register index of the TrcVIdx assigned to the CtrlIdx.
	RegVal	- Value which shall be written to the RegIdx .
<b>Return Value</b>		

## 5.3.4. Integration notes

### 5.3.4.1. Exclusive areas

This section describes the exclusive areas used by the `EthIf` module.

#### 5.3.4.1.1. SCHM\_ETHIF\_EXCLUSIVE\_AREA

<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.

### 5.3.4.2. Production errors

The module does not report any production errors.

### 5.3.4.3. 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

CODE

CONFIG\_DATA\_UNSPECIFIED

CONFIG\_DATA\_8

CONFIG\_DATA\_16

VAR\_INIT\_8

VAR\_INIT\_32

VAR\_INIT\_UNSPECIFIED

VAR\_CLEARED\_8

VAR\_CLEARED\_UNSPECIFIED

CONST\_32

CODE

#### 5.3.4.4. 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.

##### 5.3.4.4.1. lim.EthIf.EB\_INTREQ\_EthIf\_0001

<b>Description</b>	EthIf must not use receive or transmit interrupts. The Integrator must assure that neither EthIf uses interrupts nor function EthIf_MainFunctionRx is called in interrupt context.
<b>Rationale</b>	

##### 5.3.4.4.2. lim.EthIf.EB\_INTREQ\_EthIf\_0002

<b>Description</b>	The reinitialization process shall not interrupt other module functions. If reinitialization of the module is required, the call of EthIf_Init() shall not interrupt other module functions.
<b>Rationale</b>	The reinitialisation process will reset all internal variables. Continuing an interrupted module function after reinitialization can lead to undefined module behavior.



#### 5.3.4.4.3. lim.EthIf.EB\_INTREQ\_EthIf\_0003

<b>Description</b>	The call of EthIf_GetControllerMode() will always result in the call of Eth_GetControllerMode(). The returned mode is not processed within EthIf. In case of an erroneous behaviour of the hardware (controller return another value as previously set by upper layer), the upper layer shall take care to set EthIf to the new mode by calling the respective APIs.
<b>Rationale</b>	It is not specified which layer will call EthIf_GetControllerMode() and for what reason since the upper layer already set the mode.

#### 5.3.4.4.4. lim.EthIf.EB\_INTREQ\_EthIf\_0004

<b>Description</b>	EthIf_Init() shall not be preempted by any other module API calls. It needs to be ensured that the function call EthIf_Init() is not preempted by any other module API calls.
<b>Rationale</b>	During the call of EthIf_Init() global variables and pointers get initialized. It is easy for the integrator to avoid this preemption, thus no data protection mechanism has been implemented for function EthIf_Init().

#### 5.3.4.4.5. lim.EthIf.EB\_INTREQ\_EthIf\_0005

<b>Description</b>	The Integrator must assure that different EthIf_MainFunction-Rx_PrioProcessing[name]() functions (for prioritized reception of traffic) as well as the EthIf_MainFunctionRx() do not preempt each other.
<b>Rationale</b>	

#### 5.3.4.4.6. lim.EthIf.EB\_INTREQ\_EthIf\_0006

<b>Description</b>	Config parameter EthIfInitControllersTransceivers shall be enabled only for Eth drivers and transceivers below ASR 4.2.2. Functions Eth_ControllerInit() and EthTrcv_TransceiverInit() shall be called by integration code for EB Eth drivers and transceivers ASR 4.2.2 or higher.
<b>Rationale</b>	

#### 5.3.4.4.7. lim.EthIf.EB\_INTREQ\_EthIf\_0007

<b>Description</b>	If EB Eth drivers and transceivers ASR 4.3.0 are used, then config parameter EthIfSupportEthAPI needs to be set to ASR430_EB. A compilation error stating a syn-
--------------------	--



	tax issue with EthIf_TxConfirmation() is an indication that EthIfSupportEthAPI is not properly configured.
<b>Rationale</b>	

#### 5.3.4.4.8. lim.EthIf.EB\_INTREQ\_EthIf\_0008

<b>Description</b>	If EthIfSwitch and/or EthIfSwitchPortGroup elements are configured, then EthTrcv configuration parameter: EthTrcvGeneral/EthTrcvGetLinkStateApi must be set and at least one of EthSwtPorts, configured in EthSwt/EthSwtConfig/*/EthSwtPort must have an EthSwtPortTrcvRef defined.
<b>Rationale</b>	

#### 5.3.4.4.9. lim.EthIf.EB\_INTREQ\_EthIf\_0009

<b>Description</b>	Locking per switch port for all switch APIs prevents a preemptive access to the same Trcv registers in the chip. However, different switch ports may access e.g. same SPI. Preemption of SPI can not be solved by EthIf since it has no knowledge about the topology of the connected transceivers. This is out of scope of EthIf and should be solved in SPI. It shall be ensured that same Trcv instance (registers) is not shared between switch ports and configured transceivers. Each Trcv instance shall be used only once, either for a dedicated switch port or a configured Trcv in EthIf configuration.
<b>Rationale</b>	

#### 5.3.4.4.10. lim.EthIf.EB\_INTREQ\_EthIf\_0010

<b>Description</b>	If config parameter EthIfRetransmitApiEnable is enabled, DataPtr parameter used in EthIf_Retransmit API shall provide space for VLAN header in front of the used buffer.
<b>Rationale</b>	

#### 5.3.4.4.11. lim.EthIf.EB\_INTREQ\_EthIf\_0011

<b>Description</b>	If Eth drivers of version below ASR 4.3.0 are used, then EthCtrlIdx of all referenced EthCtrl must be 0-based and consecutive.
<b>Rationale</b>	



#### 5.3.4.4.12. lim.EthIf.EB\_INTREQ\_EthIf\_0012

<b>Description</b>	If support of drivers that contain vendorId and vendorApiInfix is required, then every driver which requires it (Eth, EthTrcv or EthSwt) needs to have a valid BswModuleDescription which shall be referenced from EthIf to provide it with VendorId and VendorApiInfix. Support is not available for ASR versions below 4.3.0.
<b>Rationale</b>	

#### 5.3.4.4.13. lim.EthIf.EB\_INTREQ\_EthIf\_0013

<b>Description</b>	Port group mode and link state are related in a way that when mode is set to DOWN, link state will be reported LINK_STATE_DOWN although ports, when used by another active port group, could still be ACTIVE and consequently accumulated port group link state ACTIVE. In case when port group is under control of BswM, shut-down of ports when port group mode is set to DOWN could be delayed as specified in [SWS_EthIf_00270] and [SWS_EthIf_00271]. During this time period, EthIf accumulates and reports the actual port group link state to BswM. When delay timer expires EthIf will report port group link state LINK_STATE_DOWN as described in previous paragraph. Integrator shall be aware of this behaviour, verify and if necessary adjust the related project specific BswM logic.
<b>Rationale</b>	

## 5.4. EthSM

### 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="#">EthSMConfig</a>	1..1	This container is used to provide a name to the configuration of the AUTOSAR EthSM module.
<a href="#">EthSMGeneral</a>	1..1	This container contains the global parameter of the Ethernet State Manager.

**Containers included**

<a href="#">ReportToDem</a>	1..1	<b>Label:</b> Production error handling Production error handling
<a href="#">EthSMDefensiveProgramming</a>	1..1	<b>Label:</b> Defensive Programming Options  Parameters for defensive programming
<a href="#">EthSMNetwork</a>	1..255	This container contains the Ethernet network-specific parameters of each Ethernet network.  It also contains the controller and transceiver IDs assigned to a Ethernet network.
<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

Parameter Name	<b>IMPLEMENTATION_CONFIG_VARIANT</b>
Label	Config Variant
Multiplicity	1..1
Type	ENUMERATION
Default value	VariantPostBuild
Range	VariantPostBuild

**5.4.1.1. CommonPublishedInformation****Parameters included**

Parameter name	Multiplicity
<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



#### Parameters included

<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>	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>	3
<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>	6
<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>	14
<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>	143
<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. EthSMConfig

#### 5.4.1.3. EthSMGeneral

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



#### Parameters included

<a href="#">EthSMVersionInfoApi</a>	1..1
<a href="#">EthSMSingleNetworkOptEnable</a>	1..1
<a href="#">EthSMMaxNetworks</a>	1..1
<a href="#">EthSMMultiCoreSupport</a>	1..1
<a href="#">EthSMDevAuthSupport</a>	1..1
<a href="#">EthSMRelocatablePbcfgEnable</a>	1..1

#### Parameter Name

**EthSMDevErrorDetect**

<b>Description</b>	Enables and disables the development error detection and notification mechanism.
<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

#### Parameter Name

**EthSMDummyMode**

<b>Description</b>	This configuration parameter is not used. Feature is not supported.  Disables the API to the EthIf.  The API to the ComM is available but the functionality is deactivated.  The function calls from the ComM will be answered with the return value E_OK.
<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

#### Parameter Name

**EthSMMainFunctionPeriod**

<b>Description</b>	Specifies the period in seconds that the MainFunction has to be triggered with.
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.01



<b>Range</b>	<=1 >=0.005	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>EthSMVersionInfoApi</b>	
<b>Description</b>	Enables and disables the version info API.	
<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>EthSMSingleNetworkOptEnable</b>	
<b>Description</b>	Optimizes code for the purpose of using a single Ethernet network only. <ul style="list-style-type: none"> <li>▶ True: The EthSM supports a single Ethernet network only.</li> <li>▶ False: The EthSM supports multiple Ethernet networks.</li> </ul> <b>Optimization Effect:</b> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (code):</b> Enabling this parameter reduces the ROM consumption of the module code.</li> <li>▶ <b>Execution time reduction (code):</b> Enabling this parameter reduces the execution time of the module code.</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>EthSMMaxNetworks</b>	
<b>Description</b>	Number of Ethernet networks supported by the EthSM.	
<b>Multiplicity</b>	1..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>EthSMMultiCoreSupport</b>	
<b>Description</b>	Enables module Multi core support.	
<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>EthSMDevAuthSupport</b>	
<b>Description</b>	Enables module Device Authentication support.	
<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>EthSMRelocatablePbcfgEnable</b>	
<b>Description</b>	Enables/disable support for relocatable postbuild configuration.  ▶ True: Postbuild configuration relocatable in memory. ▶ False: Postbuild configuration not relocatable in memory.	
<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	

#### 5.4.1.4. ReportToDem

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>



#### Parameters included

<a href="#">EthSMDemCtrlTestResultReportToDem</a>	1..1
<a href="#">EthSMDemCtrlTestResultReportToDemDetErrorId</a>	1..1

<b>Parameter Name</b>	<b>EthSMDemCtrlTestResultReportToDem</b>	
<b>Label</b>	EthSMDemCtrlTestResult report to	
<b>Description</b>	<p>Selects the handling of the production error EthSMDemCtrlTestResult.</p> <ul style="list-style-type: none"> <li>▶ DEM: The error is reported to the Diagnostics Event Manager (Dem).</li> <li>▶ DET: The error is reported to the Development Error Tracer (Det) if enabled.</li> <li>▶ DISABLE: The error is not reported at all.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (code):</b> Setting this parameter to a value of DISABLE reduces the ROM consumption of the module code.</li> <li>▶ <b>Execution time reduction (code):</b> Setting this parameter to a value of DISABLE reduces the execution time of the module code.</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>EthSMDemCtrlTestResultReportToDemDetErrorId</b>	
<b>Label</b>	EthSMDemCtrlTestResult Dem To Det error ID	
<b>Description</b>	If a production error is reported towards the Det, EthSMDemCtrlTestResultReportToDemDetErrorId defines the error ID which is reported towards the Det.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	129	
<b>Range</b>	<=255 >=0	



<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

#### 5.4.1.5. EthSMDefensiveProgramming

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">EthSMDefProgEnabled</a>	1..1
<a href="#">EthSMPrecondAssertEnabled</a>	1..1
<a href="#">EthSMPostcondAssertEnabled</a>	1..1
<a href="#">EthSMStaticAssertEnabled</a>	1..1
<a href="#">EthSMUnreachAssertEnabled</a>	1..1
<a href="#">EthSMInvariantAssertEnabled</a>	1..1

<b>Parameter Name</b>	<b>EthSMDefProgEnabled</b>
<b>Label</b>	Enable Defensive Programming
<b>Description</b>	<p>Enables or disables the defensive programming feature for the module EthSM.</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>	<b>VariantPostBuild:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>EthSMPrecondAssertEnabled</b>
<b>Label</b>	Enable Precondition Assertions
<b>Description</b>	<p>Enables handling of precondition assertion checks reported from the module EthSM.</p> <p>Dependency on parameter(s):</p>



	<ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EthSMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EthSMDefProgEnabled</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>EthSMPPostcondAssertEnabled</b>
<b>Label</b>	Enable Postcondition Assertions
<b>Description</b>	<p>Enables handling of postcondition assertion checks reported from the module EthSM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EthSMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EthSMDefProgEnabled</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>EthSMStaticAssertEnabled</b>
<b>Label</b>	Enable Static Assertions
<b>Description</b>	<p>Enables handling of static assertion checks reported from the module EthSM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EthSMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EthSMDefProgEnabled</code>): must be enabled</li> </ul>
<b>Multiplicity</b>	1..1



Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>EthSMUnreachAssertEnabled</b>	
Label	Enable Unreachable Code Assertions	
Description	<p>Enables handling of unreachable code assertion checks reported from the module EthSM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EthSMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EthSMDefProgEnabled</code>): must be enabled</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>EthSMInvariantAssertEnabled</b>	
Label	Enable Invariant Assertions	
Description	<p>Enables handling of invariant assertion checks reported from functions of the module EthSM.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>EthSMDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>EthSMDefProgEnabled</code>): must be enabled</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild



<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

#### 5.4.1.6. EthSMNetwork

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">EthSMDemEventParameter-Refs</a>	0..1	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.

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

<b>Parameter Name</b>	<b>EthSMEthIfControllerRef</b>	
<b>Description</b>	ID of the Ethernet controller assigned to the configured network handle. Reference to one of the controllers of EthIf configuration.	
<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>EthSMComMNetworkHandleRef</b>	
<b>Description</b>	Unique handle to identify one certain Ethernet network. Reference to one of the network handles configured for the ComM.	
<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>EthSMDevAuthNoComNotificationEnable</b>
<b>Description</b>	Enables DevAuth support for this EthSMNetwork.
<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

#### 5.4.1.7. EthSMDemEventParameterRefs

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

<b>Parameter Name</b>	<b>ETHSM_E_LINK_DOWN</b>
<b>Description</b>	<p>Reference to configured DEM event to report bus off errors for this Eth network.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ <code>EthSMDemCtrlTestResultReportToDem</code>: Select DEM to enable the reporting of <code>ETHSM_E_LINK_DOWN</code>.</li> </ul> <p>Further notes:</p> <ul style="list-style-type: none"> <li>▶ Activation: This error is reported when the transceiver switches to "down" while communication has already been established and is requested because of communication request.</li> <li>▶ Healing: This error is healed as soon as a transition from <code>ETHSM_STATE_ONHOLD</code> to <code>ETHSM_STATE_ONLINE</code> is successfully performed, which is triggered by <code>EthSM_TrcvLinkStateChg(ETHTRCV_LINK_STATE_ACTIVE)</code>.</li> <li>▶ Trigger debounce: None. The error is reported on first occurrence.</li> <li>▶ Rate of diagnostic checks: Checked on every <code>EthSM_Mainfunction()</code> call.</li> </ul>
<b>Multiplicity</b>	0..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild



Origin	AUTOSAR_ECUC
--------	--------------

#### 5.4.1.8. PublishedInformation

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

Parameter Name	<b>PbcfgMSupport</b>
Label	PbcfgM support
Description	Specifies whether or not the EthSM can use the PbcfgM module for post-build support.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	<b>PublishedInformation:</b>
Origin	Elektrobit Automotive GmbH

#### 5.4.2. Application programming interface (API)

##### 5.4.2.1. Macro constants

###### 5.4.2.1.1. ETHSM\_CTRLMODEINDICATION\_SVCID

Purpose	Defines API service ID of function EthSM_CtrlModelIndication().
Value	0x09U

###### 5.4.2.1.2. ETHSM\_DEVAUTHMODEINDICATION\_SVCID

Purpose	Defines API service ID of function EthSM_DevAuthNoComModelIndication().
Value	0x11U



#### 5.4.2.1.3. ETHSM\_E\_BUSSMMODEIND

Purpose	API request returns an error.
Value	0x09U

#### 5.4.2.1.4. ETHSM\_E\_INVALID\_NETWORK\_HANDLE

Purpose	API requests called with invalid parameter in parameter list.
Value	0x04U

#### 5.4.2.1.5. ETHSM\_E\_INVALID\_NETWORK\_MODE

Purpose	API requests called with invalid communication mode.
Value	0x01U

#### 5.4.2.1.6. ETHSM\_E\_INVALID\_TRCV\_LINK\_STATE

Purpose	API requests called with invalid parameter in parameter list.
Value	0x06U

#### 5.4.2.1.7. ETHSM\_E\_INVALID\_TcpIpMode

Purpose	API requests called with invalid parameter in parameter list.
Value	0x05U

#### 5.4.2.1.8. ETHSM\_E\_PARAM\_CONTROLLER

Purpose	API requests called with invalid parameter in parameter list.
Value	0x07U

#### 5.4.2.1.9. ETHSM\_E\_PARAM\_POINTER

Purpose	API requests called with invalid pointer in parameter list.
Value	0x03U



#### 5.4.2.1.10. ETHSM\_E\_PARAM\_TRANSCEIVER

Purpose	API requests called with invalid parameter in parameter list.
Value	0x08U

#### 5.4.2.1.11. ETHSM\_E\_UNINIT

Purpose	API requests called before EthSM module is initialized.
Value	0x02U

#### 5.4.2.1.12. ETHSM\_GETCURRENTCOMMODE\_SVCID

Purpose	Defines API service ID of function <a href="#">EthSM_GetCurrentComMode()</a> .
Value	0x04U

#### 5.4.2.1.13. ETHSM\_GETCURRENTINTERNALMODE\_SVCID

Purpose	Defines API service ID of function EthSM_GetCurrentInternalMode().
Value	0x03U

#### 5.4.2.1.14. ETHSM\_GETVERSIONINFO\_SVCID

Purpose	Defines API service ID of function <a href="#">EthSM_GetVersionInfo()</a> .
Value	0x02U

#### 5.4.2.1.15. ETHSM\_INIT\_SVCID

Purpose	Defines API service ID of function <a href="#">EthSM_Init()</a> .
Value	0x07U

#### 5.4.2.1.16. ETHSM\_INSTANCE\_ID

Purpose	Defines the instance number of this Ethernet State Manager. Since multiple instances of Ethernet State Manager are not supported the Instance Id is always zero.
---------	--



<b>Value</b>	0U
--------------	----

#### 5.4.2.1.17. ETHSM\_MAINFUNCTION\_SVCID

<b>Purpose</b>	Defines API service ID of function <a href="#">EthSM_MainFunction()</a> .
<b>Value</b>	0x01U

#### 5.4.2.1.18. ETHSM\_REQUESTCOMMODE\_SVCID

<b>Purpose</b>	Defines API service ID of function <a href="#">EthSM_RequestComMode()</a> .
<b>Value</b>	0x05U

#### 5.4.2.1.19. ETHSM\_TCPIPMODEINDICATION\_SVCID

<b>Purpose</b>	Defines API service ID of function <a href="#">EthSM_TcpIpModeIndication()</a> .
<b>Value</b>	0x08U

#### 5.4.2.1.20. ETHSM\_TRCVLINKSTATECHG\_SVCID

<b>Purpose</b>	Defines API service ID of function EthSM_TrcvLinkStateChg().
<b>Value</b>	0x06U

#### 5.4.2.1.21. ETHSM\_TRCVMODEINDICATION\_SVCID

<b>Purpose</b>	Defines API service ID of function EthSM_TrcvModeIndication().
<b>Value</b>	0x10U

### 5.4.2.2. Functions

#### 5.4.2.2.1. EthSM\_GetCurrentComMode

<b>Purpose</b>	Gets the current communication mode.
----------------	--------------------------------------



<b>Synopsis</b>	<code>Std_ReturnType EthSM_GetCurrentComMode ( NetworkHandleType NetworkHandle , ComM_ModeType * ComM_ModePtr );</code>	
<b>Service ID</b>	0x04	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	NetworkHandle	- Handle of the affected network
<b>Parameters (out)</b>	ComM_ModePtr	- Address to write the current Com mode to.
<b>Return Value</b>	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
<b>Description</b>	This service gets the current communication mode of the network.	

#### 5.4.2.2.2. EthSM\_GetVersionInfo

<b>Purpose</b>	Get version information of the Ethernet State Manager.	
<b>Synopsis</b>	<code>void EthSM_GetVersionInfo ( Std_VersionInfoType * versioninfo );</code>	
<b>Service ID</b>	0x02	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (out)</b>	versioninfo	- Pointer 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>	

#### 5.4.2.2.3. EthSM\_Init

<b>Purpose</b>	<a href="#">EthSM_Init()</a> initializes the EthSM module.
<b>Synopsis</b>	<code>Std_ReturnType EthSM_Init ( const EthSM_ConfigType * ConfigPtr );</code>



<b>Service ID</b>	0x07	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	ConfigPtr	- Address of the post-build configuration data structure.
<b>Return Value</b>	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
<b>Description</b>	This service initializes the EthSM module. It shall be the first function of the module which to be called.	

#### 5.4.2.2.4. EthSM\_IsValidConfig

<b>Purpose</b>	Checks compatibility of the post-build-time configuration.	
<b>Synopsis</b>	Std_ReturnType <b>EthSM_IsValidConfig</b> ( const void * voidConfigPtr ) ;	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	voidConfigPtr	Pointer to the configuration data of the EthSM module.
<b>Return Value</b>	Result of compatibility check	
	E_OK	Provided configuration is compatible.
	E_NOT_OK	Provided configuration is not compatible.
<b>Description</b>	This service checks the compatibility of the post-build-time configuration against the source code.	

#### 5.4.2.2.5. EthSM\_MainFunction

<b>Purpose</b>	EthSM main function.	
<b>Synopsis</b>	void <b>EthSM_MainFunction</b> ( void ) ;	
<b>Service ID</b>	0x01	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Description</b>	This service operate the effects of the EthSM state machine.	



#### 5.4.2.2.6. EthSM\_RequestComMode

<b>Purpose</b>	Service to set the requested Com mode.	
<b>Synopsis</b>	Std_ReturnType <b>EthSM_RequestComMode</b> ( NetworkHandleType NetworkHandle , ComM_ModeType ComM_Mode );	
<b>Service ID</b>	0x05	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	NetworkHandle	- Handle of the affected network.
	ComM_Mode	- Com mode requested.
<b>Return Value</b>	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
<b>Description</b>	This service sets the Com mode requested by the upper layer.	

#### 5.4.2.2.7. EthSM\_TcpIpModeIndication

<b>Purpose</b>	TcpIp mode indication.	
<b>Synopsis</b>	void <b>EthSM_TcpIpModeIndication</b> ( uint8 CtrlIdx , TcpIp_StateType TcpIpState );	
<b>Service ID</b>	0x08	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	- Controller index which changed the TcpIp state.
	TcpIpState	- New state the TcpIp has changed to.
<b>Description</b>	This service is called by the TcpIp to report the actual TcpIp state.	

### 5.4.3. Integration notes

#### 5.4.3.1. Exclusive areas

This section describes the exclusive areas used by the EthSM module.



#### 5.4.3.1.1. SCHM\_ETHSM\_EXCLUSIVE\_AREA\_0

<b>Protected data structures</b>	The exclusive area protects the variables containing actual ComM mode, the actual Tcplp state and the actual EthTrcv link state and the corresponding notification flags.  Furthermore the initialization of all global variables is protected with this exclusive area.
<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.  The locking mechanism can be disabled if it is ensured that <code>EthSM_MainFunction()</code> does not preempt one of the API functions <code>EthSM_RequestComMode()</code> , <code>EthSM_TrcvLinkStateChg()</code> or <code>EthSM_TcpIpModeIndication()</code> and vice versa.  The locking mechanism can be disabled if it is ensured that no EthSM API function preempts <code>EthSM_Init()</code> .

#### 5.4.3.2. Production errors

Production errors are not reported by the `EthSM` module.

#### 5.4.3.3. 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
CONFIG_DATA_UNSPECIFIED
VAR_INIT_8
VAR_CLEARED_8



VAR\_INIT\_16

VAR\_INIT\_UNSPECIFIED

VAR\_CLEARED\_16

VAR\_CLEARED\_UNSPECIFIED

CONST\_32

CONST\_UNSPECIFIED

CODE

CODE

#### 5.4.3.4. 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.

##### 5.4.3.4.1. lim.EthSm.EB\_INTREQ\_EthSm\_0001

<b>Description</b>	The reinitialization process shall not interrupt other module functions If reinitialization of the module is required, the call of EthSM_Init shall not interrupt other module functions.
<b>Rationale</b>	The reinitialisation process resets all internal variables. Continuing an interrupted module function after reinitialization can lead to undefined module behavior.

## 5.5. Sd

### 5.5.1. Configuration parameters

Containers included		
Container name	Multiplicity	Description
<a href="#">CommonPublishedInformation</a>	1..1	<b>Label:</b> Common Published Information

**Containers included**

		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.
<a href="#">SdConfig</a>	1..n	This container contains the configuration parameters and sub containers of the AUTOSAR Service Discovery module.
<a href="#">SdGeneral</a>	1..1	This container lists the general configuration parameters for the Service Discovery module.
<a href="#">SdDefensiveProgramming</a>	1..1	<b>Label:</b> Defensive Programming Options  Parameters for defensive programming

**Parameters included**

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

<b>Parameter Name</b>	<a href="#">IMPLEMENTATION_CONFIG_VARIANT</a>	
<b>Label</b>	Config Variant	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	VariantPostBuild	
<b>Range</b>	VariantPostBuild	
<b>Configuration class</b>	<a href="#">VariantPostBuild:</a>	VariantPostBuild

**5.5.1.1. CommonPublishedInformation****Parameters included**

Parameter name	Multiplicity
<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

**Parameters included**

<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>	4
<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>	2
<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>	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>	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>	4
<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>	11
<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>	171
<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. 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 Sd 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

### 5.5.1.3. SdConfig

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">SdInstance</a>	0..n	This container represents an instance of the SD; i.e. the SD configuration for a certain link.

### 5.5.1.4. SdInstance

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">SdClientService</a>	0..n	This container specifies all parameters used by Client services.
<a href="#">SdClientTimer</a>	0..n	This container specifies all timers used by the Service Discovery module for Client Services.
<a href="#">SdInstanceDemEventParameterRefs</a>	0..1	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.
<a href="#">SdInstanceMulticastRxPdu</a>	1..1	This container specifies the received PDU.
<a href="#">SdInstanceTxPdu</a>	1..1	This container specifies the transmitted PDU.
<a href="#">SdInstanceUnicastRxPdu</a>	1..1	This container specifies the received PDU.
<a href="#">SdServerService</a>	0..n	This container specifies all parameters used by Server services.
<a href="#">SdServerTimer</a>	0..n	This container specifies all timers used by the Service Discovery module for Server Services.



#### Parameters included

Parameter name	Multiplicity
<a href="#">SdInstanceHostname</a>	0..1
<a href="#">SdMaximumRemoteNodes</a>	1..1
<a href="#">SdSubscribeEventgroupRetryMax</a>	1..1
<a href="#">SdSubscribeEventgroupRetryDelay</a>	0..1

Parameter Name	<b>SdInstanceHostname</b>	
Description	Configuration parameter to specify the Hostname.	
Multiplicity	0..1	
Type	STRING	
Configuration class	<b>PostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>SdMaximumRemoteNodes</b>	
Description	Maximum number of remote nodes supported by this SdInstance.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65533 >=0	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>SdSubscribeEventgroupRetryMax</b>	
Description	Maximum count of retry a subscription, if a subscription to an event group is not acknowledged by SubscribeEventGroupAck or SubscribeEventGroupNack. 0x0=no retry, 0xFF=retry forever (as long as the event group is requested).	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>SdSubscribeEventgroupRetryDelay</b>	



<b>Description</b>	Time in seconds when a subscription to an event group shall be retriggered, if no SubscribeEventGroupAck or SubscribeEventGroupNack was received.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.01	
<b>Range</b>	<=50 >=0.001	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

### 5.5.1.5. SdClientService

Containers included		
Container name	Multiplicity	Description
<a href="#">SdClientCapabilityRecord</a>	0..n	Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service.
<a href="#">SdConsumedEventGroup</a>	0..n	A Service may have event groups which can be consumed. A service consumer has to subscribe to the corresponding event-group. After the subscription the event consumer takes the role of a server and the event provider that of a client.
<a href="#">SdBlacklistedVersions</a>	0..n	
<a href="#">SdConsumedMethods</a>	0..1	Container element for representing the data path for accessing the server methods.

Parameters included	
Parameter name	Multiplicity
<a href="#">SdClientServiceAutoRequire</a>	1..1
<a href="#">SdClientServiceHandleId</a>	1..1
<a href="#">SdClientServiceId</a>	1..1
<a href="#">SdClientServiceInstanceId</a>	1..1
<a href="#">SdClientServiceMajorVersion</a>	1..1
<a href="#">SdClientServiceMinorVersion</a>	1..1
<a href="#">SdVersionDrivenFindBehavior</a>	1..1



#### Parameters included

<a href="#">SdClientServiceTcpRef</a>	0..1
<a href="#">SdClientServiceTimerRef</a>	1..1
<a href="#">SdClientServiceUdpRef</a>	0..1

<b>Parameter Name</b>	<b>SdClientServiceAutoRequire</b>	
<b>Description</b>	If existing and set to true, this Service will be set to "required" on start.	
<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>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdClientServiceHandleId</b>	
<b>Description</b>	The HandleId by which the BswM can identify this Client Service Instance.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Range</b>	<=65535 >=0	
<b>Configuration class</b>	<a href="#">VariantPostBuild:</a>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdClientServiceId</b>	
<b>Description</b>	Id to identify the service. This is unique for the service interface.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Range</b>	<=65534 >=0	
<b>Configuration class</b>	<a href="#">VariantPostBuild:</a>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdClientServiceInstancId</b>	
-----------------------	---------------------------------	--



<b>Description</b>	Configuration parameter to specify Instance Id of the service as used in SD entries.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Range</b>	<=65534 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdClientServiceMajorVersion</b>	
<b>Description</b>	Major version number of the Service as used in the SD entries.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	1	
<b>Range</b>	<=254 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdClientServiceMinorVersion</b>	
<b>Description</b>	Minor version number of the Service as used in the SD Service Entries.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	4294967295	
<b>Range</b>	<=4294967295 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdVersionDrivenFindBehavior</b>	
<b>Description</b>	Defines the possible acceptance kinds for required service instances.  ► EXACT_OR_ANY_MINOR_VERSION: Search for ANY or specific minor version service instance and select either ALL returned service instances (in	



	<p>case of ANY) or exactly the specific minor version service instances defined in SdClientServiceMinorVersion.</p> <ul style="list-style-type: none"> <li>▶ MINIMUM_MINOR_VERSION: Search for ANY minor version service instance and select only those service instances which have an equal or greater minor version than given in SdClientServiceMinorVersion.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	EXACT_OR_ANY_MINOR_VERSION
<b>Range</b>	EXACT_OR_ANY_MINOR_VERSION MINIMUM_MINOR_VERSION
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SdClientServiceTcpRef</b>
<b>Description</b>	Reference to the SoAdSocketConnection representing the data path (TCP) for communication with methods.
<b>Multiplicity</b>	0..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>SdClientServiceTimerRef</b>
<b>Description</b>	The reference of the SdClientTimer container for this service.
<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>SdClientServiceUdpRef</b>
<b>Description</b>	Reference to the SoAdSocketConnection representing the data path (UDP) for communication with methods.
<b>Multiplicity</b>	0..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild



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

### 5.5.1.6. SdClientCapabilityRecord

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">SdClientServiceCapabilityRecordKey</a>	1..1
<a href="#">SdClientServiceCapabilityRecordValue</a>	0..1

<b>Parameter Name</b>	<b>SdClientServiceCapabilityRecordKey</b>
<b>Description</b>	Defines a CapabilityRecord key.
<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>SdClientServiceCapabilityRecordValue</b>
<b>Description</b>	Defines the corresponding CapabilityRecord value.
<b>Multiplicity</b>	0..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.5.1.7. SdConsumedEventGroup

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">SdClientCapabilityRecord</a>	0..n	Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service.

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



#### Parameters included

<a href="#">SdConsumedEventGroupHandleId</a>	1..1
<a href="#">SdConsumedEventGroupId</a>	1..1
<a href="#">SdConsumedEventGroupMulticastGroupRef</a>	0..n
<a href="#">SdConsumedEventGroupTimerRef</a>	1..1
<a href="#">SdConsumedEventGroupMulticastActivationRef</a>	0..1
<a href="#">SdConsumedEventGroupTcpActivationRef</a>	0..1
<a href="#">SdConsumedEventGroupUdpActivationRef</a>	0..1

<b>Parameter Name</b>	<b>SdConsumedEventGroupAutoRequire</b>	
<b>Description</b>	If existing and set to true, this EventGroup will be set to "required" on start.	
<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>SdConsumedEventGroupHandleId</b>	
<b>Description</b>	The HandleId by which the BswM can identify this EventGroup.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Range</b>	<=65535 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdConsumedEventGroupId</b>	
<b>Description</b>	The Eventgroup Id of this eventGroup as a unique identifier of the eventgroup in this service. This identifier is used for EventGroup entries as well.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Range</b>	<=65534	



	>=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>SdConsumedEventGroupMulticastGroupRef</b>
<b>Description</b>	Reference to the SoAdSocketConnectionGroup representing the multicast data path (UDP).
<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>SdConsumedEventGroupTimerRef</b>
<b>Description</b>	The reference of the SdClientTimer container for this eventGroup.
<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>SdConsumedEventGroupMulticastActivationRef</b>
<b>Description</b>	The reference of a Routing Group in order to activate and setup the Socket Connection for Multicast Events of this EventGroup. The multicast address from the received Multicast option is setup by SoAd_RequestIpAddrAssignment.
<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>SdConsumedEventGroupTcpActivationRef</b>
<b>Description</b>	The reference of the Routing Group for activation of the data path for receiving TCP events.
<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>SdConsumedEventGroupUdpActivationRef</b>	
<b>Description</b>	The reference of the Routing Group for activation of the data path for receiving UDP events.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.5.1.8. SdClientCapabilityRecord

<b>Parameters included</b>		
<b>Parameter name</b>		<b>Multiplicity</b>
<a href="#">SdClientServiceCapabilityRecordKey</a>		1..1
<a href="#">SdClientServiceCapabilityRecordValue</a>		0..1

<b>Parameter Name</b>	<b>SdClientServiceCapabilityRecordKey</b>	
<b>Description</b>	Defines a CapabilityRecord key.	
<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>SdClientServiceCapabilityRecordValue</b>	
<b>Description</b>	Defines the corresponding CapabilityRecord value.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	STRING	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.5.1.9. SdBlacklistedVersions

<b>Parameters included</b>		
<b>Parameter name</b>		<b>Multiplicity</b>



#### Parameters included

<a href="#">SdBlacklistedMinorVersion</a>	1..1
<b>Parameter Name</b>	<b>SdBlacklistedMinorVersion</b>
<b>Description</b>	Blacklisted MinorVersions.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Range</b>	<=4294967294 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.5.1.10. SdConsumedMethods

##### Parameters included

Parameter name	Multiplicity
<a href="#">SdClientServiceActivationRef</a>	1..1
<b>Parameter Name</b>	<b>SdClientServiceActivationRef</b>
<b>Description</b>	Reference to a SoAdRoutingGroupRef to activate/deactivate the data path for the methods.
<b>Multiplicity</b>	1..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.5.1.11. SdClientTimer

##### Parameters included

Parameter name	Multiplicity
<a href="#">SdClientTimerInitialFindDelayMax</a>	0..1
<a href="#">SdClientTimerInitialFindDelayMin</a>	0..1
<a href="#">SdClientTimerInitialFindRepetitionsBaseDelay</a>	0..1



#### Parameters included

<a href="#">SdClientTimerInitialFindRepetitionsMax</a>	0..1
<a href="#">SdClientTimerRequestResponseMaxDelay</a>	0..1
<a href="#">SdClientTimerRequestResponseMinDelay</a>	0..1
<a href="#">SdClientTimerTTL</a>	1..1

<b>Parameter Name</b>	<b>SdClientTimerInitialFindDelayMax</b>	
<b>Description</b>	Max value in [s] to delay randomly the transmission of a find message. This parameter is mandatory for ClientService.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdClientTimerInitialFindDelayMin</b>	
<b>Description</b>	Min value in [s] to delay randomly the transmission of a find message. This parameter is mandatory for ClientService.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.0	
<b>Range</b>	<=4294967294 >=0.0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdClientTimerInitialFindRepetitionsBaseDelay</b>	
<b>Description</b>	The base delay in [s] for find repetitions. Successive finds have an exponential back off delay (1x base delay, 2x base delay, 4x base delay, ...). This parameter is mandatory for ClientService.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.01	
<b>Range</b>	<=Infinity	



	>=0.0
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>SdClientTimerInitialFindRepetitionsMax</b>	
<b>Description</b>	Configuration for the maximum number of find repetitions. This parameter is mandatory for ClientService.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	3	
<b>Range</b>	<=10 >=0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdClientTimerRequestResponseMaxDelay</b>	
<b>Description</b>	Maximum allowable response delay to entries received by multicast in seconds. This parameter is mandatory for ConsumedEventGroups.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdClientTimerRequestResponseMinDelay</b>	
<b>Description</b>	Minimum allowable response delay to the find message in seconds. This parameter is mandatory for ConsumedEventGroups.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.0	
<b>Range</b>	<=4294967294 >=0.0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild



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

<b>Parameter Name</b>	SdClientTimerTTL	
<b>Description</b>	Time to live for find and subscribe messages.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	1	
<b>Range</b>	<=16777215 >=1	
<b>Configuration class</b>	VariantPostBuild:	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.5.1.12. SdInstanceDemEventParameterRefs

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">SD_E_MALFORMED_MSG</a>	0..1
<a href="#">SD_E_OUT_OF_RES</a>	0..1
<a href="#">SD_E_SUBSCR_NACK_RECV</a>	0..1
<a href="#">SD_E_SERVER_NOT_AVAILABLE</a>	0..1

<b>Parameter Name</b>	SD_E_MALFORMED_MSG	
<b>Description</b>	Reference to the DemEventParameter which shall be issued when the SD Instance received malformed message.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	PreCompile:	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	SD_E_OUT_OF_RES	
<b>Description</b>	Reference to the DemEventParameter which shall be issued when the SD Instance does not have enough resources to handle client.	
<b>Multiplicity</b>	0..1	



Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>SD_E_SUBSCR_NACK_RECV</b>	
Description	Reference to the DemEventParameter which shall be issued when receiving SubscribeEventgroupNack entry.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>SD_E_SERVER_NOT_AVAILABLE</b>	
Description	Reference to the DemEventParameter which shall be issued when a server service which had previously sent an OFFER message does not send any more OFFERS after the previous OFFER TTL dies out.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

### 5.5.1.13. SdInstanceMulticastRxPdu

Parameters included		
Parameter name	Multiplicity	
SdRxPdulId	1..1	
SdRxPduRef	1..1	

Parameter Name	<b>SdRxPdulId</b>	
Description	ID of the PDU that will be received via the API Sd_SoAdIfRxIndication().	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	



<b>Parameter Name</b>	<b>SdRxPduRef</b>	
<b>Description</b>	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

#### 5.5.1.14. SdInstanceTxPdu

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

<b>Parameter Name</b>	<b>SdTxDRef</b>
<b>Description</b>	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	AUTOSAR_ECUC

#### 5.5.1.15. SdInstanceUnicastRxPdu

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

<b>Parameter Name</b>	<b>SdRxPduld</b>
<b>Description</b>	ID of the PDU that will be received via the API Sd_SoAdlfRxIndication().
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	SdRxPduRef	
<b>Description</b>	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

### 5.5.1.16. SdServerService

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">SdEventHandler</a>	0..n	Container Element for representing an EventGroup as part of the Service Instance.
<a href="#">SdProvidedMethods</a>	0..1	Container element for representing the needed elements of the data path for the methods provided by the service.
<a href="#">SdServerCapabilityRecord</a>	0..n	Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service.

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">SdServerServiceAutoAvailable</a>	1..1
<a href="#">SdServerServiceHandleId</a>	1..1
<a href="#">SdServerServiceId</a>	1..1
<a href="#">SdServerServiceInstanceId</a>	1..1
<a href="#">SdServerServiceMajorVersion</a>	1..1
<a href="#">SdServerServiceMinorVersion</a>	1..1
<a href="#">SdServerServiceTcpRef</a>	0..1
<a href="#">SdServerServiceTimerRef</a>	1..1
<a href="#">SdServerServiceUdpRef</a>	0..1



<b>Parameter Name</b>	<b>SdServerServiceAutoAvailable</b>	
<b>Description</b>	If existing and set to true, this Service will be set to "Available" on start.	
<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>SdServerServiceHandleId</b>	
<b>Description</b>	The HandleId by which the BswM can identify this Server Service Instance.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Range</b>	<=65535 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdServerServiceId</b>	
<b>Description</b>	Id to identify the service. This is unique for the service interface.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Range</b>	<=65534 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdServerServiceInstanceld</b>	
<b>Description</b>	Configuration parameter to specify Instance Id of the Service implemented by the Server Service.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Range</b>	<=65534	



	>=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdServerServiceMajorVersion</b>	
<b>Description</b>	Major version number of the Service as used in SD Entries.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	1	
<b>Range</b>	<=254 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdServerServiceMinorVersion</b>	
<b>Description</b>	Minor version number of the Service as used e.g. in Offer Service entries.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Range</b>	<=4294967294 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdServerServiceTcpRef</b>	
<b>Description</b>	Reference to SoAdSocketConnectionGroup used for methods.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdServerServiceTimerRef</b>	
<b>Description</b>	The reference of the SdServerTimer container for this service.	



<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>SdServerServiceUdpRef</b>
<b>Description</b>	Reference to SoAdSocketConnectionGroup used for methods.
<b>Multiplicity</b>	0..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.5.1.17. SdEventHandler

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">SdEventHandlerMulticast</a>	0..1	The subcontainer including the Routing Group for Activation of Events sent over Multicast.
<a href="#">SdEventHandlerTcp</a>	0..1	The subcontainer including the Routing Groups for Activation and Trigger Transmit for Events sent over TCP.
<a href="#">SdEventHandlerUdp</a>	0..1	The subcontainer including the Routing Groups for Activation and Trigger Transmit for Events sent over UDP.
<a href="#">SdServerCapabilityRecord</a>	0..n	Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service.

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">SdEventHandlerEventGroupId</a>	1..1
<a href="#">SdEventHandlerHandleId</a>	1..1
<a href="#">SdEventHandlerMulticastThreshold</a>	1..1
<a href="#">SdEventHandlerTimerRef</a>	1..1

<b>Parameter Name</b>	<b>SdEventHandlerEventGroupId</b>
-----------------------	-----------------------------------



<b>Description</b>	The EventGroup Id of this EventGroup as a unique identifier of the EventGroup in this service. This identifier is used for EventGroup entries as well.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Range</b>	<=65534 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdEventHandlerHandleId</b>	
<b>Description</b>	The HandleId by which the BswM can identify this EventGroup.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Range</b>	<=65535 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdEventHandlerMulticastThreshold</b>	
<b>Description</b>	Specifies the number of subscribed clients that trigger the Server to change the transmission of events to Multicast.	
<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	

<b>Parameter Name</b>	<b>SdEventHandlerTimerRef</b>	
<b>Description</b>	The reference of the SdServerTimer container for this EventGroup.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild



Origin	AUTOSAR_ECUC
--------	--------------

### 5.5.1.18. SdEventHandlerMulticast

Parameters included	
Parameter name	Multiplicity
<a href="#">SdEventActivationRef</a>	0..1
<a href="#">SdMulticastEventSoConRef</a>	0..1

Parameter Name	<b>SdEventActivationRef</b>	
Description	Reference to a SoAdRoutingGroup for activation of the data path for a subscribed client (start sending events after subscribe).	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	<b>PostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>SdMulticastEventSoConRef</b>	
Description	Reference to the SoAdSocketConnection representing the multicast data path (UDP).	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	<b>PostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.5.1.19. SdEventHandlerTcp

Parameters included	
Parameter name	Multiplicity
<a href="#">SdEventActivationRef</a>	0..1
<a href="#">SdEventTriggeringRef</a>	0..1
Parameter Name	
<b>SdEventActivationRef</b>	



<b>Description</b>	Reference to a SoAdRoutingGroup for activation of the data path for a subscribed client (start sending events after subscribe).	
<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>SdEventTriggeringRef</b>	
<b>Description</b>	Reference to a SoAdRoutingGroup that is used for triggered transmit. Triggering is needed to sent out initial events on the server side after a client got subscribed.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.5.1.20. SdEventHandlerUdp

<b>Parameters included</b>		
<b>Parameter name</b>		<b>Multiplicity</b>
<a href="#"><u>SdEventActivationRef</u></a>		0..1
<a href="#"><u>SdEventTriggeringRef</u></a>		0..1

<b>Parameter Name</b>	<b>SdEventActivationRef</b>	
<b>Description</b>	Reference to a SoAdRoutingGroup for activation of the data path for a subscribed client (start sending events after subscribe).	
<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>SdEventTriggeringRef</b>	
<b>Description</b>	Reference to a SoAdRoutingGroup that is used for triggered transmit. Triggering is needed to sent out initial events on the server side after a client got subscribed.	



<b>Multiplicity</b>	0..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.5.1.21. SdServerCapabilityRecord

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">SdServerCapabilityRecordKey</a>	1..1
<a href="#">SdServerCapabilityRecordValue</a>	0..1

<b>Parameter Name</b>	<b>SdServerCapabilityRecordKey</b>
<b>Description</b>	Defines a CapabilityRecord key.
<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>SdServerCapabilityRecordValue</b>
<b>Description</b>	Defines the corresponding CapabilityRecord value.
<b>Multiplicity</b>	0..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.5.1.22. SdProvidedMethods

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



<b>Description</b>	Reference to a SoAdRoutingGroup to activated and deactivate the data path for methods of the service.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	SYMBOLIC-NAME-REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.5.1.23. SdServerCapabilityRecord

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">SdServerCapabilityRecordKey</a>	1..1
<a href="#">SdServerCapabilityRecordValue</a>	0..1

<b>Parameter Name</b>	<b>SdServerCapabilityRecordKey</b>
<b>Description</b>	Defines a CapabilityRecord key.
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>SdServerCapabilityRecordValue</b>
<b>Description</b>	Defines the corresponding CapabilityRecord value.
<b>Multiplicity</b>	0..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>PostBuild:</b>
<b>Origin</b>	AUTOSAR_ECUC

### 5.5.1.24. SdServerTimer

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



#### Parameters included

<a href="#">SdServerTimerInitialOfferDelayMin</a>	0..1
<a href="#">SdServerTimerInitialOfferRepetitionBaseDelay</a>	0..1
<a href="#">SdServerTimerInitialOfferRepetitionsMax</a>	0..1
<a href="#">SdServerTimerOfferCyclicDelay</a>	0..1
<a href="#">SdServerTimerRequestResponseMaxDelay</a>	1..1
<a href="#">SdServerTimerRequestResponseMinDelay</a>	1..1
<a href="#">SdServerTimerTTL</a>	1..1

Parameter Name	<b>SdServerTimerInitialOfferDelayMax</b>	
<b>Description</b>	Max value in [s] to delay randomly the first offer. This parameter is mandatory for ServerService.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

Parameter Name	<b>SdServerTimerInitialOfferDelayMin</b>	
<b>Description</b>	Min value in [s] to delay randomly the first offer. This parameter is mandatory for ServerService.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Range</b>	<=4294967294 >=0.0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

Parameter Name	<b>SdServerTimerInitialOfferRepetitionBaseDelay</b>	
<b>Description</b>	The base delay in [s] for offer repetitions. Successive offers have an exponential back off delay (1x base delay, 2x base delay, 4x base delay, ...). This parameter is mandatory for ServerService.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	



<b>Range</b>	<=Infinity >=0.0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdServerTimerInitialOfferRepetitionsMax</b>	
<b>Description</b>	Configure the maximum amount of offer repetition. This parameter is mandatory for ServerService.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	3	
<b>Range</b>	<=10 >=0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdServerTimerOfferCyclicDelay</b>	
<b>Description</b>	Interval between cyclic offers in the main phase. This parameter is mandatory for ServerService.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Range</b>	<=Infinity >=0.0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdServerTimerRequestResponseMaxDelay</b>	
<b>Description</b>	Maximum allowable response delay to entries received by multicast in seconds.	
<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>SdServerTimerRequestResponseMinDelay</b>	
<b>Description</b>	Minimum allowable response delay to entries received by multicast in seconds.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.0	
<b>Range</b>	<=4294967294 >=0.0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SdServerTimerTTL</b>	
<b>Description</b>	Time to live for offer service.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Range</b>	<=16777215 >=1	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.5.1.25. SdGeneral

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">VendorSpecific</a>	1..1	Contains the vendor specific configuration parameters of the AUTOSAR Sd module.

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">SdHeaderFileInclusion</a>	0..n
<a href="#">SdDevErrorDetect</a>	1..1
<a href="#">SdMainFunctionCycleTime</a>	1..1
<a href="#">SdSubscribeEventgroupRetryEnable</a>	1..1
<a href="#">SdVersionInfoApi</a>	1..1



#### Parameters included

<a href="#">SdRelocatablePbcfgEnable</a>	1..1
<a href="#">SdCallOutEventMapping</a>	1..1
<a href="#">SdMaxInstanceCount</a>	1..1
<a href="#">SdMaxClientServiceCount</a>	1..1
<a href="#">SdMaxServerServiceCount</a>	1..1
<a href="#">SdMaxClientServiceEventGroupCount</a>	1..1
<a href="#">SdMaxServerServiceEventCount</a>	1..1
<a href="#">SdMaxConsumedEventGroupMulticastReferences</a>	1..1

Parameter Name	SdHeaderFileInclusion
<b>Multiplicity</b>	0..n
<b>Type</b>	STRING
<b>Origin</b>	AUTOSAR_ECUC

Parameter Name	SdDevErrorDetect
<b>Description</b>	Enables and disables the development error detection and notification mechanism.
<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

Parameter Name	SdMainFunctionCycleTime
<b>Description</b>	This parameter defines the cycle time in seconds of the periodic calling of Sd main function.
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.005
<b>Range</b>	<=1.0 >=1.0E-4
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC



<b>Parameter Name</b>	<b>SdSubscribeEventgroupRetryEnable</b>	
<b>Description</b>	Switch to enable or disable retry functionality to subscribe to Eventgroups of ServerServices with TTL set to 0xFFFFFFF.	
<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>SdVersionInfoApi</b>	
<b>Description</b>	Enables and disables the version info API.	
<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>SdRelocatablePbcfgEnable</b>	
<b>Description</b>	Enables/disable support for relocatable postbuild configuration.  ▶ True: Postbuild configuration relocatable in memory. ▶ False: Postbuild configuration not relocatable in memory.	
<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>SdCallOutEventMapping</b>	
<b>Description</b>	Enables and disables a call out to indicate any SWC when a client service is available and from which IP address this client service will be receiving its consumed event groups.	
<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>SdMaxInstanceCount</b>
<b>Description</b>	Maximum number of instances that can be defined in the configuration.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	8
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SdMaxClientServiceCount</b>
<b>Description</b>	Maximum number of client services that can be defined in the configuration.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	32
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SdMaxServerServiceCount</b>
<b>Description</b>	Maximum number of client services that can be defined in the configuration.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	32
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SdMaxClientServiceEventGroupCount</b>
<b>Description</b>	Maximum number of client services that can be defined in the configuration.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	32
<b>Configuration class</b>	<b>VariantPostBuild:</b>



<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

<b>Parameter Name</b>	<b>SdMaxServerServiceEventCount</b>	
<b>Description</b>	Maximum number of server services that can be defined in the configuration.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	32	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>SdMaxConsumedEventGroupMulticastReferences</b>	
<b>Description</b>	Maximum number of multicast references to the existing consumed event groups. This number can never exceed the total number of consumed event groups defined in the configuration	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.5.1.26. VendorSpecific

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#"><u>ProductionErrorHandler</u></a>	1..1	
<a href="#"><u>Sd_SendDataBackServerNotAvailable</u></a>	0..1	<p>This container provides the functionality to call a user defined function.</p> <p>If this container is not configured this functionality is disabled.</p>
<a href="#"><u>Sd_SendDataBackMalformedMsg</u></a>	0..1	<p>This container provides the functionality to call a user defined function.</p> <p>If this container is not configured this functionality is disabled.</p>
<a href="#"><u>Sd_SendDataBackSubscribeNackReceived</u></a>	0..1	<p>This container provides the functionality to call a user defined function.</p> <p>If this container is not configured this functionality is disabled.</p>



#### Parameters included

Parameter name	Multiplicity
<a href="#">SdTxFrameBufferSize</a>	1..1
<a href="#">SdTxOptionsBufferSize</a>	1..1
<a href="#">SdRxOptionsBufferSize</a>	1..1

Parameter Name	SdTxFrameBufferSize
<b>Description</b>	Defines the internal buffer size used to assemble frames for transmissions of Sd control messages. Indirectly defines the maximal size of Sd control frames. This buffer size also holds 8 byte of the SOME/IP header, resulting in a SOME/IP payload 8 bytes less.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	1408
<b>Range</b>	<=1408 >=40
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

Parameter Name	SdTxOptionsBufferSize
<b>Description</b>	Defines the maximum number of options transmitted within a Sd control messages.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	40
<b>Range</b>	<=250 >=3
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

Parameter Name	SdRxOptionsBufferSize
<b>Description</b>	Defines the maximum number of options being processes in a received Sd control messages.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER



<b>Default value</b>	120
<b>Range</b>	<=250 >=3
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.5.1.27. ProductionErrorHandler

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">Enable_SD_E_MALFORMED_MSG</a>	1..1
<a href="#">SD_E_MALFORMED_MSG_ReportToDemDetErrorId</a>	1..1
<a href="#">Enable_SD_E_OUT_OF_RES</a>	1..1
<a href="#">SD_E_OUT_OF_RES_ReportToDemDetErrorId</a>	1..1
<a href="#">Enable_SD_E_SUBSCR_NACK_RECV</a>	1..1
<a href="#">SD_E_SUBSCR_NACK_RECV_ReportToDemDetErrorId</a>	1..1
<a href="#">Enable_SD_E_SERVER_NOT_AVAILABLE</a>	1..1
<a href="#">SD_E_SERVER_NOT_AVAILABLE_ReportToDemDetErrorId</a>	1..1

<b>Parameter Name</b>	<b>Enable_SD_E_MALFORMED_MSG</b>
<b>Label</b>	Enable SD_E_MALFORMED_MSG
<b>Description</b>	<p>Enables DEM reporting in SD.</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> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</li> <li>▶ <b>Execution time reduction (code):</b> Disabling this parameter reduces the execution time of the module code.</li> </ul>
<b>Multiplicity</b>	1..1



Type	ENUMERATION	
Default value	DISABLE	
Range	DEM DET DISABLE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>SD_E_MALFORMED_MSG_ReportToDemDetErrorId</b>	
Label	SD_E_MALFORMED_MSG Det Error ID	
Description	If a production error is reported towards the Det, this parameter defines the error id which is reported towards the Det.	
Multiplicity	1..1	
Type	INTEGER	
Default value	129	
Range	<=255	
Configuration class	PreCompile:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>Enable_SD_E_OUT_OF_RES</b>	
Label	Enable SD_E_OUT_OF_RES	
Description	<p>Enables DEM reporting in SD.</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> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</li> <li>▶ <b>Execution time reduction (code):</b> Disabling this parameter reduces the execution time of the module code.</li> </ul>	
Multiplicity	1..1	
Type	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>SD_E_OUT_OF_RES_ReportToDemDetErrorId</b>
<b>Label</b>	SD_E_OUT_OF_RES Det Error ID
<b>Description</b>	If a production error is reported towards the Det, this parameter defines the error id which is reported towards the Det.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	130
<b>Range</b>	<=255
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>Enable_SD_E_SUBSCR_NACK_RECV</b>
<b>Label</b>	Enable SD_E_SUBSCR_NACK_RECV
<b>Description</b>	<p>Enables DEM reporting in SD.</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> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</li> <li>▶ <b>Execution time reduction (code):</b> Disabling this parameter reduces the execution time of the module code.</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>SD_E_SUBSCR_NACK_RECV_ReportToDemDetErrorId</b>
<b>Label</b>	SD_E_SUBSCR_NACK_RECV Det Error ID
<b>Description</b>	If a production error is reported towards the Det, this parameter defines the error id which is reported towards the Det.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	131
<b>Range</b>	<=255
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>Enable_SD_E_SERVER_NOT_AVAILABLE</b>
<b>Label</b>	Enable SD_E_SERVER_NOT_AVAILABLE
<b>Description</b>	<p>Enables DEM reporting in SD.</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> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</li> <li>▶ <b>Execution time reduction (code):</b> Disabling this parameter reduces the execution time of the module code.</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>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SD_E_SERVER_NOT_AVAILABLE_ReportToDemDetErrorId</b>	
<b>Label</b>	SD_E_SERVER_NOT_AVAILABLE Det Error ID	
<b>Description</b>	If a production error is reported towards the Det, this parameter defines the error id which is reported towards the Det.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	133	
<b>Range</b>	<=255	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.5.1.28. Sd\_SendDataBackServerNotAvailable

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

<b>Parameter Name</b>	<b>Sd_ServerNotAvailableCallBack</b>
<b>Description</b>	<p>User defined callback function to be called each time that a Server Service that had previously sent an OFFER stops sending OFFERs. This function will only be called if the client service is requested.</p> <p><b>Syntax:</b> (void) UserDefinedFunction(uint16 ServiceID, uint16 InstanceID, TcpIp_SockAddrType* UdpIpAddrPtr, TcpIp_SockAddrType* TcpIpAddrPtr)</p> <p>with UserDefinedFunction as placeholder for a user defined function name provided with this configuration field.</p> <p><b>Note:</b> User defined header files can be added to configuration container Sd-HeaderFileInclusion.</p>



<b>Multiplicity</b>	1..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.5.1.29. Sd\_SendDataBackMalformedMsg

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

<b>Parameter Name</b>	<b>Sd_MalformedMsgCallBack</b>
<b>Description</b>	User defined callback function to be called each time that a malformed Sd UDP frame is received.  Syntax: (void) UserDefinedFunction(void)  with UserDefinedFunction as placeholder for a user defined function name provided with this configuration field.  <b>Note:</b> User defined header files can be added to configuration container Sd-HeaderFileInclusion.
<b>Multiplicity</b>	1..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.5.1.30. Sd\_SendDataBackSubscribeNackReceived

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

<b>Parameter Name</b>	<b>Sd_SubscribeNackReceivedCallBack</b>
<b>Description</b>	User defined callback function to be called each time that a subscribe nack entry is received.



	<p>Syntax: (void) UserDefinedFunction(void)</p> <p>with UserDefinedFunction as placeholder for a user defined function name provided with this configuration field.</p> <p><b>Note:</b> User defined header files can be added to configuration container Sd-HeaderFileInclusion.</p>
<b>Multiplicity</b>	1..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>VariantPreCompile:</b> VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.5.1.31. SdDefensiveProgramming

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">SdDefProgEnabled</a>	1..1
<a href="#">SdPrecondAssertEnabled</a>	1..1
<a href="#">SdPostcondAssertEnabled</a>	1..1
<a href="#">SdStaticAssertEnabled</a>	1..1
<a href="#">SdUnreachAssertEnabled</a>	1..1
<a href="#">SdInvariantAssertEnabled</a>	1..1

<b>Parameter Name</b>	<b>SdDefProgEnabled</b>
<b>Label</b>	Enable Defensive Programming
<b>Description</b>	<p>Enables or disables the defensive programming feature for the module Sd.</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>	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>SdPrecondAssertEnabled</b>	
<b>Label</b>	Enable Precondition Assertions	
<b>Description</b>	<p>Enables handling of precondition assertion checks reported from the module Sd.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>SdDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>SdDefProgEnabled</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>SdPostcondAssertEnabled</b>	
<b>Label</b>	Enable Postcondition Assertions	
<b>Description</b>	<p>Enables handling of postcondition assertion checks reported from the module Sd.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>SdDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>SdDefProgEnabled</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>SdStaticAssertEnabled</b>	
<b>Label</b>	Enable Static Assertions	
<b>Description</b>	Enables handling of static assertion checks reported from the module Sd.	



	<p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>SdDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>SdDefProgEnabled</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>SdUnreachAssertEnabled</b>
<b>Label</b>	Enable Unreachable Code Assertions
<b>Description</b>	<p>Enables handling of unreachable code assertion checks reported from the module Sd.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>SdDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>SdDefProgEnabled</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>SdInvariantAssertEnabled</b>
<b>Label</b>	Enable Invariant Assertions
<b>Description</b>	<p>Enables handling of invariant assertion checks reported from functions of the module Sd.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>SdDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>SdDefProgEnabled</code>): must be enabled</li> </ul>
<b>Multiplicity</b>	1..1



Type	BOOLEAN
Default value	false
Configuration class	<b>VariantPreCompile:</b> VariantPreCompile
Origin	Elektrobit Automotive GmbH

## 5.5.2. Application programming interface (API)

### 5.5.2.1. Macro constants

#### 5.5.2.1.1. SD\_CHECKBUFFERFILLLEVEL\_SVCID

Purpose	Defines API id of function Sd_Send_checkBufferFillLevel.
Value	0x44U

#### 5.5.2.1.2. SD\_CLIENTSERVICECEVGEXITAVAILABLE\_SVCID

Purpose	Defines API id of function Sd_ClientService_CEvg_exit_AVAILABLE.
Value	0x1BU

#### 5.5.2.1.3. SD\_CLIENTSERVICEGOTOAVAILABLE\_SVCID

Purpose	Defines API id of function Sd_ClientService_mainFunction.
Value	0x19U

#### 5.5.2.1.4. SD\_CLIENTSERVICEMAINFUNCTION\_SVCID

Purpose	Defines API id of function Sd_ClientService_mainFunction.
Value	0x18U



#### 5.5.2.1.5. SD\_CLIENTSERVICEMAITMERS\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_mainTimersFunction.
<b>Value</b>	0x16U

#### 5.5.2.1.6. SD\_CLIENTSERVICEMAITTTL\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_mainTTL.
<b>Value</b>	0x17U

#### 5.5.2.1.7. SD\_CLIENTSERVICESETSTATE\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientServiceSetState.
<b>Value</b>	0x08U

#### 5.5.2.1.8. SD\_CLIENTSERVICESINITAFTERTHALT\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientServices_init_after_halt.
<b>Value</b>	0x1AU

#### 5.5.2.1.9. SD\_CLIENTSERVICE\_COMSMEVENT\_OFFER\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_ComSM_event_Offer.
<b>Value</b>	0x13U

#### 5.5.2.1.10. SD\_CLIENTSERVICE\_COMSMEVENT\_START\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_ComSM_event_start.
<b>Value</b>	0x10U

#### 5.5.2.1.11. SD\_CLIENTSERVICE\_COMSMEVENT\_STOPOFFER\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_ComSM_event_Stopoffer.
----------------	--



<b>Value</b>	0x14U
--------------	-------

#### 5.5.2.1.12. SD\_CLIENTSERVICE\_COMSMEVENT\_STOP\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_ComSM_event_stop.
<b>Value</b>	0x11U

#### 5.5.2.1.13. SD\_CLIENTSERVICE\_COMSMEVENT\_TIMEOUT\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_ComSM_event_timeout.
<b>Value</b>	0x12U

#### 5.5.2.1.14. SD\_CLIENTSERVICE\_COMSM\_EVENT\_STOP\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_ComSM_event_stop.
<b>Value</b>	0x1FU

#### 5.5.2.1.15. SD\_CLIENTSERVICE\_COMSM\_EVENT\_TIMEOUT

<b>Purpose</b>	Defines API id of function Sd_ClientService_ComSM_event_timeout.
<b>Value</b>	0x0BU

#### 5.5.2.1.16. SD\_CLIENTSERVICE\_COMSM\_SOCONMODECHECK\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_ComSM_SoConModeCheck.
<b>Value</b>	0x15U

#### 5.5.2.1.17. SD\_CLIENTSERVICE\_CONSUMEDEVENTGROUPSETSTATE\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_consumedEventGroupSetState.
----------------	---



<b>Value</b>	0x1EU
--------------	-------

#### 5.5.2.1.18. SD\_CLIENTSERVICE\_EVENT\_OFFER\_SERVICE\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_eventOfferService.
<b>Value</b>	0x1CU

#### 5.5.2.1.19. SD\_CLIENTSERVICE\_PROCESSOFFERS\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_ProcessOffers.
<b>Value</b>	0x1DU

#### 5.5.2.1.20. SD\_CLIENTSERVICE\_TTL\_RUN\_OUT

<b>Purpose</b>	Defines API id of function Sd_ClientService_TTL_Run_Out.
<b>Value</b>	0x0CU

#### 5.5.2.1.21. SD\_COMPAREADDRESS\_SVCID

<b>Purpose</b>	Defines API id of function Sd_LocallpAddrAssignmentChg.
<b>Value</b>	0x04U

#### 5.5.2.1.22. SD\_CONNECTION\_SETTING\_FAILED

<b>Purpose</b>	service called with invalid mode request
<b>Value</b>	0x0BU

#### 5.5.2.1.23. SD\_CONSUMEDEVENTGROUPSETSTATE\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ConsumedEventGroupSetState.
----------------	---



<b>Value</b>	0x09U
--------------	-------

#### 5.5.2.1.24. SD\_CONTROL\_SOCKET\_IMPOSSIBLE

<b>Purpose</b>	service called with invalid mode request
<b>Value</b>	0x0AU

#### 5.5.2.1.25. SD\_CONTROL\_SOCKET\_NOT\_RELEASED

<b>Purpose</b>	service called with invalid mode request
<b>Value</b>	0x09U

#### 5.5.2.1.26. SD\_E\_CLIENT\_IMPOSSIBLE\_STATE

<b>Purpose</b>	Initialization failed error.
<b>Value</b>	0x07U

#### 5.5.2.1.27. SD\_E\_COUNT\_OF\_RETRY\_SUBSCRIPTION\_EXCEEDED

<b>Purpose</b>	Initialization failed error.
<b>Value</b>	0x06U

#### 5.5.2.1.28. SD\_E\_DIVISOR\_IS\_NEGATIVE

<b>Purpose</b>	Divisor is negative error.
<b>Value</b>	0x0DU

#### 5.5.2.1.29. SD\_E\_INVALID\_ARG

<b>Purpose</b>	Initialization failed error.
----------------	------------------------------



<b>Value</b>	0x03U
--------------	-------

#### 5.5.2.1.30. SD\_E\_INV\_ID

<b>Purpose</b>	API service called with invalid Id.
<b>Value</b>	0x04U

#### 5.5.2.1.31. SD\_E\_INV\_MODE

<b>Purpose</b>	API service called with invalid mode request.
<b>Value</b>	0x05U

#### 5.5.2.1.32. SD\_E\_IPADDR\_ASSIGNMENT\_NOT\_FINISHED

<b>Purpose</b>	IP Address assignment not finished error.
<b>Value</b>	0x0CU

#### 5.5.2.1.33. SD\_E\_NOT\_INITIALIZED

<b>Purpose</b>	API service called before initializing the module.
<b>Value</b>	0x01U

#### 5.5.2.1.34. SD\_E\_OUT\_OF\_RES

<b>Purpose</b>	Initialization failed error.
<b>Value</b>	0x08U

#### 5.5.2.1.35. SD\_E\_PARAM\_POINTER

<b>Purpose</b>	Null pointer has been passed as an argument.
<b>Value</b>	0x02U



#### 5.5.2.1.36. SD\_E\_RETRY\_INFINITE\_TTL\_NOT\_INFINITE

<b>Purpose</b>	IP Address assignment not finished error.
<b>Value</b>	0x0EU

#### 5.5.2.1.37. SD\_E\_RETRY\_PROCESS\_BIGGER\_THAN\_TTL

<b>Purpose</b>	Divisor is negative error.
<b>Value</b>	0x0FU

#### 5.5.2.1.38. SD\_E\_SHOULD\_NOT\_HAPPEN

<b>Purpose</b>	Definition of defensive programming SD_E_SHOULD_NOT_HAPPEN.
<b>Value</b>	0xffffU

#### 5.5.2.1.39. SD\_GETRANDOM\_SVCID

<b>Purpose</b>	Defines API id of function Sd_Get_Random.
<b>Value</b>	0x60U

#### 5.5.2.1.40. SD\_GETVERSIONINFO\_SVCID

<b>Purpose</b>	Defines API id of function Sd_GetVersionInfo.
<b>Value</b>	0x02U

#### 5.5.2.1.41. SD\_HANDLERESPONSETIMERS\_SVCID

<b>Purpose</b>	Defines API id of function Sd_HandleResponseTimers.
<b>Value</b>	0x62U

#### 5.5.2.1.42. SD\_INIT\_SVCID

<b>Purpose</b>	Defines API id of function Sd_Init.
----------------	-------------------------------------



<b>Value</b>	0x01U
--------------	-------

#### 5.5.2.1.43. SD\_INSTANCEMAINFUNCTION\_SVCID

<b>Purpose</b>	Defines API id of function Sd_Instance_MainFunction.
<b>Value</b>	0x0AU

#### 5.5.2.1.44. SD\_INSTANCE\_ID

<b>Purpose</b>	AUTOSAR module identification.
<b>Value</b>	0U

#### 5.5.2.1.45. SD\_INTERNAL\_API\_ID

<b>Purpose</b>	API ID of module internal functions to be used in assertions.
<b>Value</b>	DET_INTERNAL_API_ID

#### 5.5.2.1.46. SD\_INVARIANT\_ASSERT

<b>Purpose</b>	Report an assertion violation to Det.
<b>Value</b>	DET_INVARIANT_ASSERT((Condition), SD_MODULE_ID, SD_INSTANCE_ID, (Apild))

#### 5.5.2.1.47. SD\_INVARIANT\_ASSERT\_NO\_EVAL

<b>Purpose</b>	Report an assertion violation to Det.
<b>Value</b>	DET_INVARIANT_ASSERT_NO_EVAL((Condition), SD_MODULE_ID, SD_INSTANCE_ID, (Apild))

#### 5.5.2.1.48. SD\_LOCALIPADDRASSIGNMENTCHG\_SVCID

<b>Purpose</b>	Defines API id of function Sd_LocalIpAddrAssignmentChg.
----------------	---



<b>Value</b>	0x05U
--------------	-------

#### 5.5.2.1.49. SD\_MAINFUNCTION\_SVCID

<b>Purpose</b>	Defines API id of function Sd_MainFunction.
<b>Value</b>	0x06U

#### 5.5.2.1.50. SD\_POSTCONDITION\_ASSERT

<b>Purpose</b>	Report an assertion violation to Det.
<b>Value</b>	DET_POSTCONDITION_ASSERT((Condition), SD_MODULE_ID, SD_INSTANCE_ID, (Apild))

#### 5.5.2.1.51. SD\_POSTCONDITION\_ASSERT\_NO\_EVAL

<b>Purpose</b>	Report an assertion violation to Det.
<b>Value</b>	DET_POSTCONDITION_ASSERT_NO_EVAL((Condition), SD_MODULE_ID, SD_INSTANCE_ID, (Apild))

#### 5.5.2.1.52. SD\_PRECONDITION\_ASSERT

<b>Purpose</b>	Report an assertion violation to Det.
<b>Value</b>	DET_PRECONDITION_ASSERT((Condition), SD_MODULE_ID, SD_INSTANCE_ID, (Apild))

#### 5.5.2.1.53. SD\_PRECONDITION\_ASSERT\_NO\_EVAL

<b>Purpose</b>	Report an assertion violation to Det.
<b>Value</b>	DET_PRECONDITION_ASSERT_NO_EVAL((Condition), SD_MODULE_ID, SD_INSTANCE_ID, (Apild))

#### 5.5.2.1.54. SD\_RECEIVEGETENTRYOPTIONS\_SVCID

<b>Purpose</b>	Defines API id of function Sd_Receive_getEntryOptions.
----------------	--



<b>Value</b>	0x50U
--------------	-------

#### 5.5.2.1.55. SD\_RESETREMOTENODERXSTATE\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ResetRemoteNodeRxState.
<b>Value</b>	0x61U

#### 5.5.2.1.56. SD\_RXINDICATION\_SVCID

<b>Purpose</b>	Defines API id of function Sd_RxIndication.
<b>Value</b>	0x42U

#### 5.5.2.1.57. SD\_SENDQUEUEOPTIONS\_SVCID

<b>Purpose</b>	Defines API id of function Sd_Send_queueOptions.
<b>Value</b>	0x43U

#### 5.5.2.1.58. SD\_SEND\_FLUSHBUFFER\_SVCID

<b>Purpose</b>	Defines API id of function Sd_Send_flushBuffer.
<b>Value</b>	0x45U

#### 5.5.2.1.59. SD\_SERVERSERVICEADDEVENTGROUPSUBS\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ServerService_addEventGroupSubscriber.
<b>Value</b>	0x29U

#### 5.5.2.1.60. SD\_SERVERSERVICEADDMULTEVENTGROUPSUBS\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ServerService_addMulticastEventGroupSubscriber.
----------------	---



<b>Value</b>	0x28U
--------------	-------

#### 5.5.2.1.61. SD\_SERVERSERVICEEVENTSTOPSUBSCRIBE\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ServerService_eventStopSubscribe.
<b>Value</b>	0x27U

#### 5.5.2.1.62. SD\_SERVERSERVICEHANDLETTL\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_mainTTL.
<b>Value</b>	0x24U

#### 5.5.2.1.63. SD\_SERVERSERVICEMAINFUNCTION\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_mainFunction.
<b>Value</b>	0x25U

#### 5.5.2.1.64. SD\_SERVERSERVICEMAITIMERS\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ClientService_mainTimersFunction.
<b>Value</b>	0x23U

#### 5.5.2.1.65. SD\_SERVERSERVICEREMOVEEVENTGROUPSUBS\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ServerService_removeEventGroupSubscriber.
<b>Value</b>	0x2AU

#### 5.5.2.1.66. SD\_SERVERSERVICERESETREMOTECONNECTION\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ServerService_resetRemoteConnection.
----------------	--



<b>Value</b>	0x26U
--------------	-------

#### 5.5.2.1.67. SD\_SERVERSERVICESETSTATE\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ServerServiceSetState.
<b>Value</b>	0x07U

#### 5.5.2.1.68. SD\_SERVERSERVICE\_COMSMEVENT\_START\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ServerService_ComSM_event_start.
<b>Value</b>	0x20U

#### 5.5.2.1.69. SD\_SERVERSERVICE\_COMSMEVENT\_STOPHALT\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ServerService_ComSM_event_stophalt.
<b>Value</b>	0x21U

#### 5.5.2.1.70. SD\_SERVERSERVICE\_COMSMEVENT\_TIMEOUT\_SVCID

<b>Purpose</b>	Defines API id of function Sd_ServerService_ComSM_event_timeout.
<b>Value</b>	0x22U

#### 5.5.2.1.71. SD\_STATIC\_ASSERT

<b>Purpose</b>	Report an static assertion violation to Det.
<b>Value</b>	DET_STATIC_ASSERT(expr)

#### 5.5.2.1.72. SD\_UNREACHABLE\_CODE\_ASSERT

<b>Purpose</b>	Report an unreachable code assertion violation to Det.
<b>Value</b>	DET_UNREACHABLE_CODE_ASSERT(SD_MODULE_ID, SD_INSTANCE_ID, (Apild))



### 5.5.2.2. Functions

#### 5.5.2.2.1. Sd\_ClientServiceSetState

Purpose	Sets the Client Service instance state.	
Synopsis	<pre>Std_ReturnType <b>Sd_ClientServiceSetState</b> ( uint16 ClientServiceHandleId , Sd_ClientServiceSetStateType ClientServiceState );</pre>	
Service ID	0x08	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	ClientServiceHandleId	ID to identify the Client Service instance.
	ClientServiceState	The state the Client Service instance shall be set to.
Return Value	Result of function call	
	E_OK	State accepted
	E_NOT_OK	State not accepted
Description	This API function sets the Client Service instance state.	

#### 5.5.2.2.2. Sd\_ConsumedEventGroupSetState

Purpose	Sets the Client EventGroup state.	
Synopsis	<pre>Std_ReturnType <b>Sd_ConsumedEventGroupSetState</b> ( uint16 ConsumedEventGroupHandleId , Sd_ConsumedEventGroupSetStateType ConsumedEventGroupState );</pre>	
Service ID	0x09	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	ConsumedEventGroupHandleId	ID to identify the Client EventGroup instance.
	ConsumedEventGroupState	The state the Client EventGroup instance shall be set to.
Return Value	Result of function call	
	E_OK	State accepted
	E_NOT_OK	State not accepted



<b>Description</b>	This API function sets the Client EventGroup state.
--------------------	---

#### 5.5.2.2.3. Sd\_GetVersionInfo

<b>Purpose</b>	Get version information of the Sd module.	
<b>Synopsis</b>	<pre>void <b>Sd_GetVersionInfo</b> ( Std_VersionInfoType * Sd_VersionInfo );</pre>	
<b>Service ID</b>	0x0B	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (out)</b>	versioninfo	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>	

#### 5.5.2.2.4. Sd\_Init

<b>Purpose</b>	Initializes the Sd module.
<b>Synopsis</b>	<pre>void <b>Sd_Init</b> ( const Sd_ConfigType * ConfigPtr );</pre>
<b>Service ID</b>	0x01
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	This service initializes Sd module internal variables.

#### 5.5.2.2.5. Sd\_IsValidConfig

<b>Purpose</b>	Checks compatibility of the post-build-time configuration.
<b>Synopsis</b>	<pre>Std_ReturnType <b>Sd_IsValidConfig</b> ( const void * voidConfigPtr );</pre>



<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	voidConfigPtr	Pointer to the configuration data of the Sd module.
<b>Return Value</b>	Result of compatibility check	
	E_OK	Provided configuration is compatible.
	E_NOT_OK	Provided configuration is not compatible.
<b>Description</b>	This service checks the compatibility of the post-build-time configuration against the source code.	

#### 5.5.2.2.6. Sd\_LocalIpAddrAssignmentChg

<b>Purpose</b>	Local IP address assignment change indication.	
<b>Synopsis</b>	<pre>void <b>Sd_LocalIpAddrAssignmentChg</b> ( SoAd_SoConIdType SoConId ,     TcpIp_IpAddrStateType State );</pre>	
<b>Service ID</b>	0x05	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SoConId	Socket connection index specifying the socket connection where the IP address assignment has changed.
	State	State of IP address assignment.
<b>Description</b>	This function gets called by the SoAd if an IP address assignment related to a socket connection changes (i.e. new address assigned or assigned address becomes invalid).	

#### 5.5.2.2.7. Sd\_MainFunction

<b>Purpose</b>	Main Function of module Sd.
<b>Synopsis</b>	<pre>void <b>Sd_MainFunction</b> ( void );</pre>
<b>Service ID</b>	0x06
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant



<b>Description</b>	This API triggers all periodic actions to be performed by Sd.
--------------------	---

#### 5.5.2.2.8. Sd\_RxIndication

<b>Purpose</b>	Sd Rx Indication.	
<b>Synopsis</b>	<code>void Sd_RxIndication ( PduIdType RxPduId , PduInfoType * PduInfoPtr );</code>	
<b>Service ID</b>	0x42	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	RxPduId	ID of the received I-PDU.
	PduInfoPtr	Contains the length (SduLength) of the received I-PDU and a pointer to a buffer (SduDataPtr) containing the I-PDU.
<b>Description</b>	Indication of a received I-PDU from a lower layer communication interface module.	

#### 5.5.2.2.9. Sd\_ServerServiceSetState

<b>Purpose</b>	Sets the Server Service instance state.	
<b>Synopsis</b>	<code>Std_ReturnType Sd_ServerServiceSetState ( uint16 ServiceServiceHandleId , Sd_ServerServiceSetStateType ServerServiceState );</code>	
<b>Service ID</b>	0x07	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	ServiceServiceHandleId	ID to identify the Server Service instance.
	ServerServiceState	The state the Server Service instance shall be set to.
<b>Return Value</b>	Result of function call	
	E_OK	State accepted
	E_NOT_OK	State not accepted
<b>Description</b>	This API function sets the Server Service instance state.	



### 5.5.2.2.10. Sd\_StartRandom

<b>Purpose</b>	Initialize the random number generator.	
<b>Synopsis</b>	<code>void Sd_StartRandom ( uint32 Seed );</code>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	Seed	Start value for number generator
<b>Parameters (in,out)</b>	Seed	Start value for number generator
<b>Description</b>	Initialize the random number generator for random delays.	

## 5.5.3. Integration notes

### 5.5.3.1. Exclusive areas

This section describes the exclusive areas used by the Sd module.

#### 5.5.3.1.1. SCHM\_SD\_EXCLUSIVE\_AREA\_0

<b>Protected data structures</b>	All shared data that shall be protected from mutual access.
<b>Recommended locking mechanism</b>	<p>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.</p> <p>The Sd uses a critical section to protect its internal queue operations. Map this exclusive area to an interrupt locking mechanism, e.g. ALL_INTERRUPT_BLOCKING in the Rte module configuration.</p>

### 5.5.3.2. Production errors

Production errors are not reported by the SD 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:

Memory section
VAR_CLEARED_8
CONST_8
VAR_INIT_8
VAR_CLEARED_16
VAR_INIT_32
VAR_CLEARED_UNSPECIFIED
CONST_UNSPECIFIED
CONFIG_DATA_UNSPECIFIED
CODE

### 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's guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

#### 5.5.3.4.1. Sd.EB.IntReq.EB\_INTREQ\_Sd\_0001

<b>Description</b>	The integrator must assure that the following functions do not interrupt each other or themselves: <ul style="list-style-type: none"><li>▶ Sd_MainFunction</li><li>▶ Sd_RxIndication</li><li>▶ Sd_LocallpAddrAssignmentChg</li><li>▶ Sd_StartRandom</li></ul>
<b>Rationale</b>	This limitation reduces code size and execution time.



#### 5.5.3.4.2. Sd.EB.IntReq.EB\_INTREQ\_Sd\_0002

<b>Description</b>	The integrator must assure that the following functions do not interrupt each other or themselves: <ul style="list-style-type: none"><li>▶ Sd_ServerServiceSetState</li><li>▶ Sd_ClientServiceSetStat</li><li>▶ Sd_ConsumedEventGroupSetState</li></ul>
<b>Rationale</b>	This limitation reduces code size and execution time.

#### 5.5.3.4.3. Sd.EB.IntReq.EB\_INTREQ\_Sd\_0003

<b>Description</b>	The integrator must assure that all Client services using the same SocketConnection are provided on the same endpoint.
<b>Rationale</b>	The RemoteAddress of a socket connection will be overwritten if an OfferService entry is received for different Services that use the same Socket connection.

## 5.6. SoAd

### 5.6.1. 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="#">PublishedInformation</a>	1..1	<b>Label:</b> EB Published Information Additional published parameters not covered by CommonPublishedInformation container.
<a href="#">SoAdDefensiveProgramming</a>	1..1	<b>Label:</b> Defensive Programming Options Parameters for defensive programming
<a href="#">SoAdGeneral</a>	1..1	This container contains all global configuration parameters of SoAd.

**Containers included**

<a href="#">SoAdBswModules</a>	0..n	Each container describes a specific BSW module that the SoAd shall interface to. The reason to have it as own configuration container instead of implication of the routing path is to be able to configure CDDs properly and to force modules to be used in a post-build situation even though no routing is made to/from this module (future configurations may include these modules).
<a href="#">SoAdConfig</a>	1..n	This container contains the configuration parameters and subcontainers of the AUTOSAR SoAd module. This container is a MultipleConfigurationContainer, i.e. this container and its subcontainers exist once per configuration set.

**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>	Config Variant
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	VariantPostBuild
<b>Range</b>	VariantPostBuild

**5.6.1.1. CommonPublishedInformation****Parameters included**

Parameter name	Multiplicity
<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>	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>	2
<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>	8
<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>	18
<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>	56



<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

### 5.6.1.2. 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 SoAd 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	

### 5.6.1.3. SoAdDefensiveProgramming

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">SoAdDefProgEnabled</a>	1..1
<a href="#">SoAdPrecondAssertEnabled</a>	1..1
<a href="#">SoAdPostcondAssertEnabled</a>	1..1
<a href="#">SoAdStaticAssertEnabled</a>	1..1
<a href="#">SoAdUnreachAssertEnabled</a>	1..1
<a href="#">SoAdInvariantAssertEnabled</a>	1..1

<b>Parameter Name</b>	<b>SoAdDefProgEnabled</b>
<b>Label</b>	Enable Defensive Programming
<b>Description</b>	<p>Enables or disables the defensive programming feature for the module SoAd.</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>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SoAdPrecondAssertEnabled</b>
<b>Label</b>	Enable Precondition Assertions
<b>Description</b>	<p>Enables handling of precondition assertion checks reported from the module SoAd.</p> <p>Dependency on parameter(s):</p>



	<ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>SoAdDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>SoAdDefProgEnabled</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>SoAdPostcondAssertEnabled</b>
<b>Label</b>	Enable Postcondition Assertions
<b>Description</b>	<p>Enables handling of postcondition assertion checks reported from the module SoAd.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>SoAdDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>SoAdDefProgEnabled</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>SoAdStaticAssertEnabled</b>
<b>Label</b>	Enable Static Assertions
<b>Description</b>	<p>Enables handling of static assertion checks reported from the module SoAd.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>SoAdDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>SoAdDefProgEnabled</code>): must be enabled</li> </ul>
<b>Multiplicity</b>	1..1



Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>SoAdUnreachAssertEnabled</b>	
Label	Enable Unreachable Code Assertions	
Description	<p>Enables handling of unreachable code assertion checks reported from the module SoAd.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>SoAdDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>SoAdDefProgEnabled</code>): must be enabled</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>SoAdInvariantAssertEnabled</b>	
Label	Enable Invariant Assertions	
Description	<p>Enables handling of invariant assertion checks reported from functions of the module SoAd.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>SoAdDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>SoAdDefProgEnabled</code>): must be enabled</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild

<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

### 5.6.1.4. SoAdGeneral

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">VendorSpecific</a>	1..1	Contains the vendor specific configuration parameters of the AUTOSAR SoAd module.

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">SoAdDevErrorDetect</a>	1..1
<a href="#">SoAdGetAndResetMeasurementDataApi</a>	1..1
<a href="#">SoAdIPv6AddressEnabled</a>	1..1
<a href="#">SoAdMainFunctionPeriod</a>	1..1
<a href="#">SoAdSoConMax</a>	1..1
<a href="#">SoAdRoutingGroupMax</a>	1..1
<a href="#">SoAdVersionInfoApi</a>	1..1
<a href="#">SoAdTlsEnabled</a>	1..1
<a href="#">SoAdMainFunctionPeriodTx</a>	1..1
<a href="#">SoAdEnableMainFunctionTx</a>	1..1

<b>Parameter Name</b>	<b>SoAdDevErrorDetect</b>	
<b>Description</b>	Pre-processor switch for enabling development error detection support.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	true	
<b>Configuration class</b>	<a href="#">VariantPostBuild:</a>	<a href="#">VariantPostBuild</a>
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SoAdGetAndResetMeasurementDataApi</b>	
<b>Description</b>	Pre-processor switch for enabling the Get and Reset Measurement Data API.	
<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>SoAdIPv6AddressEnabled</b>
<b>Description</b>	Allows for increased memory allocation to store IPv6 addresses. <ul style="list-style-type: none"> <li>▶ true: Enables support for IPv6 addresses</li> <li>▶ false: Only IPv4 addresses are supported</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>SoAdMainFunctionPeriod</b>
<b>Description</b>	Determines the frequency at which the SoAd_MainFunction() is called in [s].
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.1
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>SoAdSoConMax</b>
<b>Description</b>	Specifies the maximum number of SoAd socket connections.  <b>Parameter is unused. The maximum number of socket connections is defined by the amount of reserved RAM (to be configured with SoAd-DataMemSize) and the used type (to be configured with SoAdSoConId-Type).</b>
<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>SoAdRoutingGroupMax</b>
-----------------------	----------------------------



<b>Description</b>	Specifies the maximum number of SoAd routing groups.
<b>Parameter is unused. The maximum number of routing groups is defined by the amount of reserved RAM (to be configured with SoAdDataMemSize) and the used type (to be configured with SoAdRoutingGroupIdType).</b>	
<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>SoAdVersionInfoApi</b>
<b>Description</b>	Activates the SoAd_GetVersionInfo() API. <ul style="list-style-type: none"> <li>▶ true: Enables the SoAd_GetVersionInfo() API.</li> <li>▶ false: SoAd_GetVersionInfo() API is not included.</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>SoAdTlsEnabled</b>
<b>Description</b>	<ul style="list-style-type: none"> <li>▶ true: Enables support for TLS extension.</li> <li>▶ false: Disables support for TLS extension.</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>SoAdMainFunctionPeriodTx</b>
<b>Description</b>	Determines the frequency at which the SoAd_MainFunctionTx() is called in [s]. The calculated value is set to the SoAdMainFunctionPeriod which is the default cycle time for SoAd_MainFunctionTx(). The parameter is needed only when SoAd Tx main function is called additionally and depends on SoAdEnableMainFunctionTx (must be set to true).



<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SoAdEnableMainFunctionTx</b>
<b>Description</b>	This configuration parameter enables the use of SoAd_MainFunctionTx() to be triggered externally. SoAd_MainFunctionTx() triggers the timing-independent Tx frame processing of SoAd_MainFunction().  ▶ true: SoAd_MainFunctionTx() can be called externally. ▶ false: SoAd_MainFunctionTx() is defined internally only.
<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

### 5.6.1.5. VendorSpecific

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">SoAdDataMemSize</a>	0..1
<a href="#">SoAdDataMemSizeExtension</a>	1..1
<a href="#">SoAdRelocatableCfgEnable</a>	1..1
<a href="#">SoAdRouteMax</a>	1..1
<a href="#">SoAdTimerType</a>	1..1
<a href="#">SoAdSoConIdType</a>	1..1
<a href="#">SoAdRoutingGroupIdType</a>	1..1
<a href="#">SoAdReportInvPDUHeaderIdToDetEnable</a>	1..1
<a href="#">SoAdSocketRouteDestFanInEnable</a>	1..1

<b>Parameter Name</b>	<b>SoAdDataMemSize</b>
<b>Description</b>	Size of internal SoAd data in units of bytes (static memory allocation). The memory required by a post-build configuration must be smaller than this constant. If



	the parameter is disabled, the module configuration generator calculates the value itself based on the current configuration.  <b>This parameter shall be used to reserve RAM at precompile time to be used for configuration changes at post-build time.</b>
<b>Multiplicity</b>	0..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Configuration class</b>	<b>Link:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

Parameter Name	SoAdDataMemSizeExtension
<b>Description</b>	Configuration parameter to allow RAM size usage of more than 64 KiB.  ▶ TRUE: RAM size NOT limited by 64 KiB at maximum (switched on). ▶ FALSE: RAM size limited by 64 KiB at maximum (switched off).  <b>Optimization Effect:</b>  ▶ <b>ROM reduction (config):</b> Disabling this parameter reduces the ROM consumption of the module configuration.
<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

Parameter Name	SoAdRelocatableCfgEnable
<b>Description</b>	If set to TRUE, the post-build configuration data contains only relative offsets to the configuration start address (and is therefore relocatable). If set to FALSE, the post-build configuration data contains absolute pointers (and is therefore not relocatable).
<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>SoAdRouteMax</b>
<b>Description</b>	<p>This parameter defines the maximum number of SoAdPduRouteDest containers and the maximum number of SoAdSocketRouteDest containers.</p> <ul style="list-style-type: none"> <li>▶ INDEX_UINT8: The maximum number of SoAdPduRouteDest containers is 255 and the maximum number of SoAdSocketRouteDest containers is 255.</li> <li>▶ INDEX_UINT16: The maximum number of SoAdPduRouteDest containers is 65535 and the maximum number of SoAdSocketRouteDest containers is 65535.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> The smaller the index, the smaller the ROM consumption of the module configuration.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	INDEX_UINT16
<b>Range</b>	INDEX_UINT8 INDEX_UINT16
<b>Configuration class</b>	VariantPostBuild: VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SoAdTimerType</b>
<b>Description</b>	<p>SoAdTimerType defines the maximum range for timers and time-outs that can be used in the configuration of SoAd (e.g. for the nPduUdpTx feature) by determining the size of SoAd_TimerType. The configurable timer/time-out values also depend on the configuration of SoAdMainFunctionPeriod.</p> <p><b>Example:</b></p> <p>If SoAdMainFunctionPeriod is set to 10 ms, the following SoAdTimerType settings allow the respective timer/time-out values:</p> <ul style="list-style-type: none"> <li>▶ UINT8: values up to 2.55 s</li> <li>▶ UINT16: values up to 655.35 s (~11 min)</li> <li>▶ UINT32: values up to 42949672.95 s (well over one year)</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> The smaller the type, the smaller the ROM consumption of the module configuration.</li> </ul>



<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	UINT8	
<b>Range</b>	UINT8 UINT16 UINT32	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>SoAdSoConIdType</b>			
<b>Description</b>	<p>SoAdSoConIdType allows the optimization of the memory consumption of the socket connections. Changing this value has an impact on the number of socket connection that can be configured at post-build time.</p> <p><b>Possible values:</b></p> <ul style="list-style-type: none"> <li>▶ UINT8: The maximum number of connections is 254.</li> <li>▶ UINT16: The maximum number of connections is 65534.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> The smaller the type, the smaller the ROM consumption of the module configuration.</li> <li>▶ <b>RAM reduction (config):</b> The smaller the type, the smaller the RAM consumption of the module.</li> </ul>			
<b>Multiplicity</b>				
<b>Type</b>				
<b>Default value</b>				
<b>Range</b>	UINT8 UINT16			
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild		
<b>Origin</b>	Elektrobit Automotive GmbH			

<b>Parameter Name</b>	<b>SoAdRoutingGroupIdType</b>	
<b>Description</b>	<p>SoAdRoutingGroupIdType allows the optimization of the memory consumption of the routing groups. Changing this value has an impact on the number of routing groups that can be configured at post-build time.</p>	



	<p><b>The number of routing groups can be higher than the maximum of SoAd- RoutingGroupId if specific routing is used.</b></p> <p><b>Possible values:</b></p> <ul style="list-style-type: none"> <li>▶ <b>UINT8:</b> The maximum number of connections is 254.</li> <li>▶ <b>UINT16:</b> The maximum number of connections is 65534.</li> <li>▶ <b>UINT32:</b> The maximum number of connections is 4294967295.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> The smaller the type, the smaller the ROM con- sumption of the module configuration.</li> <li>▶ <b>RAM reduction (config):</b> The smaller the type, the smaller the RAM con- sumption of the module.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	UINT16
<b>Range</b>	UINT8 UINT16 UINT32
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>SoAdReportInvPDUHeaderIdToDetEnable</b>
<b>Description</b>	<p>Enables the reporting of SOAD_E_INV_PDUHEADER_ID to DET whenever an invalid PDU header ID is received and DET reporting is enabled (i.e. SoAdDev-ErrorDetect = TRUE).</p> <ul style="list-style-type: none"> <li>▶ <b>true:</b> Report of SOAD_E_INV_PDUHEADER_ID to DET.</li> <li>▶ <b>false:</b> No report to DET.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (code):</b> Setting this parameter to FALSE reduces the ROM consumption of the module code.</li> <li>▶ <b>Execution time reduction (code):</b> Setting this parameter to FALSE re- duces the execution time of the module code.</li> </ul>
<b>Multiplicity</b>	1..1



Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>SoAdSocketRouteDestFanInEnable</b>	
Description	<p>Determines whether it shall be allowed that a global PDU can be referenced by more than one SoAdSocketRouteDest with SoAdRxUpperLayerType = 'IF'.</p> <ul style="list-style-type: none"> <li>▶ true: Global PDUs can be referenced by multiple SoAdSocketRouteDest (SoAdRxUpperLayerType = 'IF').</li> <li>▶ false: Global PDUs can only be referenced by one SoAdSocketRouteDest (SoAdRxUpperLayerType = 'IF').</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

### 5.6.1.6. SoAdBswModules

Parameters included	
Parameter name	Multiplicity
<a href="#">SoAdIf</a>	1..1
<a href="#">SoAdTp</a>	1..1
<a href="#">SoAdIfTriggerTransmit</a>	1..1
<a href="#">SoAdIfTxConfirmation</a>	1..1
<a href="#">SoAdSoConModeChg</a>	1..1
<a href="#">SoAdLocallIpAddrAssignmentChg</a>	1..1
<a href="#">SoAdUseCallerInfix</a>	1..1
<a href="#">SoAdUseTypeInfix</a>	1..1
<a href="#">SoAdBswModuleRef</a>	1..1

Parameter Name	<b>SoAdIf</b>
----------------	---------------



<b>Description</b>	Specifies if the BSW module supports the Communication Interface APIs or not. If the value is true, the APIs are supported. A module can have both Communication Interface APIs and Transport Protocol APIs (e.g. the PduR module).	
<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>SoAdTp</b>	
<b>Description</b>	Specifies if the BSW module supports the TransportProtocol APIs or not. If the value is true, the APIs are supported. A module can have both Communication Interface APIs and Transport Protocol APIs (e.g. the PduR module).	
<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>SoAdIfTriggerTransmit</b>	
<b>Description</b>	Specifies if the BSW module supports the TriggerTransmit API or not. If the value is true, the API is supported.	
<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>SoAdIfTxConfirmation</b>	
<b>Description</b>	Specifies if the BSW module supports the TxConfirmation API or not. If the value is true, the API is supported.	
<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>SoAdSoConModeChg</b>	
<b>Description</b>	Specifies if the BSW module supports the SoConModeChg API or not. If the value is true, the API is supported.	
<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>SoAdLocallpAddrAssigmentChg</b>	
<b>Description</b>	Specifies if the BSW module supports the LocallpAddrAssigmentChg API or not. If the value is true, the API is supported.	
<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>SoAdUseCallerInfix</b>	
<b>Description</b>	Specifies if SoAd shall use the infix "SoAd" when calling an upper layer module function.  E.g. if SoAdUseCallerInfix is TRUE for the upper layer "ABC", then SoAd calls ABC_SoAdIfRxIndication(). Otherwise SoAd would call ABC_IfRxIndication().  ▶ true: CallerInfix is used. ▶ false: CallerInfix is not used.	
	<b>This parameter is not used.</b>	
<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>SoAdUseTypeInfix</b>
<b>Description</b>	<p>Specifies if SoAd shall use the API type infix "TP" or "IF" when calling an upper layer module function.</p> <p>E.g. if SoAdUseTypeInfix is TRUE for the upper layer "ABC", then SoAd calls ABC_IfRxIndication(). Otherwise SoAd would call ABC_RxIndication().</p> <ul style="list-style-type: none"> <li>▶ true: TypeInfix is used.</li> <li>▶ false: TypeInfix is not used.</li> </ul> <p><b>This parameter is not used.</b></p>
<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>SoAdBswModuleRef</b>
<b>Description</b>	<p>This is a reference to one BSW module's configuration (i.e. not the ECUC parameter definition template). For example, there could be several configurations of LinIf and this reference selects one of them.</p> <p><b>This parameter is not used.</b></p>
<b>Multiplicity</b>	1..1
<b>Type</b>	FOREIGN-REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.6.1.7. SoAdConfig

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">SoAdPduRoute</a>	0..n	Describes the path of a PDU from an upper layer of the SoAd to the socket in the TCP/IP stack for transmission.
<a href="#">SoAdSocketConnection-Group</a>	1..n	Specifies the configuration of a socket connection group, i.e. it specifies the socket connections belonging to the group and the parameters that are common for all socket connec-



#### Containers included

		tions of the group. A socket connection specifies how data can be received and transmitted via a TCP or UDP socket.
<a href="#">SoAdSocketRoute</a>	0..n	Describes the path of a PDU from a socket in the TCP/IP stack to an upper layer of the SoAd after reception in the TCP/IP Stack.
<a href="#">SoAdRoutingGroup</a>	0..n	Specifies a routing group that can be activated or deactivated for PDU routing. A routing group consists of PDUs. Routing of PDUs can either be forwarding of PDUs from the upper layer to a TCP or UDP socket of the TCP/IP stack specified by a SoAdPduRoute or the other way around specified by a SoAdSocketRoute.

#### 5.6.1.8. SoAdPduRoute

##### Containers included

Container name	Multiplicity	Description
<a href="#">SoAdPduRouteDest</a>	1..n	Describes the upper layer destination PDU for a message received on a Tcplp socket.

##### Parameters included

Parameter name	Multiplicity
<a href="#">SoAdTxPduld</a>	1..1
<a href="#">SoAdTxPduRef</a>	1..1
<a href="#">SoAdTxUpperLayerType</a>	1..1
<a href="#">SoAdTxPduCollectionSemantics</a>	1..1
<a href="#">SoAdSkipIfTxConfirmation</a>	1..1

##### Parameter Name

##### SoAdTxPduld

**Description** Tx PDU ID of the PDU coming from the PDU Router.

**Multiplicity** 1..1

**Type** INTEGER

**Default value** 0

**Configuration class** VariantPostBuild: VariantPostBuild

**Origin** AUTOSAR\_ECUC



<b>Parameter Name</b>	<b>SoAdTxPduRef</b>	
<b>Description</b>	Reference to the global PDU structure.	
<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>SoAdTxUpperLayerType</b>	
<b>Description</b>	Specifies the upper layer interface type (must be "IF" in case of multiple PduRoutes).	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	IF	
<b>Range</b>	IF TP	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SoAdTxPduCollectionSemantics</b>	
<b>Description</b>	<p>Specifies if this PDU shall be collected using a queued or last-is-best semantics.</p> <p>This parameter is only relevant if the PDU collection feature is enabled.</p> <p>This feature gets enabled if SoAdSocketnPduUdpTxBufferMin is used and SoAdTxUpperLayerType is set to 'IF'.</p> <p>All PDU routes referring to the same socket connection or socket connection group shall have the same semantics, i.e. either SOAD_COLLECT_LAST_IS_BEST or SOAD_COLLECT_QUEUED.</p> <p><b>This parameter shall only be set to SOAD_COLLECT_LAST_IS_BEST if the related upper layer is configured with SoAdIfTriggerTransmit set to TRUE.</b></p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	SOAD_COLLECT_QUEUED	
<b>Range</b>	SOAD_COLLECT_LAST_IS_BEST SOAD_COLLECT_QUEUED	



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>SoAdSkipIfTxConfirmation</b>	
<b>Description</b>	<p>Enables or disables the Tx confirmation for the SoAd_IfTransmit() call.</p> <ul style="list-style-type: none"> <li>▶ TRUE: Tx confirmation gets skipped for this PDU.</li> <li>▶ FALSE: The Tx confirmation API is called for this PDU if the transmission was successful.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>Run-time reduction:</b> Enabling this parameter reduces the run-time consumption of the module's Tx confirmations.</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	

### 5.6.1.9. SoAdPduRouteDest

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">SoAdTxPduHeaderId</a>	0..1
<a href="#">SoAdTxSocketConnOrSocketConnBundleRef</a>	1..1
<a href="#">SoAdTxRoutingGroupRef</a>	0..n
<a href="#">SoAdTxUdpTriggerMode</a>	1..1
<a href="#">SoAdTxUdpTriggerTimeout</a>	0..1

<b>Parameter Name</b>	<b>SoAdTxPduHeaderId</b>
<b>Description</b>	ID to be sent on the TCP/IP connection if the PDU header option is enabled.
<b>Multiplicity</b>	0..1
<b>Type</b>	INTEGER
<b>Default value</b>	0



<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SoAdTxSocketConnOrSocketConnBundleRef</b>	
<b>Description</b>	Choice reference to a SocketConnection or to a SocketConnectionGroup on which the PDU is to be sent on. The reference to a SocketConnectionGroup shall only be used for upper layers with IF API.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	CHOICE-REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SoAdTxRoutingGroupRef</b>	
<b>Description</b>	Reference to the routing group.	
<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>SoAdTxUdpTriggerMode</b>	
<b>Description</b>	<p>Specifies whether a PDU triggers the transmission of the nPduUdpTxBuffer. If this parameter is set to TRIGGER_NEVER, SoAd shall use an nPduUdpTxBuffer for the related socket connection.</p> <p>nPduUdpTxBuffer can only be used for upper layers with IF API, i.e. this parameter shall only be set to TRIGGER_NEVER if all upper layers belonging to the related socket connection have SoAdTxUpperLayerType set to "IF".</p> <p><b>This parameter will be enabled if SoAdTxUpperLayerType is IF and referenced SoAdSocketConnectionGroup has SoAdSocketnPduUdpTxBufferMin enabled.</b></p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	TRIGGER_ALWAYS	
<b>Range</b>	TRIGGER_ALWAYS TRIGGER_NEVER	



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SoAdTxUdpTriggerTimeout</b>	
<b>Description</b>	Specifies the time-out in [s] when the nPduUdpTxBuffer shall be transmitted at the latest after this PDU is put into the buffer. This optional parameter is only relevant if SoAdTxUdpTriggerMode is TRIGGER_NEVER.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.1	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.6.1.10. SoAdSocketConnectionGroup

Containers included		
Container name	Multiplicity	Description
<a href="#">SoAdSocketConnection</a>	1..n	<p>Specifies the socket connection (ID and remote address information).</p> <p>Note: Parameters that are common to all socket connections of a socket connection group are specified directly at the group.</p>
<a href="#">SoAdSocketProtocol</a>	1..1	Specifies the transport protocol and transport protocol specific parameters used for the socket connections of the socket connection group.

Parameters included	
Parameter name	Multiplicity
<a href="#">SoAdPduHeaderEnable</a>	1..1
<a href="#">SoAdResourceManagementEnable</a>	1..1
<a href="#">SoAdSocketLocalAddressRef</a>	1..1
<a href="#">SoAdSocketLocalPort</a>	1..1
<a href="#">SoAdSocketAutomaticSoConSetup</a>	1..1
<a href="#">SoAdSocketSoConModeChgNotification</a>	1..1

**Parameters included**

<a href="#">SoAdSocketIpAddrAssignmentChgNotification</a>	1..1
<a href="#">SoAdSocketTpRxBufferMin</a>	0..1
<a href="#">SoAdSocketMsgAcceptanceFilterEnabled</a>	1..1
<a href="#">SoAdSocketFramePriority</a>	0..1

Parameter Name	<b>SoAdPduHeaderEnable</b>	
<b>Description</b>	Enables the transmission of the PDU header (ID, length) on this socket connection.	
	<ul style="list-style-type: none"> <li>▶ true: Send PDU header before data.</li> <li>▶ false: Send data only.</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	

Parameter Name	<b>SoAdResourceManagementEnable</b>	
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Enables the resource management option for this socket. May not be activated for UDP sockets in receive and not for DoIP sockets.</p> <ul style="list-style-type: none"> <li>▶ true: resource management option enabled.</li> <li>▶ false: resource management option 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	

Parameter Name	<b>SoAdSocketLocalAddressRef</b>	
<b>Description</b>	Local IP address and interface used for this connection.	
<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>SoAdSocketLocalPort</b>	
<b>Description</b>	Local UDP or TCP port used for this connection. If this parameter is set to 0, SoAd requests Tcplp to select an ephemeral port.	
<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	

<b>Parameter Name</b>	<b>SoAdSocketAutomaticSoConSetup</b>	
<b>Description</b>	Specifies if the setup of the socket connection shall be done automatically or manually. <ul style="list-style-type: none"> <li>▶ true: Setup done automatically.</li> <li>▶ false: Setup done manually via SoAd_OpenSoCon() and SoAd_CloseSoCon().</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>SoAdSocketSoConModeChgNotification</b>	
<b>Description</b>	Specifies if the SoCon mode change notification callback function of the upper layer shall be called in case of SoCon mode change.	
<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>SoAdSocketIpAddrAssignmentChgNotification</b>	
-----------------------	--	--



<b>Description</b>	Specifies if the local IP address assignment change notification callback function of the upper layer shall be called if the assignment of the local IP address used by this socket connection changes.	
<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>SoAdSocketTpRxBufferMin</b>	
<b>Description</b>	Specifies the amount of data in bytes (PDU data for the upper layer and PDU header if used) the SoAd shall be able to buffer for data reception via this socket connection and using an upper layer with TP.  Note: In case of a TCP socket where PduHeaderMode is used and an upper layer with IF-API, the required buffer size can be determined automatically.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SoAdSocketMsgAcceptanceFilterEnabled</b>	
<b>Description</b>	Specifies if the message acceptance filter is enabled (TRUE) or not (FALSE).  Note: If a wildcard is used in SoAdSocketRemoteAddress AND SoAdSocketUdpListenOnly is FALSE, this parameter must be enabled.  ▶ true: The message acceptance filter is enabled. ▶ false: The message acceptance filter is disabled.	
<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>SoAdSocketFramePriority</b>	



<b>Description</b>	Specifies the priority of the Ethernet frames.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.6.1.11. SoAdSocketConnection

Containers included		
Container name	Multiplicity	Description
<a href="#">SoAdSocketRemoteAddress</a>	0..1	Container to specify the remote address (IP address and port) for a socket connection. If SoAdSocketRemoteAddress is not specified, the remote address must be set by the upper layer via SoAd_SetRemoteAddr().

Parameters included	
Parameter name	Multiplicity
<a href="#">SoAdSocketId</a>	1..1
<a href="#">SoAdTlsConnectionRef</a>	0..1

<b>Parameter Name</b>	<a href="#">SoAdSocketId</a>	
<b>Description</b>	Socket connection identifier used as SoConId in the interaction with upper layers.	
<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	

<b>Parameter Name</b>	<a href="#">SoAdTlsConnectionRef</a>	
<b>Description</b>	Reference to the TLS connection.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild



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

### 5.6.1.12. SoAdSocketRemoteAddress

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

<b>Parameter Name</b>	<b>SoAdSocketRemotelpAddress</b>
<b>Description</b>	Specifies the IP address of the remote node. The configured address must be of the same TcplpDomainType (i.e. IPv4 or IPv6) as the TcplpLocalAddr referenced in SoAdSocketLocalAddressRef. To accept any remote IP address, set SoAdSocketRemotelpAddress to "ANY". See message acceptance policy for more details.
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Default value</b>	ANY
<b>Configuration class</b>	<a href="#">VariantPostBuild:</a> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>SoAdSocketRemotePort</b>
<b>Description</b>	Specifies the remote UDP or TCP port used for this connection. To accept any remote port, set SoAdSocketRemotePort to 0. See message acceptance policy for more details.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Configuration class</b>	<a href="#">VariantPostBuild:</a> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.6.1.13. SoAdSocketProtocol

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>



#### Containers included

<a href="#">SoAdSocketUdp</a>	1..1	Specifies that UDP is used as transport protocol for the socket connection group and parameters only related to UDP socket connections.
<a href="#">SoAdSocketTcp</a>	1..1	Specifies that TCP is used as transport protocol for the socket connection group and parameters only related to TCP socket connections

#### 5.6.1.14. SoAdSocketUdp

##### Parameters included

Parameter name	Multiplicity
<a href="#">SoAdSocketUdpAliveSupervisionTimeout</a>	0..1
<a href="#">SoAdSocketUdpListenOnly</a>	1..1
<a href="#">SoAdSocketUdpRetryEnabled</a>	1..1
<a href="#">SoAdSocketUdpStrictHeaderLenCheckEnabled</a>	1..1
<a href="#">SoAdSocketUdpTriggerTimeout</a>	0..1
<a href="#">SoAdSocketnPduUdpTxBufferSize</a>	0..1
<a href="#">SoAdTxBufferSize</a>	1..1
<a href="#">SoAdEnableShadowBufferSize</a>	1..1

Parameter Name	<a href="#">SoAdSocketUdpAliveSupervisionTimeout</a>	
Description	Specifies the time in [s] a UDP socket connection remains in the mode SOAD_SOCON_ONLINE after the latest reception of a frame from the remote peer specified by the remote address. If this optional parameter is not enabled, UDP Alive Supervision is deactivated for the related socket connection group.	
Multiplicity	0..1	
Type	FLOAT	
Configuration class	<a href="#">PostBuild:</a>	<a href="#">VariantPostBuild</a>
Origin	AUTOSAR_ECUC	

Parameter Name	<a href="#">SoAdSocketUdpListenOnly</a>
Description	Specifies if the socket connection group is used only for reception or used for both reception and transmission.  ▶ true: This UDP port cannot transmit data.



	▶ false: This UDP port can send and receive data.
<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>SoAdSocketUdpRetryEnabled</b>
<b>Description</b>	Specifies if a UdpTxBuffer shall be used to retry a UDP transmission in case of TCPIP_E_PHYS_ADDR_MISS or not.  ▶ true: A UdpTxBuffer is used to retry a UDP transmission. ▶ false: A UDP transmission is not retried, if not successful.
<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>SoAdSocketUdpStrictHeaderLenCheckEnabled</b>
<b>Description</b>	Specifies if UDP messages shall be dropped (TRUE) if the length of all contained PDUs does not match the length of the whole message or not (FALSE). Shall only be set to TRUE if SoAdPduHeaderEnable is also set to TRUE.
<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>SoAdSocketUdpTriggerTimeout</b>
<b>Description</b>	Specifies the time-out in [s] a nPduUdpTxBuffer waits for a PDU with Trigger-Mode = TRIGGER_ALWAYS. That means when the time-out expires, the nPduUdpTxBuffer is transmitted. The timer is reset after each UDP transmission. This optional parameter is only relevant if a nPduUdpTxBuffer is used.
<b>Multiplicity</b>	0..1
<b>Type</b>	FLOAT



<b>Default value</b>	0.1	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SoAdSocketnPduUdpTxBufferMin</b>	
<b>Description</b>	<p>Specifies the amount of data in bytes (PDU data provided by the upper layer and PDU header if used) the SoAd shall be able to buffer for data transmission via this socket connection if the UDP message shall be buffered for transmission of multiple PDUs per UDP.</p> <p>Note: If a UDP socket and an upper layer with TP API or an upper layer with IF API with UDP transmit retry (for single PDUs) are configured, the required buffer size can be determined automatically. This optional parameter is only relevant if a nPduUdpTxBuffer is used.</p>	
<b>Multiplicity</b>	0..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SoAdTxBufferSize</b>	
<b>Description</b>	<p>This parameter defines the number of buffers which will be shared between the socket connections of this group. Each socket connection can take maximum one buffer out of the pool. Therefore, the maximum of this parameter is limited to the number of socket connections. If no buffer is available SoAd will try to send the PDU immediately.</p> <p>This parameter will be enabled if SoAdSocketnPduUdpTxBufferMin is enabled.</p> <p>Setting this parameter to 0 will disable the buffer pool and create a buffer for each socket connection.</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>SoAdEnableShadowBufferSize</b>	
-----------------------	-----------------------------------	--



<b>Description</b>	This parameter defines the size of the shadow buffer in bytes. A shadow buffer is required to avoid PDU drops and temporally store interrupting transmit request during transmission of buffered PDUs.  This parameter will be enabled if SoAdSocketnPduUdpTxBufferMin or SoAdSocketUdpRetryEnabled are enabled.  Setting this parameter to 0 will disable the shadow buffer.  The default value is 200 bytes to temporally store a few small PDUs. Depending on the size of transmitted PDUs the value needs to be adjusted on project needs.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	200
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.6.1.15. SoAdSocketTcp

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">SoAdSocketTcpImmediateTpTxConfirmation</a>	1..1
<a href="#">SoAdSocketTcpInitiate</a>	1..1
<a href="#">SoAdSocketTcpKeepAlive</a>	1..1
<a href="#">SoAdSocketTcpKeepAliveInterval</a>	0..1
<a href="#">SoAdSocketTcpKeepAliveProbesMax</a>	0..1
<a href="#">SoAdSocketTcpKeepAliveTime</a>	0..1
<a href="#">SoAdSocketTcpNoDelay</a>	0..1
<a href="#">SoAdSocketTcpTxQuota</a>	0..1

<b>Parameter Name</b>	<b>SoAdSocketTcpImmediateTpTxConfirmation</b>
<b>Description</b>	If set to FALSE, SoAd notifies the TP upper layer via transmit confirmation after a Tcp Ack was received. If set to TRUE, SoAd notifies the TP upper layer via transmit confirmation immediately after transmit was accepted by Tcplp.
<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>SoAdSocketTcpInitiate</b>
<b>Description</b>	Specifies the initiator for this TCP connection. <ul style="list-style-type: none"> <li>▶ true: This TCP connection is initiated by this module.</li> <li>▶ false: This TCP connection is to be initiated in the listen mode.</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>SoAdSocketTcpKeepAlive</b>
<b>Description</b>	Specifies to use the keep-alive mechanism for this connection. It will not be defined for UDP sockets.
<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>SoAdSocketTcpKeepAliveInterval</b>
<b>Description</b>	Specifies the interval in seconds between subsequent keep-alive probes.
<b>Multiplicity</b>	0..1
<b>Type</b>	FLOAT
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>SoAdSocketTcpKeepAliveProbesMax</b>
<b>Description</b>	Maximum number of times that TCP retransmits an individual data segment before aborting the connection.
<b>Multiplicity</b>	0..1



Type	INTEGER
Configuration class	<b>PostBuild:</b> VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	<b>SoAdSocketTcpKeepAliveTime</b>
Description	Specifies the time in seconds between the last data packet sent and the first keep-alive probe.
Multiplicity	0..1
Type	FLOAT
Default value	7200.0
Configuration class	<b>PostBuild:</b> VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	<b>SoAdSocketTcpNoDelay</b>
Description	<p>Specifies not to use the congestion control mechanism for this connection. It will not be defined for UDP sockets.</p> <ul style="list-style-type: none"> <li>▶ true: This TCP connection will NOT use congestion control.</li> <li>▶ false: This TCP connection will use congestion control.</li> </ul> <p>If the optional parameter is not enabled, the default behavior configured for Tcplp via the parameter TcplpTcpNagleEnabled is applied.</p> <p>Note: This parameter must not be set to FALSE if TcplpTcpNagleEnabled is set to FALSE.</p>
Multiplicity	0..1
Type	BOOLEAN
Default value	true
Configuration class	<b>PostBuild:</b> VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	<b>SoAdSocketTxQuota</b>
Description	<p>Specifies the maximum amount of bytes (PDU data provided by the upper layer and PDU header if used) the SoAd may queue for transmission via TCP at the Tcplp module for each socket connection of this socket connection group.</p> <p>Rationale: prohibits that a socket connection consumes all available transmit buffers at the Tcplp and blocks transmissions via other socket connections.</p>



	If the optional parameter is not enabled, the amount of data is not limited.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.6.1.16. SoAdSocketRoute

Containers included		
Container name	Multiplicity	Description
<a href="#">SoAdSocketRouteDest</a>	1..n	Describes the upper layer destination PDU for a message received on a TcpIp socket.

Parameters included	
Parameter name	Multiplicity
<a href="#">SoAdRxPduHeaderId</a>	0..1
<a href="#">SoAdRxSocketConnOrSocketConnBundleRef</a>	1..1

<b>Parameter Name</b>	<b>SoAdRxPduHeaderId</b>	
<b>Description</b>	ID contained in the packet received on the TCP/IP connection if the PDU header option is enabled.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SoAdRxSocketConnOrSocketConnBundleRef</b>
<b>Description</b>	Choice reference to a SocketConnection or to a SocketConnectionGroup on which the PDU was received. The reference to a SocketConnectionGroup shall only be used for upper layers with IF API or for both IF and TP if the meta data item SOCKET_CONNECTION_ID_16 exists for the referenced PDU.
<b>Multiplicity</b>	1..1
<b>Type</b>	CHOICE-REFERENCE



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.6.1.17. SoAdSocketRouteDest

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">SoAdRxPduRef</a>	1..1
<a href="#">SoAdRxPduld</a>	0..1
<a href="#">SoAdRxUpperLayerType</a>	1..1
<a href="#">SoAdRxRoutingGroupRef</a>	0..n

<b>Parameter Name</b>	<b>SoAdRxPduRef</b>
<b>Description</b>	Reference to the global PDU structure.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>SoAdRxPduld</b>
<b>Description</b>	This unique identifier is used for a receive cancellation request from an upper layer of the SoAd.
<b>Multiplicity</b>	0..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Configuration class</b>	<b>PostBuild:</b>
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>SoAdRxUpperLayerType</b>
<b>Description</b>	Specifies the upper layer interface type (must be "IF" in case of multiple RxPdus).
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	IF



<b>Range</b>	IF TP
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>SoAdRxRoutingGroupRef</b>
<b>Description</b>	Reference to the routing group.
<b>Multiplicity</b>	0..n
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.6.1.18. SoAdRoutingGroup

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

<b>Parameter Name</b>	<b>SoAdRoutingGroupId</b>
<b>Description</b>	Unique ID of the Routing Group.
<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

<b>Parameter Name</b>	<b>SoAdRoutingGroupsEnabledAtInit</b>
<b>Description</b>	If set to true, this routing group is enabled after initializing the SoAd module (i.e. enabled in the SoAd_Init 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>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>SoAdRoutingGroupTxTriggerable</b>
<b>Description</b>	Specifies the IF-TxPDUs related to the PduRouteDest containers.
<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

## 5.6.2. Recommended configurations

### 5.6.2.1. SoAdRecConfigurationDflt

<b>Containers included</b>	
<b>Container name</b>	<b>Container definition</b>
PduR	<a href="#">SoAdBswModules</a>
Sd	<a href="#">SoAdBswModules</a>
DolP	<a href="#">SoAdBswModules</a>

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Value</b>

#### 5.6.2.1.1. PduR

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Value</b>
<a href="#">SoAdIf</a>	true



#### Parameters included

<a href="#">SoAdTp</a>	true
<a href="#">SoAdIfTriggerTransmit</a>	false
<a href="#">SoAdIfTxConfirmation</a>	true
<a href="#">SoAdSoConModeChg</a>	false
<a href="#">SoAdLocallpAddrAssigmentChg</a>	false
<a href="#">SoAdUseCallerInfix</a>	true
<a href="#">SoAdUseTypeInfix</a>	true

#### 5.6.2.1.2. Sd

#### Parameters included

Parameter name	Value
<a href="#">SoAdIf</a>	true
<a href="#">SoAdTp</a>	false
<a href="#">SoAdIfTriggerTransmit</a>	false
<a href="#">SoAdIfTxConfirmation</a>	false
<a href="#">SoAdSoConModeChg</a>	false
<a href="#">SoAdLocallpAddrAssigmentChg</a>	true
<a href="#">SoAdUseCallerInfix</a>	true
<a href="#">SoAdUseTypeInfix</a>	true

#### 5.6.2.1.3. DoIP

#### Parameters included

Parameter name	Value
<a href="#">SoAdIf</a>	true
<a href="#">SoAdTp</a>	true
<a href="#">SoAdIfTriggerTransmit</a>	false
<a href="#">SoAdIfTxConfirmation</a>	true
<a href="#">SoAdSoConModeChg</a>	false
<a href="#">SoAdLocallpAddrAssigmentChg</a>	true
<a href="#">SoAdUseCallerInfix</a>	true



#### Parameters included

<a href="#">SoAdUseTypeInfix</a>	true
----------------------------------	------

### 5.6.3. Application programming interface (API)

#### 5.6.3.1. Macro constants

##### 5.6.3.1.1. SOAD\_CHANGEPARAMETER\_SVCID

Purpose	Defines the API ID of function <a href="#">SoAd_ChangeParameter()</a> .
Value	0xF1U

##### 5.6.3.1.2. SOAD\_CLOSESOCON\_SVCID

Purpose	Defines the API ID of function <a href="#">SoAd_CloseSoCon()</a> .
Value	0x09U

##### 5.6.3.1.3. SOAD\_COPYTXDATA\_SVCID

Purpose	Defines the API ID of function <a href="#">SoAd_CopyTxData()</a> .
Value	0x13U

##### 5.6.3.1.4. SOAD\_DISABLEROUTING\_SVCID

Purpose	Defines the API ID of function <a href="#">SoAd_DisableRouting()</a> .
Value	0x0FU

##### 5.6.3.1.5. SOAD\_DISABLESPECIFICROUTING\_SVCID

Purpose	Defines the API ID of function <a href="#">SoAd_DisableSpecificRouting()</a> .
---------	--



<b>Value</b>	0x21U
--------------	-------

#### 5.6.3.1.6. SOAD\_ENABLEROUTING\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_EnableRouting()</a> .
<b>Value</b>	0x0EU

#### 5.6.3.1.7. SOAD\_ENABLESPECIFICROUTING\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_EnableSpecificRouting()</a> .
<b>Value</b>	0x20U

#### 5.6.3.1.8. SOAD\_E\_INV\_ARG

<b>Purpose</b>	API requests called with invalid argument.
<b>Value</b>	0x03U

#### 5.6.3.1.9. SOAD\_E\_INV\_LENGTH\_IN\_PDUHEADER

<b>Purpose</b>	Error - Length in PduHeader exceeds the possible maximum.
<b>Value</b>	0xFFDU

#### 5.6.3.1.10. SOAD\_E\_INV\_METADATA

<b>Purpose</b>	Error - Invalid meta data.
<b>Value</b>	0x09U

#### 5.6.3.1.11. SOAD\_E\_INV\_PDUHEADER\_ID

<b>Purpose</b>	Error - Unknown PduHeader ID.
----------------	-------------------------------



<b>Value</b>	0x05U
--------------	-------

#### 5.6.3.1.12. SOAD\_E\_INV\_PDUID

<b>Purpose</b>	Error - Unknown PDU ID.
<b>Value</b>	0x06U

#### 5.6.3.1.13. SOAD\_E\_INV\_SOCKETID

<b>Purpose</b>	Error - Unknown socket address.
<b>Value</b>	0x07U

#### 5.6.3.1.14. SOAD\_E\_NOBUFS

<b>Purpose</b>	Error - No buffer space available.
<b>Value</b>	0x04U

#### 5.6.3.1.15. SOAD\_E\_NOTINIT

<b>Purpose</b>	API service called before initializing the module.
<b>Value</b>	0x01U

#### 5.6.3.1.16. SOAD\_E\_PARAM\_POINTER

<b>Purpose</b>	API requests called with NULL pointer.
<b>Value</b>	0x02U

#### 5.6.3.1.17. SOAD\_E\_TRIGGERTXBUF

<b>Purpose</b>	Error - Insufficient TriggerTransmitBuffer.
----------------	---



<b>Value</b>	0xFEU
--------------	-------

#### 5.6.3.1.18. SOAD\_E\_TWO\_PDUS\_SAME\_SOCON

<b>Purpose</b>	Error - Two PDUs cannot be configured for the same socket connection simultaneously.
<b>Value</b>	0xFB0U

#### 5.6.3.1.19. SOAD\_E\_TWO\_SOCON\_SAME\_PDU

<b>Purpose</b>	Error - Two socket connections cannot use the same PDU within the same Socon-group.
<b>Value</b>	0xFCU

#### 5.6.3.1.20. SOAD\_GETANDRESETMEASUREMENTDATA\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_GetAndResetMeasurementData()</a> .
<b>Value</b>	0x45U

#### 5.6.3.1.21. SOAD\_GETLOCALADDR\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_GetLocalAddr()</a> .
<b>Value</b>	0x0CU

#### 5.6.3.1.22. SOAD\_GETPHYSADDR\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_GetPhysAddr()</a> .
<b>Value</b>	0x0DU

#### 5.6.3.1.23. SOAD\_GETREMOTEADDR\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_GetRemoteAddr()</a> .
----------------	---



<b>Value</b>	0x1CU
--------------	-------

#### 5.6.3.1.24. SOAD\_GETSOCONID\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_GetSoConId()</a> .
<b>Value</b>	0x07U

#### 5.6.3.1.25. SOAD\_GETSOCONMODE\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_GetSoConMode()</a> .
<b>Value</b>	0xF0U

#### 5.6.3.1.26. SOAD\_GETVERSIONINFO\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_GetVersionInfo()</a> .
<b>Value</b>	0x02U

#### 5.6.3.1.27. SOAD\_IFROUTINGGROUPTRANSMIT\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_IfRoutingGroupTransmit()</a> .
<b>Value</b>	0x1DU

#### 5.6.3.1.28. SOAD\_IFSPECIFICROUTINGGROUPTRANSMIT\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_IfSpecificRoutingGroupTransmit()</a> .
<b>Value</b>	0x1FU

#### 5.6.3.1.29. SOAD\_IFTRANSMIT\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_IfTransmit()</a> .
----------------	--



<b>Value</b>	0x03U
--------------	-------

#### 5.6.3.1.30. SOAD\_INIT\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_Init()</a> .
<b>Value</b>	0x01U

#### 5.6.3.1.31. SOAD\_INSTANCE\_ID

<b>Purpose</b>	Defines the instance number of this Socket Adaptor. Because multiple instances of the Socket Adaptor are not supported, the instance ID is always zero.
<b>Value</b>	0U

#### 5.6.3.1.32. SOAD\_INTERNAL\_SVCID

<b>Purpose</b>	Defines the API ID of internal functions.
<b>Value</b>	0xFFU

#### 5.6.3.1.33. SOAD\_INVALID\_ULFUNCID

<b>Purpose</b>	Invalid upper layer function array index.
<b>Value</b>	0xFFU

#### 5.6.3.1.34. SOAD\_ISCONNECTIONREADY\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_IsConnectionReady()</a> .
<b>Value</b>	0xF3U

#### 5.6.3.1.35. SOAD\_LOCALIPADDRASSIGNMENTCHG\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_LocalIpAddrAssignmentChg()</a> .
----------------	--



<b>Value</b>	0x18U
--------------	-------

#### 5.6.3.1.36. SOAD\_MAINFUNCTIONTX\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_MainFunction()</a> .
<b>Value</b>	0xF2U

#### 5.6.3.1.37. SOAD\_MAINFUNCTION\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_MainFunction()</a> .
<b>Value</b>	0x19U

#### 5.6.3.1.38. SOAD\_OPENSOCN\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_OpenSoCon()</a> .
<b>Value</b>	0x08U

#### 5.6.3.1.39. SOAD\_READDHCPHOSTNAMEOPTION\_SVCID

<b>Purpose</b>	Define API id of function <a href="#">SoAd_ReadDhcpHostNameOption()</a> .
<b>Value</b>	0x1AU

#### 5.6.3.1.40. SOAD\_RELEASEIPADDRASSIGNMENT\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_ReleaselpAddrAssignment()</a> .
<b>Value</b>	0x0BU

#### 5.6.3.1.41. SOAD\_RELEASEREMOTEADDR\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_ReleaseRemoteAddr()</a> .
----------------	---



<b>Value</b>	0x23U
--------------	-------

#### 5.6.3.1.42. SOAD\_REQUESTIPADDRASSIGNMENT\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_RequestIpAddrAssignment()</a> .
<b>Value</b>	0x0AU

#### 5.6.3.1.43. SOAD\_RXINDICATION\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_TcpIpRxIndication()</a> .
<b>Value</b>	0x12U

#### 5.6.3.1.44. SOAD\_SETREMOTEADDR\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_SetRemoteAddr()</a> .
<b>Value</b>	0x10U

#### 5.6.3.1.45. SOAD\_SETUNIQUEREMOTEADDR\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_SetUniqueRemoteAddr()</a> .
<b>Value</b>	0x1EU

#### 5.6.3.1.46. SOAD\_TCPACCEPTED\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_TcpAccepted()</a> .
<b>Value</b>	0x15U

#### 5.6.3.1.47. SOAD\_TCPCONNECTED\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_TcpConnected()</a> .
----------------	--



<b>Value</b>	0x16U
--------------	-------

#### 5.6.3.1.48. SOAD\_TCPIPEVENT\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_TcplpEvent()</a> .
<b>Value</b>	0x17U

#### 5.6.3.1.49. SOAD\_TPCANCELRECEIVE\_SVCID

<b>Purpose</b>	Defines the API ID of function SoAd_CancelReceive().
<b>Value</b>	0x06U

#### 5.6.3.1.50. SOAD\_TPCANCELTRANSMIT\_SVCID

<b>Purpose</b>	Defines the API ID of function SoAd_CancelTransmit().
<b>Value</b>	0x05U

#### 5.6.3.1.51. SOAD\_TPCHANGEPARAMETER\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_TpChangeParameter()</a> .
<b>Value</b>	0x11U

#### 5.6.3.1.52. SOAD\_TPTRANSMIT\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_TpTransmit()</a> .
<b>Value</b>	0x04U

#### 5.6.3.1.53. SOAD\_TXCONFIRMATION\_SVCID

<b>Purpose</b>	Defines the API ID of function <a href="#">SoAd_TxConfirmation()</a> .
----------------	--



<b>Value</b>	0x14U
--------------	-------

#### 5.6.3.1.54. SOAD\_WROTEDHCPPHOSTNAMEOPTION\_SVCID

<b>Purpose</b>	Define API id of function <a href="#">SoAd_WriteDhcpHostNameOption()</a> .
<b>Value</b>	0x1BU

#### 5.6.3.1.55. SoAdIf\_Transmit

<b>Purpose</b>	
<b>Value</b>	SoAd_IfTransmit

#### 5.6.3.1.56. SoAdTp\_CancelReceive

<b>Purpose</b>	
<b>Value</b>	SoAd_TpCancelReceive

#### 5.6.3.1.57. SoAdTp\_CancelTransmit

<b>Purpose</b>	
<b>Value</b>	SoAd_TpCancelTransmit

#### 5.6.3.1.58. SoAdTp\_Transmit

<b>Purpose</b>	
<b>Value</b>	SoAd_TpTransmit

### 5.6.3.2. Functions

#### 5.6.3.2.1. SoAd\_ChangeParameter

<b>Purpose</b>	Change a parameter of the TCP/IP stack.
----------------	---



<b>Synopsis</b>	<code>Std_ReturnType SoAd_ChangeParameter ( SoAd_SoConIdType SoConId , uint8 ParameterId , uint8 * ParameterValuePtr );</code>	
<b>Service ID</b>	0x11	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SoConIds. Non reentrant for the same SoConId.	
<b>Parameters (in)</b>	<code>SoConId</code>	Socket connection index specifying the socket connection for which the parameter shall be changed
	<code>ParameterId</code>	Identifier of the parameter to be changed
	<code>ParameterValuePtr</code>	Pointer to the memory containing the new parameter value
<b>Return Value</b>	<code>Std_ReturnType</code>	
	<code>E_OK</code>	The request was successful.
	<code>E_NOT_OK</code>	The request was not successful.
<b>Description</b>	This API service requests the SoAd or TCP/IP stack to change a connection parameter. For example, the Nagle algorithm may be controlled by this API.	

#### 5.6.3.2.2. SoAd\_CloseSoCon

<b>Purpose</b>	Close a socket connection.	
<b>Synopsis</b>	<code>Std_ReturnType SoAd_CloseSoCon ( SoAd_SoConIdType SoConId , boolean Abort );</code>	
<b>Service ID</b>	0x09	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	<code>SoConId</code>	Socket connection index specifying the socket connection to be closed
	<code>Abort</code>	<ul style="list-style-type: none"> <li>▶ TRUE: Socket connection is immediately terminated.</li> <li>▶ FALSE: Socket connection is terminated if no other upper layer uses this socket connection.</li> </ul>
<b>Return Value</b>	<code>Std_ReturnType</code>	
	<code>E_OK</code>	The request was successful.
	<code>E_NOT_OK</code>	The request was not successful.



<b>Description</b>	This service closes the socket connection specified by the SoConId.
--------------------	---

#### 5.6.3.2.3. SoAd\_CopyTxData

<b>Purpose</b>	Copy transmission data.	
<b>Synopsis</b>	<code>BufReq_ReturnType SoAd_CopyTxData ( TcpIp_SocketIdType SocketId, , uint8 * BufPtr , uint16 BufLength );</code>	
<b>Service ID</b>	0x13	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non reentrant	
<b>Parameters (in)</b>	SocketId	Socket handle identifying the local socket resource
	BufPtr	Pointer to buffer for transmission data
	BufLength	Length of provided data buffer
<b>Return Value</b>		
<b>Description</b>	This service requests to copy data for transmission to the buffer indicated. This call is triggered by Tcplp_Transmit(). Note: The call to <a href="#">SoAd_CopyTxData()</a> may happen in the context of Tcplp_Transmit().	

#### 5.6.3.2.4. SoAd\_DisableRouting

<b>Purpose</b>	Disable routing of a group of PDUs.	
<b>Synopsis</b>	<code>Std_ReturnType SoAd_DisableRouting ( SoAd_RoutingGroupIdType Id ) ;</code>	
<b>Service ID</b>	0x0F	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	Id	Identification of the routing group
<b>Return Value</b>	<code>Std_ReturnType</code>	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
<b>Description</b>	Disables the routing of a group of PDUs in the SoAd related to the RoutingGroup specified by the parameter Id. Routing of PDUs can be either forwarding of PDUs	



from the upper layer to a TCP or UDP socket of the TCP/IP stack specified by a PduRoute or the other way around specified by a SocketRoute.

#### 5.6.3.2.5. SoAd\_DisableSpecificRouting

<b>Purpose</b>	Disable routing of a group of PDUs on a specific socket connection.	
<b>Synopsis</b>	<code>Std_ReturnType SoAd_DisableSpecificRouting ( SoAd_RoutingGroupIdType Id , SoAd_SoConIdType SoConId );</code>	
<b>Service ID</b>	0x21	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	<code>Id</code>	Identification of the routing group
	<code>SoConId</code>	Identification of the socket connection
<b>Return Value</b>	<code>Std_ReturnType</code>	
	<code>E_OK</code>	The request was successful.
	<code>E_NOT_OK</code>	The request was not successful.
<b>Description</b>	Disables the routing of a group of PDUs in the SoAd related to the RoutingGroup specified by the parameter Id on a specific socket connection specified by the SoConId. Routing of PDUs can be either forwarding of PDUs from the upper layer to a TCP or UDP socket of the TCP/IP stack specified by a PduRoute or the other way around specified by a SocketRoute.	

#### 5.6.3.2.6. SoAd\_EnableRouting

<b>Purpose</b>	Enable routing of a group of PDUs.	
<b>Synopsis</b>	<code>Std_ReturnType SoAd_EnableRouting ( SoAd_RoutingGroupIdType Id );</code>	
<b>Service ID</b>	0x0E	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	<code>Id</code>	Identification of the routing group
<b>Return Value</b>	<code>Std_ReturnType</code>	
	<code>E_OK</code>	The request was successful.
	<code>E_NOT_OK</code>	The request was not successful.



<b>Description</b>	Enables routing of a group of PDUs in the SoAd related to the RoutingGroup specified by the parameter Id. Routing of PDUs can be either forwarding of PDUs from the upper layer to a TCP or UDP socket of the TCP/IP stack specified by a PduRoute or the other way around specified by a SocketRoute.
--------------------	--

#### 5.6.3.2.7. SoAd\_EnableSpecificRouting

<b>Purpose</b>	Enable routing of a group of PDUs on a specific socket connection.	
<b>Synopsis</b>	<code>Std_ReturnType SoAd_EnableSpecificRouting ( SoAd_RoutingGroupIdxType Id , SoAd_SoConIdType SoConId );</code>	
<b>Service ID</b>	0x20	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	<code>Id</code>	Identification of the routing group
	<code>SoConId</code>	Identification of the socket connection
<b>Return Value</b>	<code>Std_ReturnType</code>	
	<code>E_OK</code>	The request was successful.
	<code>E_NOT_OK</code>	The request was not successful.
<b>Description</b>	Enables the routing of a group of PDUs in the SoAd related to the RoutingGroup specified by the parameter Id on a specific socket connection specified by the SoConId. Routing of PDUs can be either forwarding of PDUs from the upper layer to a TCP or UDP socket of the TCP/IP stack specified by a PduRoute or the other way around specified by a SocketRoute.	

#### 5.6.3.2.8. SoAd\_GetAndResetMeasurementData

<b>Purpose</b>	Reads and resets measurement data.	
<b>Synopsis</b>	<code>Std_ReturnType SoAd_GetAndResetMeasurementData ( SoAd_MeasurementIdxType MeasurementIdx , boolean MeasurementResetNeeded , uint32 * MeasurementDataPtr );</code>	
<b>Parameters (in)</b>	<code>MeasurementIdx</code>	Index to select specific measurement data: ▶ SOAD_MEAS_DROP_TCP (0x01) - Measurement index of dropped PDUs



		caused by invalid destination TCP port.
		<ul style="list-style-type: none"> <li>▶ SOAD_MEAS_DROP_UDP (0x02) - Measurement index of dropped PDUs caused by invalid destination UDP port.</li> <li>▶ SOAD_MEAS_RESERVED_1 (0x03-0x7F) - Reserved by AU-TOSAR.</li> <li>▶ SOAD_MEAS_RESERVED_2 (0x80-0xEF) - Vendor-specific range.</li> <li>▶ SOAD_MEAS_RESERVED_3 (0xF0-0xFE) - Reserved by AU-TOSAR (future use).</li> <li>▶ SOAD_MEAS_ALL (0xFF) - Represents all measurement indexes.</li> </ul>
	MeasurementResetNeeded	Flag to trigger a reset of the measurement data
<b>Parameters (out)</b>	MeasurementDataPtr	Pointer to the data buffer where to copy measurement data
<b>Return Value</b>	Std_ReturnType	
	E_OK	The function was successfully executed.
	E_NOT_OK	The function was not successfully executed.
<b>Description</b>	This service allows to read and reset detailed measurement data for diagnostic purposes. Getting all MeasurementIdx's at once is not supported. SOAD_MEAS_ALL shall only be used to reset all MeasurementIdx's at once. A NULL_PTR shall be provided for MeasurementDataPtr in this case.	

### 5.6.3.2.9. SoAd\_GetLocalAddr

<b>Purpose</b>	Get the current local IP address.
<b>Synopsis</b>	Std_ReturnType <b>SoAd_GetLocalAddr</b> ( SoAd_SoConIdType SoConId , TcpIp_SockAddrType * LocalAddrPtr , uint8 * NetmaskPtr , TcpIp_SockAddrType * DefaultRouterPtr );
<b>Service ID</b>	0x0C



<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SoConId	Socket connection index representing the SoAd socket connection for which the actual local IP address shall be obtained
<b>Parameters (out)</b>	LocalAddrPtr	Pointer to the struct where the local address (IP address and port) is stored
	NetmaskPtr	Pointer to the memory where the network mask of the IPv4 address or address prefix of the IPv6 address in CIDR notation is stored
	DefaultRouterPtr	Pointer to the struct where the IP address of the default router (gateway) is stored (struct member port is not used and of arbitrary value)
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
<b>Description</b>	Obtains the local address (IP address and port) actually used for the SoAd socket connection specified by the SoConId, the netmask, and default router.	

#### 5.6.3.2.10. SoAd\_GetPhysAddr

<b>Purpose</b>	Get the physical IP address.	
<b>Synopsis</b>	Std_ReturnType <b>SoAd_GetPhysAddr</b> ( SoAd_SoConIdType SoConId , uint8 * PhysAddrPtr );	
<b>Service ID</b>	0x0D	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SoConId	Socket connection index representing the SoAd socket connection for which the physical source address of the related EthIf controller shall be obtained
<b>Parameters (out)</b>	PhysAddrPtr	Pointer to the memory where the physical source address (MAC address) in network byte order is stored



<b>Return Value</b>	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
<b>Description</b>	Obtains the physical source address of the Ethlf controller used by the SoAd socket connection specified by the SoConId.	

#### 5.6.3.2.11. SoAd\_GetRemoteAddr

<b>Purpose</b>	Get the remote address of a socket connection.	
<b>Synopsis</b>	Std_ReturnType <b>SoAd_GetRemoteAddr</b> ( SoAd_SoConIdType SoConId , TcpIp_SockAddrType * IpAddrPtr );	
<b>Service ID</b>	0x1C	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SoConId	Socket connection index representing the SoAd socket connection from which the remote address shall be obtained
<b>Parameters (out)</b>	IpAddrPtr	Pointer to the memory where the remote address is stored
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
<b>Description</b>	Obtains the remote address (IP address and port) of the specified socket connection. If no remote address is set for the socket connection, E_NOT_OK is returned.	

#### 5.6.3.2.12. SoAd\_GetSoConId

<b>Purpose</b>	Get socket connection ID for given PDU ID.	
<b>Synopsis</b>	Std_ReturnType <b>SoAd_GetSoConId</b> ( PduIdType TxPduId , SoAd_SoConIdType * SoConIdPtr );	
<b>Service ID</b>	0x07	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	



<b>Parameters (in)</b>	TxPduId	Transmit the PDU ID specifying the SoAd socket connection for which the socket connection index shall be returned.
<b>Parameters (out)</b>	SoConIdPtr	Pointer to the memory receiving the socket connection index
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
<b>Description</b>	Returns the socket connection index related to the specified transmit PDU ID. In case a fan-out is configured for TxPduId (i.e. multiple SoAdPduRouteDest specified), E_NOT_OK shall be returned.	

#### 5.6.3.2.13. SoAd\_GetSoConMode

<b>Purpose</b>	Get socket connection mode for the given socket connection index.	
<b>Synopsis</b>	Std_ReturnType <b>SoAd_GetSoConMode</b> ( SoAd_SoConIdType SoConId , SoAd_SoConModeType * ModePtr );	
<b>Service ID</b>	0xF0	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SoConId	Index of the socket connection for which the mode shall be returned
<b>Parameters (out)</b>	ModePtr	Pointer to the memory where the socket connection mode shall be stored
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
<b>Description</b>	Returns the socket connection mode related to the specified socket connection.	

#### 5.6.3.2.14. SoAd\_GetVersionInfo

<b>Purpose</b>	Get version information of the SoAd module.
<b>Synopsis</b>	void <b>SoAd_GetVersionInfo</b> ( Std_VersionInfoType * Versioninfo );



<b>Service ID</b>	0x02	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non reentrant	
<b>Parameters (out)</b>	Versioninfo	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>	

#### **5.6.3.2.15. SoAd\_IfRoutingGroupTransmit**

<b>Purpose</b>	Initiate transmission of IF-TxPDUs belonging to a given routing group.					
<b>Synopsis</b>	<pre>Std_ReturnType SoAd_IfRoutingGroupTransmit ( SoAd_Routing- GroupIDType Id );</pre>					
<b>Service ID</b>	0x1D					
<b>Sync/Async</b>	Asynchronous					
<b>Reentrancy</b>	Reentrant					
<b>Parameters (in)</b>	Id	Identification of the routing group				
<b>Return Value</b>	<p>Std_ReturnType</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">E_OK</td> <td style="padding: 5px;">The request was successful.</td> </tr> <tr> <td style="padding: 5px;">E_NOT_OK</td> <td style="padding: 5px;">The request was not successful.</td> </tr> </table>		E_OK	The request was successful.	E_NOT_OK	The request was not successful.
E_OK	The request was successful.					
E_NOT_OK	The request was not successful.					
<b>Description</b>	Triggers the transmission of all IF-TxPDUs identified by the parameter Id after requesting the data from the related upper layer.					

#### **5.6.3.2.16. SoAd\_IfSpecificRoutingGroupTransmit**

<b>Purpose</b>	Initiate transmission of IF-TxPDUs belonging to a given routing group on a specific socket connection.
<b>Synopsis</b>	<code>Std_ReturnType SoAd_IfSpecificRoutingGroupTransmit ( SoAd_RoutingGroupIdType Id , SoAd_SoConIdType SoConId );</code>



<b>Service ID</b>	0x1F	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	Id	Identification of the routing group
	SoConId	Identification of the socket connection
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
<b>Description</b>	Triggers the transmission of all IF-TxPDUs identified by the parameter Id on the socket connection specified by SoConId after requesting the data from the related upper layer.	

#### 5.6.3.2.17. SoAd\_IfTransmit

<b>Purpose</b>	Transfer L-PDU.	
<b>Synopsis</b>	Std_ReturnType <b>SoAd_IfTransmit</b> ( PduIdType SoAdSrcPduId , const PduInfoType * SoAdSrcPduInfoPtr );	
<b>Service ID</b>	0x03	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SoAdSrcPduId	This parameter contains a unique identifier referencing to the PDU Routing Table and thereby specifying the socket to be used for transmission of the data.
	SoAdSrcPduInfoPtr	A pointer to a structure with socket-related data: data length and pointer to a data buffer
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request was accepted.
	E_NOT_OK	The request was not accepted (e.g. due to a still ongoing transmission in the corresponding socket or the message to be transmitted is too long).
<b>Description</b>	This service is used to request the transfer of L-PDU.	



### 5.6.3.2.18. SoAd\_Init

<b>Purpose</b>	Initialize the SoAd module.	
<b>Synopsis</b>	<code>void SoAd_Init ( const SoAd_ConfigType * SoAdConfigPtr );</code>	
<b>Service ID</b>	0x01	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non reentrant	
<b>Parameters (in)</b>	SoAdConfigPtr	Points to the implementation-specific structure
<b>Description</b>	This function initializes the SoAd module.	

### 5.6.3.2.19. SoAd\_IsConnectionReady

<b>Purpose</b>	Checks if ARP entry and IpSec SA exist in Tcplp.	
<b>Synopsis</b>	<code>TcpIp_ReturnType SoAd_IsConnectionReady ( SoAd_SoConIdType SoConId , const TcpIp_SockAddrType * RemoteAddrPtr );</code>	
<b>Service ID</b>	0x1F	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SoConId	Identification of the socket connection
<b>Return Value</b>	Tcplp_ReturnType TCPIP_E_PENDING      ARP entry or IpSec SA does not exist TCPIP_E_NOT_OK      Transmission not allowed according to policy table TCPIP_OK      ARP and IpSec SA exist, data can be sent	
<b>Description</b>	Calls Tcplp with the correct SocketId and RemoteAddr of the corresponding SoConId. Passes the return value of Tcplp to the upper layer.	

### 5.6.3.2.20. SoAd\_IsValidConfig

<b>Purpose</b>	Checks the compatibility of the post-build configuration.
<b>Synopsis</b>	<code>Std_ReturnType SoAd_IsValidConfig ( const void * SoAdConfigPtr );</code>



<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SoAdConfigPtr	Pointer to the configuration data of the SoAd module.
<b>Return Value</b>		
<b>Description</b>	This service checks the compatibility of the post-build configuration against the source code.	

#### 5.6.3.2.21. SoAd\_LocallpAddrAssignmentChg

<b>Purpose</b>	Indicates an IP address change.	
<b>Synopsis</b>	<pre>void <b>SoAd_LocallpAddrAssignmentChg</b> ( TcpIp_LocalAddrIdType IpAddrId , TcpIp_IpAddrStateType State );</pre>	
<b>Service ID</b>	0x18	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non reentrant	
<b>Parameters (in)</b>	IpAddrId  State	IP address Identifier, representing an IP address specified in the Tcplp module configuration (e.g. static IPv4 address on EthIf controller 0).  State of IP address assignment: <ul style="list-style-type: none"> <li>▶ TCPIP_IPADDR_STATE_ASSIGNED</li> <li>▶ TCPIP_IPADDR_STATE_ONHOLD</li> <li>▶ TCPIP_IPADDR_STATE_UNASSIGNED</li> </ul>
<b>Description</b>	This service gets called by the TCP/IP stack if an IP address changes (i.e. a new address is assigned or an assigned address becomes invalid).	

#### 5.6.3.2.22. SoAd\_MainFunction

<b>Purpose</b>	Main function of the SoAd.
<b>Synopsis</b>	<pre>void <b>SoAd_MainFunction</b> ( void );</pre>
<b>Service ID</b>	0x19



<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non reentrant
<b>Description</b>	This function is the main function for scheduling SoAd.

### 5.6.3.2.23. SoAd\_MainFunctionTx

<b>Purpose</b>	Main function containing only the transmit part of the SoAd.
<b>Synopsis</b>	<code>void SoAd_MainFunctionTx ( void );</code>
<b>Service ID</b>	0xF2
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non reentrant
<b>Description</b>	Transmit part of the main function if enabled to be external

### 5.6.3.2.24. SoAd\_OpenSoCon

<b>Purpose</b>	Open a socket connection.	
<b>Synopsis</b>	<code>Std_ReturnType SoAd_OpenSoCon ( SoAd_SoConIdType SoConId );</code>	
<b>Service ID</b>	0x08	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SoConId	Socket connection index specifying the socket connection to be opened
<b>Return Value</b>	Std_ReturnType E_OK                      The request was successful. E_NOT_OK                The request was not successful.	
<b>Description</b>	This service opens the socket connection specified by the SoConId.	

### 5.6.3.2.25. SoAd\_ReadDhcpHostNameOption

<b>Purpose</b>	Read the DHCP host name.
----------------	--------------------------



<b>Synopsis</b>	<code>Std_ReturnType SoAd_ReadDhcpHostNameOption ( SoAd_SoConIdType SoConId , uint8 * Length , uint8 * Data );</code>	
<b>Service ID</b>	0x1A	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SoConIds. Non reentrant for the same SoConId.	
<b>Parameters (in)</b>	SoConId	Socket connection index specifying the socket connection for which the parameter shall be read
	Length	As input parameter, contains the length of the provided data buffer. Will be overwritten with the length of the actual data.
	Data	Pointer to the provided memory buffer the value is copied to.
<b>Return Value</b>	<code>Std_ReturnType</code>	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
<b>Description</b>	By this API service, an upper layer of the SoAd can read the currently configured host name option in the DHCP submodule of the TCP/IP stack.	

### 5.6.3.2.26. SoAd\_ReleaselpAddrAssignment

<b>Purpose</b>	Release IP address assignment.	
<b>Synopsis</b>	<code>Std_ReturnType SoAd_ReleaselpAddrAssignment ( SoAd_SoConIdType SoConId );</code>	
<b>Service ID</b>	0x0B	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SoConId	Socket connection index specifying the socket connection for which the IP address shall be released
<b>Return Value</b>	<code>Std_ReturnType</code>	
	E_OK	The request was accepted.
	E_NOT_OK	The request was not accepted.
<b>Description</b>	This API service releases the local IP address assignment used for the socket connection specified by the SoConId.	



#### 5.6.3.2.27. SoAd\_ReleaseRemoteAddr

<b>Purpose</b>	Release the remote address of a socket connection.	
<b>Synopsis</b>	<code>void SoAd_ReleaseRemoteAddr ( SoAd_SoConIdType SoConId );</code>	
<b>Service ID</b>	0x23	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SoConIds. Non reentrant for the same SoConId.	
<b>Parameters (in)</b>	SoConId	Index of the socket connection for which the remote address shall be released
<b>Description</b>	This API service releases the remote address (IP address and port) of the specified socket connection, i.e. it is set back to the configured remote address setting.	

#### 5.6.3.2.28. SoAd\_RequestIpAddrAssignment

<b>Purpose</b>	Request IP address assignment.	
<b>Synopsis</b>	<code>Std_ReturnType SoAd_RequestIpAddrAssignment ( SoAd_SoConIdType SoConId , TcpIp_IpAddrAssignmentType Type , TcpIp_SockAddrType * LocalIpAddrPtr );</code>	
<b>Service ID</b>	0x0A	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant for different SoConIds. Non reentrant for the same SoConId.	
<b>Parameters (in)</b>	SoConId	Socket connection index specifying the socket connection for which the IP address shall be set
	Type	Type of IP address assignment that shall be initiated
	LocalIpAddrPtr	Pointer to the structure containing the IP address that shall be assigned to the EthIf controller indirectly specified via the SoConId. Note: This parameter is only used if the parameter Type is set to TCPIP_IPADDR_ASSIGNMENT_STATIC.
<b>Return Value</b>	Std_ReturnType	



	E_OK	The request was accepted.
	E_NOT_OK	The request was not accepted.
<b>Description</b>	This API service initiates the local IP address assignment to be used for the socket connection specified by the SoConId.	

#### 5.6.3.2.29. SoAd\_RxIndication

<b>Purpose</b>	Data reception of UDP datagram or TCP stream.	
<b>Synopsis</b>	<pre>void <b>SoAd_RxIndication</b> ( TcpIp_SocketIdType SocketId , const                            TcpIp_SockAddrType * RemoteAddrPtr , uint8 * BufPtr , uint16                            Length );</pre>	
<b>Service ID</b>	0x12	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non reentrant	
<b>Parameters (in)</b>	SocketId	Socket handle identifying the local socket resource.
	RemoteAddrPtr	IP address and port of the remote host that sent the data
	BufPtr	Pointer to the received data
	Length	Data length of the received TCP segment or UDP datagram
<b>Description</b>	Indicates the reception of a new TCP segment or UDP datagram and provides all data via BufPtr.	

#### 5.6.3.2.30. SoAd\_SetRemoteAddr

<b>Purpose</b>	Set the remote address of a socket connection.
<b>Synopsis</b>	<pre>Std_ReturnType <b>SoAd_SetRemoteAddr</b> ( SoAd_SoConIdType SoConId ,  const TcpIp_SockAddrType * IpAddrPtr );</pre>
<b>Service ID</b>	0x10
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Reentrant for different SoConIds. Non reentrant for the same SoConId.



<b>Parameters (in)</b>	SoConId	Socket connection index specifying the socket connection for which the remote address shall be set
	IpAddrPtr	Struct containing the IP address and port to be set
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
<b>Description</b>	This API service sets the remote address (IP address and port) of the specified socket connection.	

#### 5.6.3.2.31. SoAd\_SetUniqueRemoteAddr

<b>Purpose</b>	Returns the index of a socket connection where the given remote address is set.	
<b>Synopsis</b>	Std_ReturnType <b>SoAd_SetUniqueRemoteAddr</b> ( SoAd_SoConIdType SoConId , const TcpIp_SockAddrType * RemoteAddrPtr , SoAd_SoConIdType * AssignedSoConIdPtr );	
<b>Service ID</b>	0x1e	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SoConGroups. Non reentrant for the same SoConGroup.	
<b>Parameters (in)</b>	SoConId	Index of any socket connection that is part of the SoAdSocketConnectionGroup
	RemoteAddrPtr	Pointer to the structure containing the requested remote IP address and port
<b>Parameters (out)</b>	AssignedSoConIdPtr	Pointer to the SoAd_SoConIdType where the index of the socket connection configured with the remote address (IpAddrPtr) shall be stored
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request was accepted.
	E_NOT_OK	The request was rejected, AssignedSoConIdPtr remains unchanged.
<b>Description</b>	This API service shall either return the socket connection index of the SoAdSocketConnectionGroup where the specified remote address (IP address and port) is set or assign the remote address to an unused socket connection from the same SoAdSocketConnectionGroup.	



### 5.6.3.2.32. SoAd\_TcpAccepted

<b>Purpose</b>	Indicates an incoming TCP connection on a server socket.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>SoAd_TcpAccepted</b> ( TcpIp_SocketIdType SocketId , TcpIp_SocketIdType SocketIdConnected , const TcpIp_SockAddrType * RemoteAddrPtr );</pre>	
<b>Service ID</b>	0x15	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non reentrant	
<b>Parameters (in)</b>	SocketId	Socket handle identifying the local socket resource
	SocketIdConnected	Socket handle for the established connection
	RemoteAddrPtr	IP address and port of the remote host
<b>Return Value</b>	Result of operation	
	E_OK	SoAd accepts the established connection.
	E_NOT_OK	SoAd refuses the established connection, Tcplp stack shall close the connection.
<b>Description</b>	<p><a href="#">SoAd_TcpAccepted()</a> gets called if the stack put a socket into the listen mode before (as server) and a peer connected to it (as client). In detail: The TCP/IP stack calls this function after a socket was set into the listen state with Tcplp_TcpListen() and a TCP connection is requested by the peer.</p>	

### 5.6.3.2.33. SoAd\_TcpConnected

<b>Purpose</b>	Indicates a successful connection from client side.	
<b>Synopsis</b>	<pre>void <b>SoAd_TcpConnected</b> ( TcpIp_SocketIdType SocketId );</pre>	
<b>Service ID</b>	0x16	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non reentrant	
<b>Parameters (in)</b>	SocketId	Socket handle identifying the local socket resource



<b>Description</b>	<a href="#">SoAd_TcpConnected()</a> gets called if the stack initiated a TCP connection before (as client) and the peer (the server) acknowledged the connection set up. In detail: The TCP/IP stack calls this function after a socket was requested to connect with Tcplp-TcpConnect() and a TCP connection is confirmed by the peer.
--------------------	---

#### 5.6.3.2.34. SoAd\_TcplpEvent

<b>Purpose</b>	Indicates an event in the TCP/IP stack.	
<b>Synopsis</b>	<code>void SoAd_TcpIpEvent ( TcpIp_SocketIdType SocketId , TcpIp_EventType Event );</code>	
<b>Service ID</b>	0x17	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non reentrant	
<b>Parameters (in)</b>	SocketId	Socket handle identifying the local socket resource
	Event	This parameter contains a description of the event just encountered.
<b>Description</b>	The service gets called if the stack encounters a condition described by the values in TcplpEvent.	

#### 5.6.3.2.35. SoAd\_TcpTlsAccepted

<b>Purpose</b>	Indicates an incoming TCP connection on a server socket that has SoAdTlsConnectionRef configured.	
<b>Synopsis</b>	<code>Std_ReturnType SoAd_TcpTlsAccepted ( TcpIp_SocketIdType SocketIdConnected );</code>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non reentrant	
<b>Parameters (in)</b>	SocketIdConnected	Socket handle for the established connection
<b>Return Value</b>	Result of operation	
	E_OK	SoAd accepts the established connection.
	E_NOT_OK	SoAd refuses the established connection, Tcplp stack shall close the connection.



#### **5.6.3.2.36. SoAd\_TpCancelReceive**

<b>Purpose</b>	Cancel reception.	
<b>Synopsis</b>	<code>Std_ReturnType SoAd_TpCancelReceive ( PduIdType PduId );</code>	
<b>Service ID</b>	0x06	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different PDU IDs. Non reentrant for the same PDU ID.	
<b>Parameters (in)</b>	PduId	Identification of the I-PDU to be cancelled
<b>Return Value</b>	Std_ReturnType E_OK                      The request was accepted (but not yet performed). E_NOT_OK                The request was not accepted (e.g. cancellation not possible).	
<b>Description</b>	Requests cancellation of the reception via TP for a specific I-PDU.	

#### **5.6.3.2.37. SoAd\_TpCancelTransmit**

<b>Purpose</b>	Cancel transmission.	
<b>Synopsis</b>	<code>Std_ReturnType SoAd_TpCancelTransmit ( PduIdType PduId );</code>	
<b>Service ID</b>	0x05	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different PDU IDs. Non reentrant for the same PDU ID.	
<b>Parameters (in)</b>	PduId	Identification of the I-PDU to be cancelled
<b>Return Value</b>	Std_ReturnType E_OK E_NOT_OK	
	E_OK	The request was accepted (but not yet performed).
	E_NOT_OK	The request was not accepted (e.g. cancellation not possible).
<b>Description</b>	Requests cancellation of the transmission via TP for a specific I-PDU.	

#### **5.6.3.2.38. SoAd\_TpChangeParameter**

<b>Purpose</b>	Change a parameter of the TCP/IP stack.
----------------	---



<b>Synopsis</b>	<code>Std_ReturnType SoAd_TpChangeParameter ( PduIdType Id , TPPParameterType Parameter , uint16 Value );</code>	
<b>Service ID</b>	0x11	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SoConIds. Non reentrant for the same SoConId.	
<b>Parameters (in)</b>	Id	Identification of the I-PDU that the parameter change shall affect
	Parameter	Identifier of the parameter to be changed
	Value	New parameter value
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request was accepted.
	E_NOT_OK	The request was not accepted.
<b>Description</b>	This API service requests the SoAd or TCP/IP stack to change a connection parameter. For example, the Nagle algorithm may be controlled by this API.	

#### 5.6.3.2.39. SoAd\_TpTransmit

<b>Purpose</b>	Transfer data.	
<b>Synopsis</b>	<code>Std_ReturnType SoAd_TpTransmit ( PduIdType SoAdSrcPduId , const PduInfoType * SoAdSrcPduInfoPtr );</code>	
<b>Service ID</b>	0x04	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SoAdSrcPduId	This parameter contains a unique identifier referencing to the PDU Routing Table and thereby specifying the socket to be used for transmission of the data.
	SoAdSrcPduInfoPtr	A pointer to a structure with socket-related data. Only the length data is valid.
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request was accepted.
	E_NOT_OK	The request was not accepted (e.g. due to a still ongoing transmission in the corresponding socket or the message to be transmitted is too long).



<b>Description</b>	This service is used to request the transfer of data.
--------------------	---

#### 5.6.3.2.40. SoAd\_TxConfirmation

<b>Purpose</b>	Acknowledges transmitted data.	
<b>Synopsis</b>	<code>void SoAd_TxConfirmation ( TcpIp_SocketIdType SocketId , uint16 Length );</code>	
<b>Service ID</b>	0x14	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non reentrant	
<b>Parameters (in)</b>	SocketId	Socket handle identifying the local socket resource
	Length	Number of transmitted data bytes
<b>Description</b>	The TCP/IP stack calls this function after the data was acknowledged by the peer for TCP or was sent to the lower layer driver using UDP.	

#### 5.6.3.2.41. SoAd\_WriteDhcpHostNameOption

<b>Purpose</b>	Write the DHCP host name.	
<b>Synopsis</b>	<code>Std_ReturnType SoAd_WriteDhcpHostNameOption ( SoAd_SoConIdType SoConId , uint8 Length , const uint8 * Data );</code>	
<b>Service ID</b>	0x1B	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SoConIds. Non reentrant for the same SoConId.	
<b>Parameters (in)</b>	SoConId	Socket connection index specifying the socket connection for which the parameter shall be changed
	Length	Length of parameter value to be set. This will be overwritten with the length of the actual data.
	Data	Pointer to the memory containing the new parameter value
<b>Return Value</b>	Std_ReturnType	
	E_OK	The request was successful.



	E_NOT_OK	The request was not successful.
Description	By this API service, an upper layer of the SoAd can set the host name option in the DHCP submodule of the TCP/IP stack.	

## 5.6.4. Integration notes

### 5.6.4.1. Exclusive areas

This section describes the exclusive areas used by the SoAd module.

#### 5.6.4.1.1. SOAD\_ENTER\_CRITSEC()

Protected data structures	All shared data that shall be protected from mutual access.
Recommended locking mechanism	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.

### 5.6.4.2. Production errors

Production errors are not reported by the SoAd module.

### 5.6.4.3. 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
CODE
VAR_INIT_8
VAR_INIT_32



VAR_INIT_UNSPECIFIED
VAR_CLEARED_UNSPECIFIED
VAR_CLEARED_8
VAR_CLEARED_32
CONST_8
CONST_32
CONST_UNSPECIFIED
CONFIG_DATA_UNSPECIFIED

#### 5.6.4.4. 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.

---

##### 5.6.4.4.1. lim.SoAd.EB\_INTREQ\_SoAd\_0001

<b>Description</b>	<p>The integrator must ensure that SoAd lower layer functions and SoAd main functions do not preempt each other. List of SoAd lower layer functions:</p> <ul style="list-style-type: none"><li>▶ SoAd_RxIndication()</li><li>▶ SoAd_TxConfirmation()</li><li>▶ SoAd_TcpTlsAccepted()</li><li>▶ SoAd_TcpAccepted()</li><li>▶ SoAd_TcpConnected()</li><li>▶ SoAd_TcplpEvent()</li><li>▶ SoAd_LocallpAddrAssignmentChg()</li></ul> <p>List of SoAd main functions:</p> <ul style="list-style-type: none"><li>▶ SoAd_MainFunction()</li><li>▶ SoAd_MainFunctionTx()</li></ul> <p>This can be achieved by setting EthIf and Eth driver in polling mode, e.g. by disabling of:</p> <ul style="list-style-type: none"><li>▶ EthIfEnableRxInterrupt</li></ul>
--------------------	---



	<ul style="list-style-type: none"> <li>▶ EthCtrlEnableRxInterrupt</li> <li>▶ EthIfEnableTxInterrupt</li> <li>▶ EthCtrlEnableTxInterrupt</li> </ul> <p>to enforce that lower layer APIs are called only in the context of main functions. In addition, all Eth stack main functions, e.g.:</p> <ul style="list-style-type: none"> <li>▶ EthIf_MainFunctionRx()</li> <li>▶ EthIf_MainFunctionTx()</li> <li>▶ EthIf_MainFunctionState()</li> <li>▶ TcpIp_MainFunction()</li> <li>▶ TcpIp_MainFunctionTx()</li> <li>▶ EthSM_MainFunction()</li> </ul> <p>must be in the same Os task or have the same task priority to eliminate preemption.</p> <p>Exception: SoAd_CopyTxData() can be called in the context of SoAd_IfTransmit() and is therefore excluded from this integration requirement.</p> <p>This integration requirement also applies to possible other not listed SoAd lower layer functions.</p>
<b>Rationale</b>	This limitation reduces code size and execution time by eliminating the need for extensive use of exclusive areas.

#### 5.6.4.4.2. lim.SoAd.EB\_INTREQ\_SoAd\_0003

<b>Description</b>	<p>The integrator must ensure that the following SoAd upper layer functions shall not preempt or be preempted by each other or the functions defined in lim.SoAd.EB_IN-TREQ_SoAd_0001. List of SoAd upper layer functions:</p> <ul style="list-style-type: none"> <li>▶ SoAd_SetRemoteAddr()</li> <li>▶ SoAd_SetUniqueRemoteAddr()</li> <li>▶ SoAd_ReleaseRemoteAddr()</li> <li>▶ SoAd_IfRoutingGroupTransmit()</li> <li>▶ SoAd_IfSpecificRoutingGroupTransmit()</li> <li>▶ SoAd_GetSoConId()</li> <li>▶ SoAd_CloseSoCon()</li> </ul>
--------------------	---



	<ul style="list-style-type: none"> <li>➤ SoAd_OpenSoCon()</li> <li>➤ SoAd_RequestIpAddrAssignment()</li> <li>➤ SoAd_ReleaseIpAddrAssignment()</li> <li>➤ SoAd_GetLocalAddr()</li> <li>➤ SoAd_GetPhysAddr()</li> <li>➤ SoAd_GetRemoteAddr()</li> <li>➤ SoAd_EnableRouting()</li> <li>➤ SoAd_EnableSpecificRouting()</li> <li>➤ SoAd_DisableRouting()</li> <li>➤ SoAd_DisableSpecificRouting()</li> <li>➤ SoAd_TpChangeParameter()</li> <li>➤ SoAd_ReadDhcpHostNameOption()</li> <li>➤ SoAd_WriteDhcpHostNameOption()</li> <li>➤ SoAd_GetSoConMode()</li> <li>➤ SoAd_ChangeParameter()</li> <li>➤ SoAd_IsConnectionReady()</li> </ul> <p>Exception: The following functions are excluded:</p> <ul style="list-style-type: none"> <li>➤ SoAd_IfTransmit()</li> <li>➤ SoAd_TpTransmit()</li> <li>➤ SoAd_TpCancelTransmit()</li> <li>➤ SoAd_TpCancelReceive()</li> </ul> <p>Exception: A preemption of SoAd upper layer functions with SoAd upper layer functions might be possible if the call occurs in the same context. Example: SoAd_TxConfirmation() calls UL_TxConfirmation(), which calls SoAd_ReleaseRemoteAddr() causing a preemption of SoAd_TxConfirmation() in the same call context.</p> <p>This integration requirement also applies to possible other not listed SoAd upper layer functions.</p>
<b>Rationale</b>	This limitation reduces code size and execution time by eliminating the need for extensive use of exclusive areas.



#### 5.6.4.4.3. lim.SoAd.EB\_INTREQ\_SoAd\_0004

<b>Description</b>	The reinitialization process shall not interrupt other module functions. If reinitialization of the module is required, the call of SoAd_Init() shall not interrupt other module functions.
<b>Rationale</b>	The reinitialization process resets all internal variables. Continuing an interrupted module function after reinitialization can lead to undefined module behavior.

#### 5.6.4.4.4. lim.SoAd.EB\_INTREQ\_SoAd\_0006

<b>Description</b>	If SoAd_IfTransmit() with SduDataPtr = NULL_PTR is called for a UDP connection, the optional header and payload to be transmitted must fit into a single UDP frame. It must be ensured that IP fragmentation does not occur. Otherwise a Det error is reported.
--------------------	---

#### 5.6.4.4.5. lim.SoAd.EB\_INTREQ\_SoAd\_0008

<b>Description</b>	If SoAd_SetUniqueRemoteAddr() is called, it must be assured that SoAd_ReleaseRemoteAddr() is called as often for the same SoConId before the remote address is released.
--------------------	--

#### 5.6.4.4.6. lim.SoAd.EB\_INTREQ\_SoAd\_0009

<b>Description</b>	If SoAd_SetUniqueRemoteAddr() is called and returns a socket connection set with SoAd_SetRemoteAddr(), it must be assured that SoAd_ReleaseRemoteAddr() is called as often for the same SoConId before the remote address can be set with SoAd_SetRemoteAddr() again.
--------------------	---

#### 5.6.4.4.7. lim.SoAd.EB\_INTREQ\_SoAd\_0010

<b>Description</b>	SoAd performs transmission over a TCP connection as a data stream. If a TP PDU transmission using this TCP connection gets aborted, the TCP connection will be closed. An ongoing transmission for a TP PDU can be aborted either via SoAd_TpCancelTransmit() or by returning BUFREQ_E_NOT_OK for UL_SoAdTpCopyTxData() when requesting available data which will be called before requesting the actual data.
--------------------	--



#### 5.6.4.4.8. lim.SoAd.EB\_INTREQ\_SoAd\_0011

<b>Description</b>	SoAd does not support time-out supervision of upper layer TP transmission data provision. If data are not immediately available, the upper layer must ensure that the call of UL_SoAdTpCopyTxData() does not return BUFREQ_E_OK with available data equal to 0 or BUFREQ_E_BUSY for an infinite time.
--------------------	---

#### 5.6.4.4.9. lim.SoAd.EB\_INTREQ\_SoAd\_0012

<b>Description</b>	The SoAd_IfTransmit() API must be called in a context with higher priority than the priority of SoAd_MainFunction() Os task.
--------------------	--

#### 5.6.4.4.10. lim.SoAd.EB\_INTREQ\_SoAd\_0013

<b>Description</b>	The integrator must ensure for SoAdSocketRoutes and SoAdPduRoutes referring a SoAdSocketConnectionGroup and refer to multiple SoAdRoutingGroups that either all of these SoAdRoutingGroups are only referred by SoAdSocketRoutes and SoAdPduRoutes referring the same SoAdSocketConnectionGroup and can be controlled via SoAd_EnableSpecificRouting() and SoAd_DisableSpecificRouting() and therefore called specific SoAdRoutingGroups or all of these SoAdRoutingGroups are only referred SoAdSocketRoutes and SoAdPduRoutes referring to different SoAdSocketConnectionGroup or SoAdSocketConnection within each SoAdRoutingGroup and can be controlled by SoAd_EnableRouting() and SoAd_DisableRouting() and therefore called global SoAdRoutingGroups. It must be ensured that a SoAdSocketRoutes or SoAdPduRoutes does not refer to a mixture of specific and global SoAdRoutingGroups.
--------------------	--

## 5.7. SomeTp

### 5.7.1. Configuration parameters

Containers included		
Container name	Multiplicity	Description
<a href="#">CommonPublishedInformation</a>	1..1	<b>Label:</b> Common Published Information



### Containers included

		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.
<a href="#">SomelTpChannel</a>	1..n	This container contains the configuration parameters of the SomelTp channel.
<a href="#">SomelTpUnSupportedChannel</a>	1..1	
<a href="#">SomelTpGeneral</a>	1..1	This container contains the general configuration parameters of the SomelTp module.

### Parameters included

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

Parameter Name	<b>IMPLEMENTATION_CONFIG_VARIANT</b>	
<b>Label</b>	Config Variant	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	VariantLinkTime	
<b>Range</b>	VariantPreCompile	
<b>Configuration class</b>	<b>VariantLinkTime:</b>	VariantLinkTime
	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile

### 5.7.1.1. CommonPublishedInformation

#### Parameters included

Parameter name	Multiplicity
<a href="#">ArMajorVersion</a>	1..1
<a href="#">ArMinorVersion</a>	1..1
<a href="#">ArPatchVersion</a>	1..1



#### Parameters included

<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	<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

Parameter Name	<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>	3
<b>Configuration class</b>	<b>PublishedInformation:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

Parameter Name	<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>	0
<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>	30
<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>	177
<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 AUTOSAR 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

### 5.7.1.2. 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 SomeIpTp 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

### 5.7.1.3. SomeIpTpChannel

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">SomeIpTpRxNSdu</a>	0..n	The following parameters needs to be configured for each N-SDU which has to be passed as one assembled RxPdu to the upper layer.
<a href="#">SomeIpTpTxNSdu</a>	0..n	The following parameters needs to be configured for each N-SDU that the SomeIpTp module transmits via the SomeIpTpChannel.

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

<b>Parameter Name</b>	<a href="#">SomeIpTpNPduSeparationTime</a>	
<b>Description</b>	Sets the duration of the minimum time in seconds the SomeIpTp module shall wait between the transmissions of N-PDUs.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Configuration class</b>	<b>VariantLinkTime:</b>	VariantLinkTime
	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	



<b>Parameter Name</b>	<b>SomelpTpRxTimeoutTime</b>	
<b>Description</b>	Timer to monitor the successful reception. It is started when the first NPdu is received, and is stopped when the last NPdu has been received.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Configuration class</b>	<b>VariantLinkTime:</b> VariantLinkTime <b>VariantPostBuild:</b> VariantPostBuild <b>VariantPreCompile:</b> VariantPreCompile	
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SomelpTpTxConfirmationTimeout</b>	
<b>Description</b>	Timeout in seconds to monitor a successful transmission of an Npdu, and is stopped when the its TxConfirmation is received. It is added to have compatibility with PduR 4.0.3 IfTxConfirmation API (i.e. SomelpTp_TxConfirmation) which is called only if successful transmission occurs without result argument, otherwise if transmission failed no SomelpTp_TxConfirmation will be called, then a timeout is needed to handle the failure case. Note: this timeout shall include the time for Tp retry mechanism for this Npdu if needed	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Configuration class</b>	<b>VariantLinkTime:</b> VariantLinkTime <b>VariantPostBuild:</b> VariantPostBuild <b>VariantPreCompile:</b> VariantPreCompile	
<b>Origin</b>	Elektrobit Automotive GmbH	

#### 5.7.1.4. SomelpTpRxNSdu

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">SomelpTpRxNPdu</a>	1..1	This container contains the configuration parameters of the NPdu that is received from a lower layer

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



<b>Parameter Name</b>	<a href="#">SomelpTpRxSduRef</a>	
<b>Description</b>	Reference to a Pdu in the COM-Stack that represents the assembled RxPdu which is passed via the PduR to the upper layer.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>VariantLinkTime:</b>	VariantLinkTime
	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

### 5.7.1.5. SomelpTpRxNPdu

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

<b>Parameter Name</b>	<a href="#">SomelpTpRxNPduHandleId</a>	
<b>Description</b>	This parameter defines the handle ID that is used by the PduR when calling SomelpTp_RxIndication.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>VariantLinkTime:</b>	VariantLinkTime
	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<a href="#">SomelpTpRxNPduRef</a>	
<b>Description</b>	Reference to a global Pdu that is used to harmonize HandleIDs in the COM-Stack.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>VariantLinkTime:</b>	VariantLinkTime
	<b>VariantPostBuild:</b>	VariantPostBuild



	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

### 5.7.1.6. SomelpTpTxNSdu

Containers included		
Container name	Multiplicity	Description
<a href="#">SomelpTpTxNPdu</a>	1..1	This container contains the configuration parameters of the segmented Tx NPdus that are transmitted to a lower layer.

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

<b>Parameter Name</b>	<b>SomelpTpTxNSduHandleId</b>	
<b>Description</b>	This parameter defines the handle ID of the NSdu that represents the original TxSdu which is segmented and passed via the PduR to the lower layer. This handle ID is used by PduR when calling SomelpTp_Transmit.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>VariantLinkTime:</b>	VariantLinkTime
	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SomelpTpTxNSduRef</b>	
<b>Description</b>	Reference to a global Pdu in the COM-Stack that represents the original TxSdu which is segmented and passed via the PduR to the lower layer.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>VariantLinkTime:</b>	VariantLinkTime
	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	



### 5.7.1.7. SomelpTpTxNPdu

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

Parameter Name	<a href="#">SomelpTpTxNPduHandleId</a>	
Description	This parameter defines the handle ID that is used by PduR when calling SomelpTp_TriggerTransmit.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	<b>VariantLinkTime:</b>	VariantLinkTime
	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	<a href="#">SomelpTpTxNPduRef</a>	
Description	Reference to a global Pdu that is used to harmonize HandleIDs in the COM-Stack.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	<b>VariantLinkTime:</b>	VariantLinkTime
	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
Origin	AUTOSAR_ECUC	

### 5.7.1.8. SomelpTpUnSupportedChannel

Containers included		
Container name	Multiplicity	Description
<a href="#">SomelpTpUnSupportedRxNsdu</a>	1..1	
<a href="#">SomelpTpUnSupportedTxNsdu</a>	1..1	



### 5.7.1.9. SomeIpTpUnSupportedRxNsdu

Containers included		
Container name	Multiplicity	Description
<a href="#">SomeIpTpUnSupportedRxN-Pdu</a>	1..1	

### 5.7.1.10. SomeIpTpUnSupportedRxNPdu

### 5.7.1.11. SomeIpTpUnSupportedTxNsdu

Containers included		
Container name	Multiplicity	Description
<a href="#">SomeIpTpUnSupportedTxN-Pdu</a>	1..1	

### 5.7.1.12. SomeIpTpUnSupportedTxNPdu

### 5.7.1.13. SomeIpTpGeneral

Parameters included	
Parameter name	Multiplicity
<a href="#">SomeIpTpDevErrorDetect</a>	1..1
<a href="#">SomeIpTpRxMainFunctionPeriod</a>	1..1
<a href="#">SomeIpTpTxMainFunctionPeriod</a>	1..1
<a href="#">SomeIpTpVersionInfoApi</a>	1..1

Parameter Name	<b>SomeIpTpDevErrorDetect</b>
Description	Switches the Development Error Detection and Notification ON or OFF.
Multiplicity	1..1
Type	BOOLEAN



<b>Configuration class</b>	<b>VariantLinkTime:</b>	VariantLinkTime
	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SomelpTpRxMainFunctionPeriod</b>	
<b>Description</b>	This parameter defines the cycle time in seconds of the periodic call of the SomelpTp_MainFunctionRx.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Range</b>	<Infinity >0.0	
<b>Configuration class</b>	<b>VariantLinkTime:</b>	VariantLinkTime
	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SomelpTpTxMainFunctionPeriod</b>	
<b>Description</b>	This parameter defines the cycle time in seconds of the periodic call of the SomelpTp_MainFunctionTx.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Range</b>	<Infinity >0.0	
<b>Configuration class</b>	<b>VariantLinkTime:</b>	VariantLinkTime
	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>SomelpTpVersionInfoApi</b>	
<b>Description</b>	Activates the SomelpTp_GetVersionInfo() API.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	



<b>Configuration class</b>	<b>VariantLinkTime:</b>	VariantLinkTime
	<b>VariantPostBuild:</b>	VariantPostBuild
	<b>VariantPreCompile:</b>	VariantPreCompile
<b>Origin</b>	AUTOSAR_ECUC	

## 5.7.2. Application programming interface (API)

### 5.7.2.1. Macro constants

#### 5.7.2.1.1. SOMEIPTP\_DET\_REPORT\_ERROR

<b>Purpose</b>	
<b>Value</b>	(void) Det_ReportError(SOMEIPTP_MODULE_ID, SOMEIPTP_INSTANCE_ID, (ApId), (ErrorId))

#### 5.7.2.1.2. SOMEIPTP\_DET\_REPORT\_RUNTIME\_ERROR

<b>Purpose</b>	
<b>Value</b>	(void) Det_ReportRuntimeError(SOMEIPTP_MODULE_ID, SOMEIPTP_INSTANCE_ID, (ApId), (ErrorId))

#### 5.7.2.1.3. SOMEIPTP\_E\_ASSEMBLY\_INTERRUPT

<b>Purpose</b>	Definition of DET error code SOMEIPTP_E_ASSEMBLY_INTERRUPT.
<b>Value</b>	0x08U

#### 5.7.2.1.4. SOMEIPTP\_E\_ASSEMBLY\_INTERRUPT\_SOR

<b>Purpose</b>	Definition of DET error code SOMEIPTP_E_ASSEMBLY_INTERRUPT for start of reception.
<b>Value</b>	0x80U



#### 5.7.2.1.5. SOMEIPTP\_E\_DISASSEMBLY\_INTERRUPT

Purpose	Definition of DET error code SOMEIPTP_E_DISASSEMBLY_INTERRUPT.
Value	0x07U

#### 5.7.2.1.6. SOMEIPTP\_E\_INCONSISTENT\_HEADER

Purpose	Definition of DET error code SOMEIPTP_E_INCONSISTENT_HEADER.
Value	0x06U

#### 5.7.2.1.7. SOMEIPTP\_E\_INCONSISTENT\_SEQUENCE

Purpose	Definition of DET error code SOMEIPTP_E_INCONSISTENT_SEQUENCE.
Value	0x05U

#### 5.7.2.1.8. SOMEIPTP\_E\_INVALID\_CALL

Purpose	Definition of DET error code SOMEIPTP_E_INVALID_CALL.
Value	0xFF

#### 5.7.2.1.9. SOMEIPTP\_E\_MESSAGE\_TYPE

Purpose	Definition of DET error code SOMEIPTP_E_MESSAGE_TYPE.
Value	0x04U

#### 5.7.2.1.10. SOMEIPTP\_E\_MISSING\_SOMEIPTP\_HEADER

Purpose	Definition of DET error code SOMEIPTP_E_MISSING_SOMEIPTP_HEADER.
Value	0x09U

#### 5.7.2.1.11. SOMEIPTP\_E\_NOTINIT

Purpose	Definition of DET error code SOMEIPTP_E_NOTINIT.
---------	--



<b>Value</b>	0x01U
--------------	-------

#### 5.7.2.1.12. SOMEIPTP\_E\_PARAM

<b>Purpose</b>	Definition of DET error code SOMEIPTP_E_PARAM.
<b>Value</b>	0x03U

#### 5.7.2.1.13. SOMEIPTP\_E\_PARAM\_POINTER

<b>Purpose</b>	Definition of DET error code SOMEIPTP_E_PARAM_POINTER.
<b>Value</b>	0x02U

#### 5.7.2.1.14. SOMEIPTP\_INSTANCE\_ID

<b>Purpose</b>	Module instance ID.
<b>Value</b>	0U
<b>Description</b>	Defines the instance number of this module. Since multiple instances are not supported this ID is always zero.

#### 5.7.2.1.15. SOMEIPTP\_SID\_GETVERSIONINFO

<b>Purpose</b>	Defines API id of function <a href="#">SomelpTp_GetVersionInfo()</a> .
<b>Value</b>	0x01U

#### 5.7.2.1.16. SOMEIPTP\_SID\_INIT

<b>Purpose</b>	Defines API id of function <a href="#">SomelpTp_Init()</a> .
<b>Value</b>	0x02U

#### 5.7.2.1.17. SOMEIPTP\_SID\_MAINFUNCTIONRX

<b>Purpose</b>	Defines API id of function <a href="#">SomelpTp_MainFunctionRx()</a> .
<b>Value</b>	0x04U



#### 5.7.2.1.18. SOMEIPTP\_SID\_MAINFUNCTIONTX

Purpose	Defines API id of function <a href="#">SomelpTp_MainFunctionTx()</a> .
Value	0x03U

#### 5.7.2.1.19. SOMEIPTP\_SID\_PDURIFTRANSMIT

Purpose	Defines Dummy API id of function <a href="#">SomelpTp_PduRIfTransmit()</a> .
Value	0xFF

#### 5.7.2.1.20. SOMEIPTP\_SID\_PDURTPCOPYRXDATA

Purpose	Defines Dummy API id of function <a href="#">SomelpTp_PduRTpCopyRxData()</a> .
Value	0xFB

#### 5.7.2.1.21. SOMEIPTP\_SID\_PDURTPCOPYTXDATA

Purpose	Defines Dummy API id of function <a href="#">SomelpTp_PduRTpCopyTxData()</a> .
Value	0xFD

#### 5.7.2.1.22. SOMEIPTP\_SID\_PDURTPRXINDICATION

Purpose	Defines Dummy API id of function <a href="#">SomelpTp_PduRTpRxIndication()</a> .
Value	0xFA

#### 5.7.2.1.23. SOMEIPTP\_SID\_PDURTPSTARTOFRECEPTION

Purpose	Defines Dummy API id of function <a href="#">SomelpTp_PduRTpStartOfReception()</a> .
Value	0xFC

#### 5.7.2.1.24. SOMEIPTP\_SID\_PDURTPTXCONFIRMATION

Purpose	Defines Dummy API id of function <a href="#">SomelpTp_PduRTpTxConfirmation()</a> .
---------	--



<b>Value</b>	0xFE
--------------	------

#### 5.7.2.1.25. SOMEIPTP\_SID\_RXINDICATION

<b>Purpose</b>	Defines API id of function <a href="#">SomelpTp_RxIndication()</a> .
<b>Value</b>	0x42U

#### 5.7.2.1.26. SOMEIPTP\_SID\_TRANSMIT

<b>Purpose</b>	Defines API id of function <a href="#">SomelpTp_Transmit()</a> .
<b>Value</b>	0x49U

#### 5.7.2.1.27. SOMEIPTP\_SID\_TRIGGERTRANSMIT

<b>Purpose</b>	Defines API id of function <a href="#">SomelpTp_TriggerTransmit()</a> .
<b>Value</b>	0x41U

#### 5.7.2.1.28. SOMEIPTP\_SID\_TXCONFIRMATION

<b>Purpose</b>	Defines API id of function <a href="#">SomelpTp_TxConfirmation()</a> .
<b>Value</b>	0x40U

### 5.7.2.2. Functions

#### 5.7.2.2.1. SomelpTp\_GetVersionInfo

<b>Purpose</b>	API to get the version information of SomelpTp module.	
<b>Synopsis</b>	<pre>void SomeIpTp_GetVersionInfo ( Std_VersionInfoType * VersionInfo );</pre>	
<b>Parameters (out)</b>	VersionInfo	- Pointer to where to store the version information of this module.
<b>Description</b>	This service returns the version information of this module.	



### 5.7.2.2.2. SomelpTp\_Init

<b>Purpose</b>	Initializes the SomelpTp module.	
<b>Synopsis</b>	<code>void SomeIpTp_Init ( const SomeIpTp_ConfigType * config );</code>	
<b>Parameters (in)</b>	config	- Base pointer to the configuration structure of the SOME/IP TP module.
<b>Description</b>	This service initializes the SomelpTp module. It shall be the first function of the module to be called.	

### 5.7.2.2.3. SomelpTp\_MainFunctionRx

<b>Purpose</b>	SomelpTp module main function for reception.
<b>Synopsis</b>	<code>void SomeIpTp_MainFunctionRx ( void );</code>
<b>Description</b>	This function performs the processing of the AUTOSAR SOME/IP TP module's reception activities.

### 5.7.2.2.4. SomelpTp\_MainFunctionTx

<b>Purpose</b>	SomelpTp module main function for Transmission.
<b>Synopsis</b>	<code>void SomeIpTp_MainFunctionTx ( void );</code>
<b>Description</b>	This function performs the processing of the AUTOSAR SOME/IP TP module's transmission activities.

### 5.7.2.2.5. SomelpTp\_PduRIfTransmit

<b>Purpose</b>	Dummy funtion for PduR.
<b>Synopsis</b>	<code>Std_ReturnType SomeIpTp_PduRIfTransmit ( PduIdType TxPduId , const PduInfoType * PduInfoPtr );</code>
<b>Service ID</b>	0xFF
<b>Return Value</b>	
<b>Description</b>	This function shall not be called, and if so it shall report developement error.



#### 5.7.2.2.6. SomeIpTp\_PduRTpCopyRxData

<b>Purpose</b>	Dummy funtion for PduR.
<b>Synopsis</b>	BufReq_ReturnType <b>SomeIpTp_PduRTpCopyRxData</b> ( PduIdType PduId , const PduInfoType * PduInfoPointer , PduLengthType * RxBufferSizePtr );
<b>Service ID</b>	0xFB
<b>Return Value</b>	
<b>Description</b>	This function shall not be called, and if so it shall report developement error.

#### 5.7.2.2.7. SomeIpTp\_PduRTpCopyTxData

<b>Purpose</b>	Dummy funtion for PduR.
<b>Synopsis</b>	BufReq_ReturnType <b>SomeIpTp_PduRTpCopyTxData</b> ( PduIdType PduId , PduInfoType * PduInfoPtr , RetryInfoType * RetryInfoPtr , PduLengthType * TxDataCntPtr );
<b>Service ID</b>	0xFD
<b>Return Value</b>	
<b>Description</b>	This function shall not be called, and if so it shall report developement error.

#### 5.7.2.2.8. SomeIpTp\_PduRTpRxIndication

<b>Purpose</b>	Dummy funtion for PduR.
<b>Synopsis</b>	void <b>SomeIpTp_PduRTpRxIndication</b> ( PduIdType RxPduId , NotifResultType Result );
<b>Service ID</b>	0xFA
<b>Description</b>	This function shall not be called, and if so it shall report developement error.

#### 5.7.2.2.9. SomeIpTp\_PduRTpStartOfReception

<b>Purpose</b>	Dummy funtion for PduR.
----------------	-------------------------



<b>Synopsis</b>	<pre>BufReq_ReturnType SomeIpTp_PduRTpStartOfReception ( PduIdType PduId , PduLengthType PduLength , PduLengthType * RxBuffer- SizePtr );</pre>
<b>Service ID</b>	0xFC
<b>Return Value</b>	
<b>Description</b>	This function shall not be called, and if so it shall report development error.

#### 5.7.2.2.10. SomelpTp\_PduRTpTxConfirmation

<b>Purpose</b>	Dummy function for PduR.
<b>Synopsis</b>	<pre>void SomeIpTp_PduRTpTxConfirmation ( PduIdType PduId , NotifRe- sultType Result );</pre>
<b>Service ID</b>	0xFE
<b>Description</b>	This function shall not be called, and if so it shall report development error.

#### 5.7.2.2.11. SomelpTp\_RxIndication

<b>Purpose</b>	SomelpTp module reception indication callback function.	
<b>Synopsis</b>	<pre>void SomeIpTp_RxIndication ( PduIdType RxPduId , PduInfoType * PduInfoPtr );</pre>	
<b>Parameters (in)</b>	RxPduId	- ID received PDU.
	PduInfoPtr	- pointer to a buffer containing the PDU data, and length of the received PDU.
<b>Description</b>	This callback function is called to indicate a received PDU from a lower layer communication interface module	

#### 5.7.2.2.12. SomelpTp\_Transmit

<b>Purpose</b>	SomelpTp module Transmit request function.
<b>Synopsis</b>	<pre>Std_ReturnType SomeIpTp_Transmit ( PduIdType TxPduId , const PduInfoType * PduInfoPtr );</pre>
<b>Service ID</b>	0x49
<b>Sync/Async</b>	Synchronous



<b>Reentrancy</b>	Reentrant for different Pdulds Non Reentrant for the same Pduld	
<b>Parameters (in)</b>	TxPduId	- Identifier of the PDU to be transmitted.
	PduInfoPtr	- Length of and pointer to the PDU data and pointer to MetaData.
<b>Return Value</b>		
Std_ReturnType	E_OK if transmit request has been accepted. E_NOT_OK if not accepted.	
<b>Description</b>	This service requests transmission of a PDU.	

#### 5.7.2.2.13. SomeIpTp\_TriggerTransmit

<b>Purpose</b>	SomeIpTp module trigger transmit callback function.	
<b>Synopsis</b>	Std_ReturnType <b>SomeIpTp_TriggerTransmit</b> ( PduIdType TxPduId , PduInfoType * PduInfoPtr );	
<b>Service ID</b>	0x41	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different Pdulds Reentrant for the same Pduld	
<b>Parameters (in)</b>	TxPduId	- ID of the SDU that is requested to be transmitted.
	[inout]	PduInfoPtr - pointer to a buffer to where the SDU data shall be copied, and the available buffer size in SduLength. On return, the service will indicate the length of the copied SDU data in SduLength.
<b>Parameters (in,out)</b>	[inout]	PduInfoPtr - pointer to a buffer to where the SDU data shall be copied, and the available buffer size in SduLength. On return, the service will indicate the length of the copied SDU data in SduLength.
<b>Return Value</b>		
Std_ReturnType	E_OK if SDU has been copied. E_NOT_OK if not copied.	
<b>Description</b>	This callback is called by the upper layer to check whether the available data fits into the buffer size If it fits, it shall copy its data into the buffer and update the length of the actual copied data	



#### 5.7.2.2.14. SomeIpTp\_TxConfirmation

<b>Purpose</b>	SomeIpTp module Transmission confirmation callback function.	
<b>Synopsis</b>	<code>void SomeIpTp_TxConfirmation ( PduIdType TxPduId );</code>	
<b>Parameters (in)</b>	TxPduId	- ID of the PDU that has been transmitted.
<b>Description</b>	This callback function is called by the lower interface module to confirm the transmission of a PDU if succeeded or failed.	

### 5.7.3. Integration notes

#### 5.7.3.1. Exclusive areas

This section describes the exclusive areas used by the `SomeIpTp` module.

##### 5.7.3.1.1. SCHM\_SOMEIPTP\_EXCLUSIVE\_AREA\_0

<b>Protected data structures</b>	All shared data in the Rx path 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 Mapping exclusive areas in the basic software modules in the Integration notes section for details.

##### 5.7.3.1.2. SCHM\_SOMEIPTP\_EXCLUSIVE\_AREA\_1

<b>Protected data structures</b>	All shared data in the Tx path 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 Mapping exclusive areas in the basic software modules in the Integration notes section for details.



### 5.7.3.2. Production errors

Production errors are not reported by the `SomeIpTp` module.

### 5.7.3.3. 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
CONFIG_DATA_UNSPECIFIED
VAR_INIT_8
VAR_CLEARED_UNSPECIFIED
CODE

### 5.7.3.4. 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 `SomeIpTp` module.

## 5.8. TcpIp

### 5.8.1. Configuration parameters

Containers included		
Container name	Multiplicity	Description
<a href="#">CommonPublishedInformation</a>	1..1	<b>Label:</b> Common Published Information



### Containers included

		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.
<a href="#">TcplpConfig</a>	1..1	This container contains the configuration parameters and sub containers of the AUTOSAR Tcplp module.
<a href="#">TcplpDefensiveProgramming</a>	1..1	<b>Label:</b> Defensive Programming Options Parameters for defensive programming
<a href="#">TcplpGeneral</a>	1..1	This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack.

### Parameters included

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

Parameter Name	<b>IMPLEMENTATION_CONFIG_VARIANT</b>
Multiplicity	1..1
Type	ENUMERATION
Default value	VariantPostBuild
Range	VariantPostBuild

### 5.8.1.1. CommonPublishedInformation

#### Parameters included

Parameter name	Multiplicity
<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>	3
<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>	3
<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>	5
<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>	14
<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>	170



<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

### 5.8.1.2. 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 Tcpip 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	

### 5.8.1.3. TcplpConfig

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcplpCtrl</a>	1..n	Specifies the Ethlf controller used for IP communication and Tcplp errors that shall be reported to DEM.
<a href="#">TcplpDhcpServerConfig</a>	0..n	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Specifies the configuration parameters of the DHCP Server sub-module.</p>
<a href="#">TcplpIpConfig</a>	0..1	Specifies the configuration parameters of the IP (Internet Protocol) sub-module.
<a href="#">TcplpLocalAddr</a>	1..253	Specifies the local IP (Internet Protocol) addresses used for IP communication.
<a href="#">TcplpNvmBlock</a>	0..1	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Configuration of optional usage of Nvm in case the Tcplp module requires non volatile memory in the Ecu to store information (e.g. IP Address received via DHCP and shall be stored).</p>
<a href="#">TcplpPhysAddrConfig</a>	0..1	Specifies the physical address configuration.
<a href="#">TcplpSocketOwnerConfig</a>	1..1	Specifies the upper layer modules of Tcplp using the socket API.
<a href="#">TcplpTcpConfig</a>	0..1	Specifies the configuration parameters of the TCP (Transmission Control Protocol) sub-module.
<a href="#">TcplpUdpConfig</a>	0..1	This container is a subcontainer of TcplpConfig and specifies the configuration parameters of the UDP (User Datagram Protocol) sub-module.
<a href="#">TcplpDuplicateAddressDetectionConfig</a>	0..1	Specifies the DAD callout function
<a href="#">TcplpRxPolicyCheckIngressHandlerConfig</a>	0..1	Specifies the policy check ingress handler function



#### Containers included

<a href="#">TcplpRxPolicyCheckRouting-HandlerConfig</a>	0..1	Specifies the policy check routing handler function
<a href="#">TcplpRxPacketPost-ProcessHandlerConfig</a>	0..1	Specifies the post process handler function
<a href="#">TcplpRxPacketDropHandler-Config</a>	0..1	Specifies the packet drop handler function
<a href="#">TcplpMemoryConfig</a>	1..1	Specifies the configuration parameters for memory pools.
<a href="#">TcplpIpSecConfig</a>	0..1	Specifies the configuration parameters of the IpSec sub-module.

#### 5.8.1.4. TcplpCtrl

##### Containers included

Container name	Multiplicity	Description
<a href="#">TcplpOffloadChecksum</a>	1..1	This container is a subcontainer of TcplpCtrl and specifies the parameters of the software/hardware checksum calculations.
<a href="#">TcplpIpVXCtrl</a>	1..1	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Specifies whether this controller is an Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) instance.

##### Parameters included

Parameter name	Multiplicity
<a href="#">TcplpIpFramePrioDefault</a>	0..1
<a href="#">TcplpDhcpServerConfigRef</a>	0..1
<a href="#">TcplpEthIfCtrlRef</a>	1..1

Parameter Name	TcplpIpFramePrioDefault
<b>Description</b>	Specifies the default value for the frame priority used by all sockets.
<b>Multiplicity</b>	0..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Range</b>	<=7



	>=0
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	TcpIpDhcpServerConfigRef
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Reference to a TcpIpDhcpServerConfig which shall be used for this controller setting (VLAN).
<b>Multiplicity</b>	0..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	TcpIpEthIfCtrlRef
<b>Description</b>	Reference to EthIf controller where the IP address shall be assigned
<b>Multiplicity</b>	1..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.8.1.5. TcpIpOffloadChecksum

Parameters included	
Parameter name	Multiplicity
<a href="#">TcIpEnableOffloadChecksumIPv4</a>	1..1
<a href="#">TcIpEnableOffloadChecksumICMP</a>	1..1
<a href="#">TcIpEnableOffloadChecksumTCP</a>	1..1
<a href="#">TcIpEnableOffloadChecksumUDP</a>	1..1

<b>Parameter Name</b>	TcpIpEnableOffloadChecksumIPv4
<b>Description</b>	Enables (FALSE) or disables (TRUE) the IPv4 checksum calculation in software.
<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>TcpIpEnableOffloadChecksumICMP</b>
<b>Description</b>	Enables (FALSE) or disables (TRUE) the ICMP checksum calculation in software.
<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>TcpIpEnableOffloadChecksumTCP</b>
<b>Description</b>	Enables (FALSE) or disables (TRUE) the TCP checksum calculation in software.
<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>TcpIpEnableOffloadChecksumUDP</b>
<b>Description</b>	Enables (FALSE) or disables (TRUE) the UDP checksum calculation in software.
<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

### 5.8.1.6. TcpIpVXCtrl

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>



#### Containers included

<a href="#">TcplIpV4Ctrl</a>	1..1	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Specifies an Internet Protocol version 4 (IPv4) instance.
<a href="#">TcplIpV6Ctrl</a>	1..1	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Specifies an Internet Protocol version 6 (IPv6) instance.

#### 5.8.1.7. TcplIpV4Ctrl

##### Containers included

Container name	Multiplicity	Description
<a href="#">TcplIpV4MtuConfig</a>	1..1	<i>The functionality related to this parameter is not supported by the current implementation.</i>  This container specifies the Maximum Transmission Unit parameters for this IPv4 instance.

##### Parameters included

Parameter name	Multiplicity
<a href="#">TcplArpConfigRef</a>	0..1
<a href="#">TcplAutoIpConfigRef</a>	0..1
<a href="#">TcplDhcpConfigRef</a>	0..1
<a href="#">TcplFragmentationConfigRef</a>	0..1

Parameter Name	<a href="#">TcplArpConfigRef</a>	
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Reference to ARP configuration for this IPv4 instance.	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	<b>PreCompile:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<a href="#">TcplAutoIpConfigRef</a>
----------------	-------------------------------------



<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Reference to Autolp configuration for this IPv4 instance.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>TcpIpDhcpConfigRef</b>	
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Reference to DHCP configuration for this IPv4 instance.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>TcpIpFragmentationConfigRef</b>	
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Reference to Fragmentation configuration for this IPv4 instance.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.8.1.8. TcpIpV4MtuConfig

Parameters included		
Parameter name	Multiplicity	
<a href="#">TcplpV4PathMtuEnabled</a>	1..1	
<a href="#">TcplpV4PathMtuTimeout</a>	1..1	



<b>Parameter Name</b>	<b>TcpIpIpV4PathMtuEnabled</b>	
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>If enabled the IPv4 processes incoming ICMPv4 "Packet Too Big" messages and stores a MTU value for each destination address.</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>TcpIpIpV4PathMtuTimeout</b>	
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>If this value is &gt;0 the IpV4 will reset the MTU value stored for each destination after n seconds.</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	600.0	
<b>Range</b>	<=86400.0 >=1.0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.8.1.9. TcpIpIpV6Ctrl

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcplIpV6MtuConfig</a>	1..1	This container specifies the Maximum Transmission Unit parameters for this IPv6 instance.

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>



#### Parameters included

<a href="#">TcpllpV6DhcpConfigRef</a>	0..1
<a href="#">TcpllpV6FragmentationConfigRef</a>	0..1
<a href="#">TcpllpV6NdpConfigRef</a>	1..1

Parameter Name	TcpllpV6DhcpConfigRef
<b>Description</b>	Reference to DHCPv6 configuration.
<b>Multiplicity</b>	0..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

Parameter Name	TcpllpV6FragmentationConfigRef
<b>Description</b>	Reference to IPv6 Fragmentation Configuration
<b>Multiplicity</b>	0..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

Parameter Name	TcpllpV6NdpConfigRef
<b>Description</b>	Reference to Neighbor Discovery Protocol Configuration.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.8.1.10. TcpllpV6MtuConfig

Parameters included	
Parameter name	Multiplicity
<a href="#">TcpllpV6PathMtuEnabled</a>	1..1
<a href="#">TcpllpV6PathMtuTimeout</a>	1..1
Parameter Name	TcpllpV6PathMtuEnabled



<b>Description</b>	If enabled the IPv6 processes incoming ICMPv6 "Packet Too Big" messages and stores a MTU value for each destination address.	
<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>TcpIpIpV6PathMtuTimeout</b>	
<b>Description</b>	If this value is >0 the IpV6 will reset the MTU value stored for each destination after n seconds.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	600.0	
<b>Range</b>	<=86400.0 >=1.0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.8.1.11. TcplpDhcpServerConfig

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcplpDhcpAddressAssignment</a>	0..n	Defines a Ethernet Switch port based IP address assignment.

<b>Parameters included</b>		
<b>Parameter name</b>	<b>Multiplicity</b>	
<a href="#">TcplpDhcpDefaultRouter</a>	0..1	
<a href="#">TcplpDhcpNetmask</a>	0..1	
<a href="#">TcplpDhcpEthIfSwitchRef</a>	0..1	

<b>Parameter Name</b>	<b>TcplpDhcpDefaultRouter</b>
<b>Description</b>	IP address of default router (gateway).



<b>Multiplicity</b>	0..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpDhcpNetmask</b>
<b>Description</b>	Network mask of IPv4 address or address prefix of IPv6 address in CIDR Notation, i.e. decimal value between 0 and 32 (IPv4) or 0 and 128 (IPv6) that describes the number of significant bits defining the network number or prefix of an IP address.
<b>Multiplicity</b>	0..1
<b>Type</b>	INTEGER
<b>Range</b>	<=128 >=0
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpDhcpEthIfSwitchRef</b>
<b>Description</b>	Reference to EthIfSwitch representation.
<b>Multiplicity</b>	0..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.8.1.12. TcpIpDhcpAddressAssignment

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">TcIpDhcpAddressLowerBound</a>	1..1
<a href="#">TcIpDhcpAddressUpperBound</a>	1..1
<a href="#">TcIpDhcpSwitchPortRef</a>	0..1
<b>Parameter Name</b>	<b>TcIpDhcpAddressLowerBound</b>
<b>Description</b>	The lower bound IP address which shall be assigned.



<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>TcpIpDhcpAddressUpperBound</b>
<b>Description</b>	The upper bound IP address which shall be assigned.
<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>TcpIpDhcpSwitchPortRef</b>
<b>Description</b>	Reference to Ethernet Switch port.
<b>Multiplicity</b>	0..1
<b>Type</b>	SYMBOLIC-NAME-REFERENCE
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.8.1.13. TcplpIpConfig

Containers included		
Container name	Multiplicity	Description
<a href="#">TcplpIpFragmentationConfig</a>	1..1	Specifies the configuration parameters of IPv4 & IPv6 packet fragmentation/reassembly.
<a href="#">TcplpIpV4Config</a>	0..1	Specifies the configuration parameters of the IPv4 (Internet Protocol version 4) sub-module.
<a href="#">TcplpIpV6Config</a>	0..1	Specifies the configuration parameters of the IPv6 (Internet Protocol version 6) sub-module.

### 5.8.1.14. TcplpIpFragmentationConfig

Parameters included	
Parameter name	Multiplicity



#### Parameters included

<a href="#">TcpllpFragMemReserved</a>	0..1
<a href="#">TcpllpFragmentationRxEnabled</a>	1..1
<a href="#">TcpllpReassemblyTimeout</a>	1..1
<a href="#">TcpllpReassemblyBufferCount</a>	1..1
<a href="#">TcpllpReassemblyBufferSize</a>	1..1
<a href="#">TcpllpFragmentationTxEnabled</a>	1..1
<a href="#">TcpllpTxFragmentBufferCount</a>	1..1
<a href="#">TcpllpTxFragmentBufferSize</a>	1..1
<a href="#">TcpllpTxFragmentSegmentCount</a>	1..1

<b>Parameter Name</b>	<b>TcpllpFragMemReserved</b>	
<b>Description</b>	Size of internal IpFrag (fragmentation and reassembly) data in units of bytes (static memory allocation) - Memory required by post-build configuration must be smaller than this constant.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>TcpllpFragmentationRxEnabled</b>	
<b>Description</b>	Enables or disables support for reassembling of incoming datagrams that are fragmented according to IETF RFC 815 (IP Datagram Reassembly Algorithms).	
	<ul style="list-style-type: none"> <li>▶ true: IP Datagram Reassembly enabled</li> <li>▶ false: IP Datagram Reassembly 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>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>TcpllpReassemblyTimeout</b>	



<b>Description</b>	Time after which an incomplete datagram gets discarded.
	RFC1122 (from 1989) suggests a value between 60 and 120 seconds.
	A large value can quickly lead to reassembly buffer exhaustion if fragments are lost.
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	60.0
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIplpReassemblyBufferCount</b>
<b>Description</b>	Number of fragmented IP datagrams that can be reassembled in parallel.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	2
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIplpReassemblyBufferSize</b>
<b>Description</b>	Size of each reassembly buffer.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	1500
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIplpFragmentationTxEnabled</b>
<b>Description</b>	Enables or disables support for fragmenting outgoing datagrams according to IETF RFC 791 / RFC 2460  Available choices: <ul style="list-style-type: none"><li>▶ OFF: IP Datagram splitting disabled.</li><li>▶ OUTOFORDER: The header fragment with the checksum will be transmitted last to avoid buffering.</li></ul>



	▶ INORDER: All data will be buffered in Ethernet transmit buffers, so the first fragment with the header and the checksum can be transmitted first.
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	OFF
<b>Range</b>	OFF OUTOFORDER INORDER
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

Parameter Name	TcpIpTxFragmentBufferCount
<b>Description</b>	For in-order transmission: TcpIpTxEnabled = INORDER  Maximum number of transmit buffers. Number of fragmented IP datagrams that can be sent in parallel.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	2
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

Parameter Name	TcpIpTxFragmentBufferSize
<b>Description</b>	For in-order transmission: TcpIpTxEnabled = INORDER  Maximum size of a transmitted packet.  INORDER fragmentation does not allocate memory for the data, but instead stores the data in Ethernet buffers. The maximum number of Ethernet buffers per packet is configured in TcpIpTxFragmentSegmentCount. Multiplying that with the ethernet MTU size is the virtual buffer size, which is the limit for fragmented INORDER transmissions and must be configured here.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	1500
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild



<b>Origin</b>	Elektrobit Automotive GmbH
<b>Parameter Name</b>	<b>TcpIpTxFragmentSegmentCount</b>
<b>Description</b>	For in-order transmission: TcpIpFragmentationTxEnabled = INORDER  Maximum number of transmit Ethernet buffers (fragments) per IP datagram & socket  Twelve bytes of data will be reserved per fragment and buffer to store the Ethernet buffer handles.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	5
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.8.1.15. TcpIpV4Config

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcIpArpConfig</a>	0..1	Specifies the configuration parameters of the ARP (Address Resolution Protocol) sub-module.
<a href="#">TcIpAutoIpConfig</a>	0..1	Specifies the configuration parameters of the Auto-IP (automatic private IP addressing) sub-module.
<a href="#">TcIpDhcpConfig</a>	0..1	Specifies the configuration parameters of the DHCPv4.
<a href="#">TcIpIcmpConfig</a>	0..1	Specifies the configuration parameters of the ICMP (Internet Control Message Protocol) sub-module.
<a href="#">TcIpV4ArpPacketFilter</a>	0..1	This container is a subcontainer of TcIpIpConfig and specifies the configuration parameters for the ARP packet filter.
<a href="#">TcIpIpFragmentationConfig</a>	0..n	DISABLED - TcIp/TcIpIpConfig/TcIpIpFragmentationConfig is used instead for both IPv4 & IPv6!  Specifies the configuration parameters of IPv4 packet fragmentation/reassembly.

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>



#### Parameters included

<a href="#">TcpllpV4StaticAddrDefenseMechanism</a>	1..1
--	------

Parameter Name	TcpllpV4StaticAddrDefenseMechanism
<b>Description</b>	<p>Specifies the performed defense mechanism for TCPIP_STATIC assignment method if an address conflict is detected.</p> <ul style="list-style-type: none"> <li>▶ TCPIP_DISABLE: Address Conflict Detection is not performed.</li> <li>▶ TCPIP_ABANDON_ADDR: If an address conflict is detected the address is ceased immediately.</li> <li>▶ TCPIP_DEFEND_ADDR: If an address conflict is detected an ARP announcement is transmitted. If another conflict is received within DEFEND_INTERVAL seconds the address is ceased.</li> <li>▶ TCPIP_DEFEND_INF_ADDR: If an address conflict is detected an ARP announcement is transmitted. If another conflict is received within DEFEND_INTERVAL seconds no ARP announcement is transmitted.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	TCPIP_DISABLE
<b>Range</b>	TCPIP_DISABLE TCPIP_ABANDON_ADDR TCPIP_DEFEND_ADDR TCPIP_DEFEND_INF_ADDR
<b>Configuration class</b>	VariantPostBuild: <input type="text"/> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

#### 5.8.1.16. TcplpArpConfig

#### Parameters included

Parameter name	Multiplicity
<a href="#">TcplpArpNumGratuitousARPPonStartup</a>	1..1
<a href="#">TcplpArpPacketQueueEnabled</a>	1..1
<a href="#">TcplpArpRequestTimeout</a>	1..1
<a href="#">TcplpArpTableEntryTimeout</a>	1..1

**Parameters included**

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

<b>Parameter Name</b>	<b>TcplpArpNumGratuitousARPonStartup</b>	
<b>Description</b>	Specifies the number of gratuitous ARP replies which shall be sent on assignment of a new IP address.	
<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>TcplpArpPacketQueueEnabled</b>	
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>	
	Enables (TRUE) or disables (FALSE) support of the ARP Packet Queue according to IETF RFC 1122, section 2.3.2.2.	
<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>TcplpArpRequestTimeout</b>	
<b>Description</b>	Defines how long Tcplp waits for an ARP response in seconds until the entry is removed from the cache and another ARP Request is transmitted	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	2	
<b>Range</b>	<=255 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild



<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

<b>Parameter Name</b>	<b>TcplpArpTableEntryTimeout</b>	
<b>Description</b>	Timeout in seconds after which an unused ARP entry is removed. The allowed range for this parameter could be either between 1 and 65535 seconds or Infinity. If set to Infinity no timeout occurs.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	60	
<b>Range</b>	=Infinity <=65535 >=1	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>TcplpArpTableSizeMax</b>	
<b>Description</b>	Maximum number of entries in the ARP table.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	5	
<b>Range</b>	<=65535 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.8.1.17. TcplpAutolpConfig

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">TcplpAutolpInitTimeout</a>	1..1
<a href="#">TcplpV4AutolpAddrDefenseMechanism</a>	1..1
<a href="#">TcplpAutolpv4EntriesMax</a>	1..1
<b>Parameter Name</b>	<b>TcplpAutolpInitTimeout</b>



<b>Description</b>	The time in seconds Auto-IP waits at startup, before beginning with ARP probing. This delay is used to give DHCP time to acquire a lease in case a DHCP server is present.
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	60
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

Parameter Name	TcpIpIpv4AutolpAddrDefenseMechanism
<b>Description</b>	Specifies the performed defense mechanism for TCPIP_LINKLOCAL/TCPIP_LINKLOCAL_DOIP assignment method if an address conflict is detected. <ul style="list-style-type: none"> <li>▶ TCPIP_DISABLE: Address Conflict Detection is not performed.</li> <li>▶ TCPIP_ABANDON_ADDR: If an address conflict is detected the address is ceased immediately.</li> <li>▶ TCPIP_DEFEND_ADDR: If an address conflict is detected an ARP announcement is transmitted. If another conflict is received within DEFEND_INTERVAL seconds the address is ceased.</li> <li>▶ TCPIP_DEFEND_INF_ADDR: If an address conflict is detected an ARP announcement is transmitted. If another conflict is received within DEFEND_INTERVAL seconds no ARP announcement is transmitted.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	TCPIP_DEFEND_ADDR
<b>Range</b>	TCPIP_DISABLE TCPIP_ABANDON_ADDR TCPIP_DEFEND_ADDR TCPIP_DEFEND_INF_ADDR
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

Parameter Name	TcpIpAutolpv4EntriesMax
<b>Description</b>	This parameter defines the maximum number of interface which are allowed to have an entry in TcplpLocalAddr assigned with TcplpAssignmentMethod set to TCPIP_LINKLOCAL/TCPIP_LINKLOCAL_DOIP (Ipv4).



	<p>This parameter has impact on the size of the PostBuild RAM.</p> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	1
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.8.1.18. TcplpDhcpConfig

Containers included		
Container name	Multiplicity	Description
<a href="#">TcplpDhcpConfigurableOptions</a>	0..n	

Parameters included	
Parameter name	Multiplicity
<a href="#">TcplpDhcplpv4EntriesMax</a>	1..1
<a href="#">TcplpDhcpInitDelay</a>	1..1
<a href="#">TcplpIpv4DhcpAddrDefenseMechanism</a>	1..1
<a href="#">TcplpDhcpFQDNOptionEnabled</a>	1..1
<a href="#">TcplpDhcplpv4DomainNameMaxSize</a>	1..1
<a href="#">TcplpDhcpConfigurableOptionsEnabled</a>	1..1
<a href="#">TcplpDhcpConfigurableOptionsEntriesMax</a>	1..1
<a href="#">TcplpDhcpConfigurableOptionsDataSizeMax</a>	1..1
<a href="#">TcplpDhcpArpProbingEnabled</a>	1..1
<a href="#">TcplpDhcpArpProbingType</a>	1..1

<b>Parameter Name</b>	<b>TcplpDhcplpv4EntriesMax</b>
<b>Description</b>	This parameter defines the maximum number of interface which are allowed to have an entry in TcplpLocalAddr assigned with TcplpAssignmentMethod set to TCPIP_DHCP (Ipv4).



	<p>This parameter has impact on the size of the PostBuild RAM.</p> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..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>TcpIpDhcpInitDelay</b>
<b>Description</b>	The time in seconds DHCP waits at startup, before beginning with the transmission of the DHCPDISCOVER message.
<b>Multiplicity</b>	1..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>TcpIpV4DhcpAddrDefenseMechanism</b>
<b>Description</b>	<p>Specifies the performed defense mechanism for TCPIP_DHCP assignment method if an address conflict is detected.</p> <ul style="list-style-type: none"> <li>▶ TCPIP_DISABLE: Address Conflict Detection is not performed.</li> <li>▶ TCPIP_ABANDON_ADDR: If an address conflict is detected the address is ceased immediately.</li> <li>▶ TCPIP_DEFEND_ADDR: If an address conflict is detected an ARP announcement is transmitted. If another conflict is received within DEFEND_INTERVAL seconds the address is ceased.</li> <li>▶ TCPIP_DEFEND_INF_ADDR: If an address conflict is detected an ARP announcement is transmitted. If another conflict is received within DEFEND_INTERVAL seconds no ARP announcement is transmitted.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	TCPIP_DISABLE



<b>Range</b>	TCPIP_DISABLE TCPIP_ABANDON_ADDR TCPIP_DEFEND_ADDR TCPIP_DEFEND_INF_ADDR
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpDhcpFQDNOptionEnabled</b>
<b>Description</b>	Enables (TRUE) or disables (FALSE) the support of the Fully Qualified Domain Name Option for Dynamic Host Configuration Protocol for IPv4 as defined in IETF RFC 4702 (The Dynamic Host Configuration Protocol for IPv4 (DHCPv4)).  <ul style="list-style-type: none"> <li>▶ true: Enables the transmission of the hostname option.</li> <li>▶ false: Disables the transmission of the hostname option.</li> </ul>
<b>Optimization Effect:</b>	<ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Disabling this parameter reduces the RAM consumption of the module.</li> <li>▶ <b>Execution time reduction (code):</b> Disabling this parameter reduces the execution time of the module code.</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>TcpIpDhcpIpv4DomainNameMaxSize</b>
<b>Description</b>	This parameter defines the maximum size of the DHCPv4 Domain Name.  This parameter has impact on the size of the PostBuild RAM.
<b>Optimization Effect:</b>	<ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER



<b>Default value</b>	5
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpDhcpConfigurableOptionsEnabled</b>
<b>Description</b>	Enables (TRUE) or disables (FALSE) Dhcp options  <b>Optimization Effect:</b> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> Disabling this parameter reduces the ROM consumption of the module configuration.</li> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</li> <li>▶ <b>RAM reduction:</b> Disabling this parameter reduces the RAM consumption of the module.</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>TcpIpDhcpConfigurableOptionsEntriesMax</b>
<b>Description</b>	Maximum number of Dhcp option entries in PostBuild configuration.  This parameter has impact on the size of the PostBuild RAM.  <b>Optimization Effect:</b> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> Selecting a small value for this parameter reduces the ROM consumption of the module configuration.</li> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpDhcpConfigurableOptionsDataSizeMax</b>
<b>Description</b>	Maximum data size of all Dhcp option entries in PostBuild configuration.



	<p>This parameter has impact on the size of the PostBuild RAM.</p> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> Selecting a small value for this parameter reduces the ROM consumption of the module configuration.</li> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	TcpIpDhcpArpProbingEnabled
<b>Description</b>	Enable DhcpV4 Arp probing (Duplicate Address Detection according to RFC2131 and RFC5227).
<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>	TcpIpDhcpArpProbingType
<b>Description</b>	<p>Sets the probing type for Dhcp</p> <p>Available choices:</p> <ul style="list-style-type: none"> <li>▶ PROBING_DEFAULT: Probing is done according to the default probing specifications.</li> <li>▶ PROBING_DOIP: Probing is done according to the specifications for TCPIP_LINKLOCAL_DOIP.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	PROBING_DEFAULT
<b>Range</b>	PROBING_DEFAULT PROBING_DOIP
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild

<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

### 5.8.1.19. TcplpDhcpConfigurableOptions

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">TcplpDhcpOptionCode</a>	1..1
<a href="#">TcplpDhcpOptionMaxLength</a>	1..1
<a href="#">TcplpDhcpOptionTransmit</a>	1..1
<b>Parameter Name</b>	<b>TcplpDhcpOptionCode</b>
<b>Description</b>	Code of the Dhcp option
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH
<b>Parameter Name</b>	<b>TcplpDhcpOptionMaxLength</b>
<b>Description</b>	Memory which shall be reserved for this Dhcp option
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Range</b>	<=255 >=1
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH
<b>Parameter Name</b>	<b>TcplpDhcpOptionTransmit</b>
<b>Description</b>	Indicates if the option shall be transmitted in every DHCP message or just stored when it is received from the server.  ▶ true: option is transmitted ▶ false: option is stored when received
<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	

### 5.8.1.20. TcplplcmpConfig

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcplplcmpMsgHandler</a>	0..1	This container is a subcontainer of cplplcmpConfig and specifies the configuration parameters for the ICMP message handler.

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

<b>Parameter Name</b>	<b>TcplplcmpEchoReplyEnabled</b>
<b>Description</b>	Enables or disables transmission of ICMP echo reply message in case of a ICMP echo reception. <ul style="list-style-type: none"> <li>▶ true: ICMP echo reply enabled</li> <li>▶ false: ICMP echo reply disabled</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	true
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcplplcmpTtl</b>
<b>Description</b>	Default Time-to-live value of outgoing ICMP packets.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	255
<b>Range</b>	<=255 >=1



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.8.1.21. TcplplcmplMsgHandler

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

<b>Parameter Name</b>	<b>TcplplcmplMsgHandlerHeaderFileName</b>
<b>Description</b>	This parameter specifies the name of the header file containing the definition of the ICMP message handler function..
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcplplcmplMsgHandlerName</b>
<b>Description</b>	This parameter defines the name of the ICMP message handler function <Up_-IcmpMsgHandler>.  Syntax: void Up_IcmpMsgHandler ( Tcplp_LocalAddrIdType LocalAddrId, const Tcplp_SockAddrType* RemoteAddrPtr, uint8 Ttl, uint8 Type, uint8 Code, uint16 DataLength, const uint8* DataPtr )
<b>Multiplicity</b>	1..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	AUTOSAR_ECUC

### 5.8.1.22. TcplplpV4ArpPacketFilter

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>



#### Parameters included

<a href="#">TcpllpV4ArpPacketFilterFileName</a>	1..1
<a href="#">TcpllpV4ArpPacketFilterName</a>	1..1

<b>Parameter Name</b>	<b>TcpllpV4ArpPacketFilterFileName</b>
<b>Description</b>	This parameter specifies the name of the header file containing the definition of the ARP packet filter function.
<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>TcpllpV4ArpPacketFilterName</b>
<b>Description</b>	<p>This parameter defines the name of the ARP packet filter function. The function must follow the signature:</p> <pre>&lt;function&gt;boolean &lt;function_name&gt;(Tcplp_LocalAddrIdType localAddrId, uint8 *dataPtr, uint16 lenByte)&lt;/function&gt;</pre> <p>The function shall return TRUE, if the sender of the datagram shall be explicitly added to the ARP table.</p> <p>The function shall return FALSE, if the sender of the datagram shall not be explicitly added to the ARP table.</p>
<b>Multiplicity</b>	1..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

#### 5.8.1.23. TcpllpFragmentationConfig

#### Parameters included

Parameter name	Multiplicity
<a href="#">TcpllpFragmentationRxEnabled</a>	1..1
<a href="#">TcpllpNumFragments</a>	0..1
<a href="#">TcpllpNumReassDgrams</a>	0..1

**Parameters included**

<a href="#">TcpllpReassTimeout</a>	0..1
------------------------------------	------

<b>Parameter Name</b>	<b>TcpllpFragmentationRxEnabled</b>
<b>Description</b>	Enables or disables support for reassembling of incoming datagrams that are fragmented according to IETF RFC 815 (IP Datagram Reassembly Algorithms). <ul style="list-style-type: none"> <li>▶ true: IP Datagram Reassembly enabled</li> <li>▶ false: IP Datagram Reassembly 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_ECUC

<b>Parameter Name</b>	<b>TcpllpNumFragments</b>
<b>Description</b>	Maximum number of IP fragments per datagram.
<b>Multiplicity</b>	0..1
<b>Type</b>	INTEGER
<b>Default value</b>	10
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpllpNumReassDgrams</b>
<b>Description</b>	Number of fragmented IP datagrams that can be reassembled in parallel.
<b>Multiplicity</b>	0..1
<b>Type</b>	INTEGER
<b>Default value</b>	10
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpllpReassTimeout</b>
<b>Description</b>	Time after which an incomplete datagram gets discarded. RFC1122 suggests a value between 60 and 120 seconds.



<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	60.0	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.8.1.24. TcplIpv6Config

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcplpDhcpV6Config</a>	0..1	Specifies the configuration parameters of the DHCPv6.
<a href="#">TcplpIcmpV6Config</a>	1..1	Specifies the configuration parameters of the ICMPv6 (Internet Control Message Protocol for IPv6) sub-module.
<a href="#">TcplpIpV6FragmentationConfig</a>	0..n	<p>DISABLED - Tcplp/TcplpConfig/TcplpIpFragmentationConfig is used instead for both IPv4 &amp; IPv6!</p> <p>Specifies the configuration parameters of IPv6 packet fragmentation/reassembly.</p>
<a href="#">TcplpNdpConfig</a>	0..1	Specifies the configuration parameters of the Neighbor Discovery Protocol for IPv6.
<a href="#">TcplpIpV6ConfigExtHeaderFilter</a>	0..n	This container describes the white list for the filtering of IPv6 extension headers, i.e. frames containing IPv6 extension headers not listed here shall be silently dropped.

### 5.8.1.25. TcplpDhcpV6Config

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcplpDhcpV6ConfigurableOptions</a>	0..n	

<b>Parameters included</b>		
<b>Parameter name</b>	<b>Multiplicity</b>	
<a href="#">TcplpDhcpV6CnfDelayMax</a>	1..1	
<a href="#">TcplpDhcpV6CnfDelayMin</a>	1..1	



#### Parameters included

<a href="#">TcplpDhcpV6InfDelayMax</a>	1..1
<a href="#">TcplpDhcpV6InfDelayMin</a>	1..1
<a href="#">TcplpDhcpV6SolDelayMax</a>	1..1
<a href="#">TcplpDhcpV6SolDelayMin</a>	1..1
<a href="#">TcplpDhcplpv6EntriesMax</a>	1..1
<a href="#">TcplpDhcplpv6ServerDuidMaxSize</a>	1..1
<a href="#">TcplpDhcplpv6DomainNameMaxSize</a>	1..1
<a href="#">TcplpDhcpV6FQDNOptionEnabled</a>	1..1
<a href="#">TcplpDhcpV6ConfigurableOptionsEnabled</a>	1..1
<a href="#">TcplpDhcpV6ConfigurableOptionsEntriesMax</a>	1..1
<a href="#">TcplpDhcpV6ConfigurableOptionsDataSizeMax</a>	1..1
<a href="#">TcplpDhcpV6SlaacDadEnabled</a>	1..1

Parameter Name	TcplpDhcpV6CnfDelayMax
<b>Description</b>	Maximum delay (s) before sending the first Confirm message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.  The functionality related to this parameter is not supported by the current implementation. Confirm messages are currently not transmitted by the Tcplp.
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	1.0
<b>Range</b>	<=100.0 >=0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

Parameter Name	TcplpDhcpV6CnfDelayMin
<b>Description</b>	Minimum delay (s) before the first Confirm message will be sent.  The functionality related to this parameter is not supported by the current implementation. Confirm messages are currently not transmitted by the Tcplp.
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT



<b>Default value</b>	0.0
<b>Range</b>	<=100.0 >=0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpDhcpV6InfDelayMax</b>
<b>Description</b>	Maximum delay (s) before sending the first Information Request message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.  The functionality related to this parameter is not supported by the current implementation. Information Request messages are currently not transmitted by the TcpIp.
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	1.0
<b>Range</b>	<=100.0 >=0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpDhcpV6InfDelayMin</b>
<b>Description</b>	Minimum delay (s) before the first Information Request message will be sent.  The functionality related to this parameter is not supported by the current implementation. Information Request messages are currently not transmitted by the TcpIp.
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.0
<b>Range</b>	<=100.0 >=0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC



<b>Parameter Name</b>	<b>TcpIpDhcpV6SolDelayMax</b>
<b>Description</b>	Maximum delay (s) before sending the first Solicit message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.  The functionality related to this parameter is not supported by the current implementation. Timeouts specified in the IETF Rfc 3315 are used instead. Initial retransmission time (IRT) = 1s, Maximum retransmission time (MRT) = 120s
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	1.0
<b>Range</b>	<=100.0 >=0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpDhcpV6SolDelayMin</b>
<b>Description</b>	Minimum delay (s) before the first Solicit message will be sent.  The functionality related to this parameter is not supported by the current implementation. Timeouts specified in the IETF Rfc 3315 are used instead. Initial retransmission time (IRT) = 1s, Maximum retransmission time (MRT) = 120s.
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.0
<b>Range</b>	<=100.0 >=0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpDhcplpv6EntriesMax</b>
<b>Description</b>	This parameter defines the maximum number of interface which are allowed to have an entry in TcpIpLocalAddr assigned with TcpIpAssignmentMethod set to TCPIP_DHCPv6 (Ipv6).  This parameter has impact on the size of the PostBuild RAM.

**Optimization Effect:**



	<ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..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>TcpIpDhcIpv6ServerDuidMaxSize</b>
<b>Description</b>	<p>This parameter defines the maximum supported size of the DHCPv6 Server DUID Default value is based on DUID_LLT</p> <p>This parameter has impact on the size of the PostBuild RAM.</p> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	12
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpDhcIpv6DomainNameMaxSize</b>
<b>Description</b>	<p>This parameter defines the maximum size of the DHCPv6 Domain Name.</p> <p>This parameter has impact on the size of the PostBuild RAM.</p> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	5
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH



<b>Parameter Name</b>	<b>TcpIpDhcpV6FQDNOptionEnabled</b>	
<b>Description</b>	<p>Enables (TRUE) or disables (FALSE) the support of the Fully Qualified Domain Name Option for Dynamic Host Configuration Protocol for IPv6 as defined in IETF RFC 4704 (The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option).</p> <ul style="list-style-type: none"> <li>▶ true: Enables the transmission of the hostname option.</li> <li>▶ false: Disables the transmission of the hostname option.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Disabling this parameter reduces the RAM consumption of the module.</li> <li>▶ <b>Execution time reduction (code):</b> Disabling this parameter reduces the execution time of the module code.</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>TcpIpDhcpV6ConfigurableOptionsEnabled</b>	
<b>Description</b>	Enables (TRUE) or disables (FALSE) Dhcp options	
<b>Optimization Effect:</b>	<ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> Disabling this parameter reduces the ROM consumption of the module configuration.</li> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</li> <li>▶ <b>RAM reduction:</b> Disabling this parameter reduces the RAM consumption of the module.</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>TcpIpDhcpV6ConfigurableOptionsEntriesMax</b>	



<b>Description</b>	Maximum number of DhcpV6 option entries in PostBuild configuration.  This parameter has impact on the size of the PostBuild RAM.
	<b>Optimization Effect:</b> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> Selecting a small value for this parameter reduces the ROM consumption of the module configuration.</li> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpDhcpV6ConfigurableOptionsDataSizeMax</b>	
<b>Description</b>	Maximum data size of all DhcpV6 option entries in PostBuild configuration.  This parameter has impact on the size of the PostBuild RAM.	
	<b>Optimization Effect:</b> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> Selecting a small value for this parameter reduces the ROM consumption of the module configuration.</li> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>TcpIpDhcpV6SlaacDadEnabled</b>	
<b>Description</b>	Enable DhcpV6 Duplicate Address Detection (DAD) according to RFC4862 and RFC3315.	
<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
---------------	----------------------------

### 5.8.1.26. TcplpDhcpV6ConfigurableOptions

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">TcplpDhcpV6OptionCode</a>	1..1
<a href="#">TcplpDhcpV6OptionMaxLength</a>	1..1
<a href="#">TcplpDhcpV6OptionTransmit</a>	1..1

<b>Parameter Name</b>	<b>TcplpDhcpV6OptionCode</b>	
<b>Description</b>	Code of the DhcpV6 option	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<a href="#">VariantPostBuild:</a>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>TcplpDhcpV6OptionMaxLength</b>	
<b>Description</b>	Memory which shall be reserved for this DhcpV6 option	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Range</b>	<=65535 >=1	
<b>Configuration class</b>	<a href="#">VariantPostBuild:</a>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>TcplpDhcpV6OptionTransmit</b>	
<b>Description</b>	Indicates if the option shall be transmitted in every DHCP message or just stored when it is received from the server.  ▶ true: option is transmitted ▶ false: option is stored when received	
<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

### 5.8.1.27. TcplplcmpV6Config

Containers included		
Container name	Multiplicity	Description
<a href="#">TcplplcmpV6MsgHandler</a>	0..1	This container is a subcontainer of TcplplcmpConfig and specifies the configuration parameters for the ICMPv6 message handler.

Parameters included	
Parameter name	Multiplicity
<a href="#">TcplplcmpV6EchoReplyEnabled</a>	1..1
<a href="#">TcplplcmpV6EchoReplyToMulticastEnabled</a>	1..1
<a href="#">TcplplcmpV6HopLimit</a>	1..1
<a href="#">TcplplcmpV6MsgDestinationUnreachableEnabled</a>	1..1
<a href="#">TcplplcmpV6MsgParameterProblemEnabled</a>	1..1
<a href="#">TcplplcmpV6EchoReplyAvoidFragmentation</a>	1..1

<b>Parameter Name</b>	<b>TcplplcmpV6EchoReplyEnabled</b>
<b>Description</b>	If enabled, the stack will respond to incoming ICMPv6 Echo Requests (Pings).
<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>TcplplcmpV6EchoReplyToMulticastEnabled</b>
<b>Description</b>	If enabled, the stack will respond to incoming ICMPv6 Echo Requests (Pings) addressed to multicast address.
<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>TcpIplcmpV6HopLimit</b>	
<b>Description</b>	Default Hop-Limit value of outgoing ICMPv6 packets.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	255	
<b>Range</b>	<=255	>=0
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>TcpIplcmpV6MsgDestinationUnreachableEnabled</b>	
<b>Description</b>	Dis/Enables transmission of Destination Unreachable Messages	
<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>TcpIplcmpV6MsgParameterProblemEnabled</b>	
<b>Description</b>	If enabled an ICMPv6 parameter problem message will be sent if a received packet has been dropped due to unknown options or headers that are found in the packet.	
<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>TcpIplcmpV6EchoReplyAvoidFragmentation</b>	
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>	



	If enabled, the stack will respond only to incoming ICMPv6 Echo Requests (Pings) that fit the MTU of the respective interface, i.e. can be transmitted without IPv6 fragmentation. Only relevant if TcplIcmpV6EchoReplyEnabled is enabled.
<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

### 5.8.1.28. TcplIcmpV6MsgHandler

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">TcplIcmpV6MsgHandlerHeaderFileName</a>	1..1
<a href="#">TcplIcmpV6MsgHandlerName</a>	1..1

<b>Parameter Name</b>	<b>TcplIcmpV6MsgHandlerHeaderFileName</b>
<b>Description</b>	This parameter specifies the name of the header file containing the definition of the ICMPv6 message handler function.
<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>TcplIcmpV6MsgHandlerName</b>
<b>Description</b>	This parameter defines the name of the ICMP message handler function <User_IcmpMsgHandler>.  Syntax: void Up_IcmpMsgHandler ( Tcplp_LocalAddrIdType LocalAddrId, const Tcplp_SockAddrType* RemoteAddrPtr, uint8 HopLimit, uint8 Type, uint8 Code, uint16 DataLength, const uint8* DataPtr )
<b>Multiplicity</b>	1..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild



Origin	AUTOSAR_ECUC
--------	--------------

### 5.8.1.29. TcpllpV6FragmentationConfig

Parameters included	
Parameter name	Multiplicity
<a href="#">TcpllpV6ReassemblyBufferCount</a>	1..1
<a href="#">TcpllpV6ReassemblyBufferSize</a>	0..1
<a href="#">TcpllpV6ReassemblySegmentCount</a>	0..1
<a href="#">TcpllpV6ReassemblyTimeout</a>	0..1
<a href="#">TcpllpV6TxFragmentBufferCount</a>	1..1
<a href="#">TcpllpV6TxFragmentBufferSize</a>	0..1

Parameter Name	<b>TcpllpV6ReassemblyBufferCount</b>	
Description	Number of buffers that can be used for fragment reassembly. In case of a reassembly error or if not all fragments are received in time this buffer will be blocked until the specified "Fragment Reassembly Timeout" has been exceeded.	
Multiplicity	1..1	
Type	INTEGER	
Default value	2	
Range	<=255 >=0	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>TcpllpV6ReassemblyBufferSize</b>	
Description	[RFC2460 5. Packet Size Issues]	
Multiplicity	0..1	
Type	INTEGER	
Default value	1500	
Range	<=65535 >=1500	
Configuration class	<b>PreCompile:</b>	VariantPostBuild



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

<b>Parameter Name</b>	<b>TcpIplpV6ReassemblySegmentCount</b>	
<b>Description</b>	Specifies the maximum number of consecutive data segments that can be managed in each reassembly buffer. If all fragments are received in order, only one segment will be needed.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	5	
<b>Range</b>	<=255 >=1	
<b>Configuration class</b>	PreCompile:	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>TcpIplpV6ReassemblyTimeout</b>	
<b>Description</b>	[RFC2460 4.5 Fragment Header]	
<b>Multiplicity</b>	0..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	60.0	
<b>Range</b>	<=100.0 >=0.001	
<b>Configuration class</b>	PreCompile:	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>TcpIplpV6TxFragmentBufferCount</b>	
<b>Description</b>	These buffers will be used if the IPv6 receives packets from the upper layer that do not fit into the MTU and thus must be fragmented.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	2	
<b>Range</b>	<=1000 >=1	
<b>Configuration class</b>	VariantPostBuild:	VariantPostBuild



<b>Origin</b>	AUTOSAR_ECUC
<b>Parameter Name</b>	<b>TcpIpV6TxFragmentBufferSize</b>
<b>Description</b>	Size of each fragment tx buffer in bytes
<b>Multiplicity</b>	0..1
<b>Type</b>	INTEGER
<b>Default value</b>	1500
<b>Range</b>	<=65535 >=1500
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.8.1.30. TcpIpNdpConfig

Containers included		
Container name	Multiplicity	Description
<a href="#">TcIpNdpArNudConfig</a>	0..1	Specifies the configuration parameters for NDP Address Resolution and Neighbor Unreachability Detection.
<a href="#">TcIpNdpPrefixRouterDiscoveryConfig</a>	0..1	Specifies the configuration parameters for NDP Prefix and Router Discovery.
<a href="#">TcIpNdpSlaacConfig</a>	0..1	Specifies the configuration parameters for StateLess Address AutoConfiguration.

### 5.8.1.31. TcIpNdpArNudConfig

Parameters included	
Parameter name	Multiplicity
<a href="#">TcIpNdpDefaultReachableTime</a>	1..1
<a href="#">TcIpNdpDefaultRetransTimer</a>	1..1
<a href="#">TcIpNdpDelayFirstProbeTime</a>	1..1
<a href="#">TcIpNdpMaxNeighborCacheSize</a>	1..1
<a href="#">TcIpNdpMaxRandomFactor</a>	1..1
<a href="#">TcIpNdpMinRandomFactor</a>	1..1



#### Parameters included

<a href="#">TcplpNdpNeighborUnreachabilityDetectionEnabled</a>	1..1
<a href="#">TcplpNdpNumMulticastSolicitations</a>	1..1
<a href="#">TcplpNdpNumUnicastSolicitations</a>	1..1
<a href="#">TcplpNdpPacketQueueEnabled</a>	1..1
<a href="#">TcplpNdpRandomReachableTimeEnabled</a>	1..1
<a href="#">TcplpNdpDefensiveProcessing</a>	1..1

Parameter Name	<b>TcplpNdpDefaultReachableTime</b>	
<b>Description</b>	Configuration of the ReachableTime (s) specified in [RFC4861 6.3.2. Host Variables].	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	30.0	
<b>Range</b>	<=120.0 >=0.0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

Parameter Name	<b>TcplpNdpDefaultRetransTimer</b>	
<b>Description</b>	Configures the default value (s) for the RetransTimer variable specified in [RFC4861 6.3.2. Host Variables].	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	1.0	
<b>Range</b>	<=60.0 >=0.0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

Parameter Name	<b>TcplpNdpDelayFirstProbeTime</b>	
<b>Description</b>	Delay before sending the first NUD probe in (s).	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	



<b>Default value</b>	5.0
<b>Range</b>	<=60.0 >=0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpNdpMaxNeighborCacheSize</b>
<b>Description</b>	Maximum number of entries in the neighbor cache.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	5
<b>Range</b>	<=254 >=1
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpNdpMaxRandomFactor</b>
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Maximum random factor used for randomization
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	15
<b>Range</b>	<=100 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpNdpMinRandomFactor</b>
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Minimum random factor used for randomization"
<b>Multiplicity</b>	1..1



<b>Type</b>	INTEGER
<b>Default value</b>	5
<b>Range</b>	<=100 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpNdpNeighborUnreachabilityDetectionEnabled</b>
<b>Description</b>	Note: Neighbor Unreachability Detection is always turned on per default and cannot be turned off.  Neighbor Unreachability Detection is used to remove unused entries from the neighbor cache. This feature is a basic feature of NDP and should be turned on.
<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>TcpIpNdpNumMulticastSolicitations</b>
<b>Description</b>	Maximum number of multicast solicitations that will be sent when performing address resolution.
<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>TcpIpNdpNumUnicastSolicitations</b>
<b>Description</b>	Maximum number of unicast solicitations that will be sent when performing Neighbor Unreachability Detection.
<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>TcpIpNdpPacketQueueEnabled</b>
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Enables (TRUE) or disables (FALSE) support of a NDP Packet Queue according to IETF RFC 4861, section 7.2.2.
<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>TcpIpNdpRandomReachableTimeEnabled</b>
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  If enabled the value of ReachableTime will be multiplied with a random value between MIN_RANDOM_FACTOR and MAX_RANDOM_FACTOR in order to prevent multiple nodes from transmitting at exactly the same time
<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>TcpIpNdpDefensiveProcessing</b>
<b>Description</b>	If TcpIpNdpDefensiveProcessing is set to TRUE, the NDP shall silently discard all received Neighbor Advertisements that have not been requested by a previously transmitted Neighbor Solicitation and skip the update of the Neighbor Cache upon processing received Neighbor Solicitations.



<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	

### 5.8.1.32. TcplpNdpPrefixRouterDiscoveryConfig

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcplpNdpPrefixList</a>	0..1	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Specifies a list of prefixes to be treated as "on-link" according to IETF RFC 4861 Section 5.1.</p>

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">TcplpNdpDefaultRouterListSize</a>	1..1
<a href="#">TcplpNdpDestinationCacheSize</a>	1..1
<a href="#">TcplpNdpDestinationCacheEnabled</a>	1..1
<a href="#">TcplpNdpDynamicHopLimitEnabled</a>	1..1
<a href="#">TcplpNdpDynamicMtuEnabled</a>	1..1
<a href="#">TcplpNdpDynamicReachableTimeEnabled</a>	1..1
<a href="#">TcplpNdpDynamicRetransTimeEnabled</a>	1..1
<a href="#">TcplpNdpMaxRtrSolicitationDelay</a>	1..1
<a href="#">TcplpNdpMaxRtrSolicitations</a>	1..1
<a href="#">TcplpNdpPrefixListSize</a>	1..1
<a href="#">TcplpNdpRndRtrSolicitationDelayEnabled</a>	1..1
<a href="#">TcplpNdpRtrSolicitationInterval</a>	1..1

<b>Parameter Name</b>	<b>TcplpNdpDefaultRouterListSize</b>
<b>Description</b>	value="Maximum number of default router entries."
<b>Multiplicity</b>	1..1



Type	INTEGER	
Default value	2	
Range	<=254 >=2	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>TcpIpNdpDestinationCacheSize</b>	
Description	Maximum number of entries in the destination cache.	
Multiplicity	1..1	
Type	INTEGER	
Default value	5	
Range	<=254 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	<b>TcpIpNdpDestinationCacheEnabled</b>	
Description	If enabled the destination cache shall be used.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>TcpIpNdpDynamicHopLimitEnabled</b>	
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i>	
	If enabled the default hop limit may be reconfigured based on received Router Advertisements.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>TcpIpNdpDynamicMtuEnabled</b>	
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>	
	Allow dynamic reconfiguration of link MTU via Router Advertisements.	
<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>TcpIpNdpDynamicReachableTimeEnabled</b>	
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>	
	If enabled the default Reachable Time value may be reconfigured based on received Router Advertisements.	
<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>TcpIpNdpDynamicRetransTimeEnabled</b>	
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>	
	If enabled the default Retransmit Timer value may be reconfigured based on received Router Advertisements.	
<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>TcpIpNdpMaxRtrSolicitationDelay</b>	
<b>Description</b>	Maximum delay before the first Router Solicitation will be sent after interface initialization in (s).	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	1.0	
<b>Range</b>	<=60.0 >=0.001	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>TcpIpNdpMaxRtrSolicitations</b>	
<b>Description</b>	Maximum number of Router Solicitations that will be sent before the first Router Advertisement has been received.	
<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>TcpIpNdpPrefixListSize</b>	
<b>Description</b>	Maximum number of entries in the on-link prefix list.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	5	
<b>Range</b>	<=254 >=1	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>TcpIpNdpRndRtrSolicitationDelayEnabled</b>	
-----------------------	---	--



<b>Description</b>	If enabled the first router solicitation will be delayed randomly from [0...MAX_-RTR_SOLICITATION_DELAY]. Otherwise the first router solicitation will be sent after exactly MAX_RTR_SOLICITATION_DELAY milliseconds. prefix list.	
<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>TcplpNdpRtrSolicitationInterval</b>	
<b>Description</b>	Interval between consecutive Router Solicitations in (s).	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	4.0	
<b>Range</b>	<=60.0 >=0.001	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.8.1.33. TcplpNdpPrefixList

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcplpNdpPrefixListEntry</a>	1..n	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Single entry in the prefix list</p>

### 5.8.1.34. TcplpNdpPrefixListEntry

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



#### Parameters included

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

Parameter Name	TcplpNdpPrefixListEntryPrefixAddress
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  The prefix of an IP address. This prefix can be used for on-link determination
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

Parameter Name	TcplpNdpPrefixListEntryPrefixLength
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  The number of leading bits in the Prefix that are valid
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Range</b>	<=128 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.8.1.35. TcplpNdpSlaacConfig

Parameters included	
Parameter name	Multiplicity
<a href="#">TcplpNdpSlaacDadNumberOfTransmissions</a>	1..1
<a href="#">TcplpNdpSlaacDadRetransmissionDelay</a>	1..1
<a href="#">TcplpNdpSlaacDelayEnabled</a>	1..1
<a href="#">TcplpNdpSlaacOptimisticDadEnabled</a>	1..1

Parameter Name	TcplpNdpSlaacDadNumberOfTransmissions
----------------	---------------------------------------



<b>Description</b>	"Number of Neighbor Solicitations that have to be unanswered in order to set an autoconfigurated address to PREFERRED (usable) state.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	1	
<b>Range</b>	<=254 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>TcpIpNdpSlaacDadRetransmissionDelay</b>	
<b>Description</b>	Sets the maximum value for the address configuration delay (s)	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	1.0	
<b>Range</b>	<=10.0 >=0.0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>TcpIpNdpSlaacDelayEnabled</b>	
<b>Description</b>	If enabled transmission of the first DAD Neighbor Solicitation will be delayed by a random value from [0...MAX_DAD_DELAY].	
<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>TcpIpNdpSlaacOptimisticDadEnabled</b>	
<b>Description</b>	Enable Optimistic Duplicate Address Detection (DAD) according to RFC4429.	
<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

### 5.8.1.36. TcplIpv6ConfigExtHeaderFilter

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">TcplIpv6ConfigExtHeaderFilterEntry</a>	1..n
<b>Parameter Name</b>	
<b>Description</b>	IPv6 Extension Header type allowed by this filter
<b>Multiplicity</b>	1..n
<b>Type</b>	INTEGER
<b>Range</b>	<=255 >=0
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.8.1.37. TcplpLocalAddr

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcplpAddrAssignment</a>	1..n	This container is a subcontainer of TcplpLocalAddr and specifies the assignment policy for the IP address.
<a href="#">TcplpStaticIpAddressConfig</a>	0..1	This container is a subcontainer of TcplpLocalAddr and specifies a static IP address including directly related parameters.

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



#### Parameters included

<a href="#">TcplpCtrlRef</a>	1..1
<a href="#">TcplpLocalAddrIpv6ExtHeaderFilterRef</a>	0..1

<b>Parameter Name</b>	<b>TcplpAddrId</b>
<b>Description</b>	IP address table identifier assigned by TCP/IP stack.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Range</b>	<=65535 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcplpAddressType</b>
<b>Description</b>	Address type <ul style="list-style-type: none"> <li>▶ TCPIP_UNICAST: Unicast address</li> <li>▶ TCPIP_MULTICAST: Multicast address</li> <li>▶ TCPIP_ANYCAST: Anycast address</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	TCPIP_UNICAST
<b>Range</b>	TCPIP_MULTICAST TCPIP_UNICAST TCPIP_ANYCAST
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcplpDomainType</b>
<b>Description</b>	Address family <ul style="list-style-type: none"> <li>▶ TCPIP_AF_INET: IPv4 address</li> <li>▶ TCPIP_AF_INET6: IPv6 address</li> </ul>



<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	TCPIP_AF_INET
<b>Range</b>	TCPIP_AF_INET TCPIP_AF_INET6
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	TcpIpCtrlRef
<b>Description</b>	Reference to a TcpIpCtrl specifying the Ethif Controller where the IP address shall be assigned and DEM errors that shall be reported in case of an error on this controller.
<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>	TcpIpLocalAddrIPv6ExtHeaderFilterRef
<b>Description</b>	Reference to a set of IPv6 Extension Headers which are allowed for this local IPv6 address
<b>Multiplicity</b>	0..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.8.1.38. TcpIpAddrAssignment

Parameters included	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">TcIpAssignmentLifetime</a>	0..1
<a href="#">TcIpAssignmentMethod</a>	1..1
<a href="#">TcIpAssignmentPriority</a>	1..1
<a href="#">TcIpAssignmentTrigger</a>	1..1

**Parameters included**

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

Parameter Name	TcplpAssignmentLifetime
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Defines the lifetime of a dynamically fetched IP address.
<b>Multiplicity</b>	0..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	TCPIP_FORGET
<b>Range</b>	TCPIP_FORGET TCPIP_STORE
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU

Parameter Name	TcplpAssignmentMethod
<b>Description</b>	<p>Address type</p> <ul style="list-style-type: none"> <li>▶ TCPIP_STATIC: <ul style="list-style-type: none"> <li>▶ TCPIP_AF_INET: Assignment of static IPv4 address</li> <li>▶ TCPIP_AF_INET6: Assignment of static IPv6 address</li> </ul> </li> <li>▶ TCPIP_LINKLOCAL: <ul style="list-style-type: none"> <li>▶ TCPIP_AF_INET: Assignment of IPv4 Link Local address; timing according to IETF RFC 3927 - Dynamic Configuration of IPv4 Link-Local Addresses</li> <li>▶ TCPIP_AF_INET6: Assignment of IPv6 Link Local address</li> </ul> </li> <li>▶ TCPIP_LINKLOCAL_DOIP: <ul style="list-style-type: none"> <li>▶ TCPIP_AF_INET: Assignment of IPv4 Link Local address; timing according to ISO 13400 (DoIP)</li> <li>▶ TCPIP_AF_INET6: Not applicable</li> </ul> </li> <li>▶ TCPIP_DHCP: <ul style="list-style-type: none"> <li>▶ TCPIP_AF_INET: Assignment of IPv4 address through a DHCPv4 server</li> <li>▶ TCPIP_AF_INET6: Assignment of IPv6 address through a DHCPv6 server</li> </ul> </li> </ul>



	<ul style="list-style-type: none"> <li>▶ TCPIP_IPV6_ROUTER:           <ul style="list-style-type: none"> <li>▶ TCPIP_AF_INET: Not applicable</li> <li>▶ TCPIP_AF_INET6: Assignment of IPv6 address through Router Advertisements</li> </ul> </li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	TCPIP_STATIC
<b>Range</b>	TCPIP_DHCP TCPIP_IPV6_ROUTER TCPIP_LINKLOCAL TCPIP_LINKLOCAL_DOIP TCPIP_STATIC
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpAssignmentPriority</b>
<b>Description</b>	Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	1
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpAssignmentTrigger</b>
<b>Description</b>	Trigger of address assignment.
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	TCPIP_AUTOMATIC
<b>Range</b>	TCPIP_AUTOMATIC TCPIP_MANUAL
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild



<b>Origin</b>	AUTOSAR_ECUC
<b>Parameter Name</b>	<b>TcplpUseSimpleDhcpClient</b>
<b>Description</b>	<p><i>This parameter is only relevant if TcplpAssignmentMethod equals TCPIP_DHCP.</i></p> <p>It specifies if the IP address shall be assigned through the exchange of 2 messages with the DHCP server. See parameter TcplpDhcpSimpleClientEnabled.</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

### 5.8.1.39. TcplpStaticIpConfig

Parameters included	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">TcplpDefaultRouter</a>	0..1
<a href="#">TcplpNetmask</a>	0..1
<a href="#">TcplpStaticIpAddress</a>	1..1

<b>Parameter Name</b>	<b>TcplpDefaultRouter</b>
<b>Description</b>	<p>IP address of default router (gateway) with following notation dependent on parameter TcplpDomainType:</p> <ul style="list-style-type: none"> <li>▶ TCPIP_AF_INET: Four octets separated by character "." in decimal format</li> <li>▶ TCPIP_AF_INET6: 8 block to 16bit separated by character ":" in hexadecimal format</li> </ul>
<b>Multiplicity</b>	0..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcplpNetmask</b>
<b>Description</b>	Network mask of IPv4 address or address prefix of IPv6 address in CIDR Notation, i.e. decimal value between 0 and 32 (IPv4) or 0 and 128 (IPv6) that de-



	scribes the number of significant bits defining the network number or prefix of an IP address.
<b>Multiplicity</b>	0..1
<b>Type</b>	INTEGER
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcplpStaticIpAddress</b>	
<b>Description</b>	Static IP Address with following notation dependent on parameter TcplpDomain-Type:  ▶ TCPIP_AF_INET: Four octets separated by character "." in decimal format ▶ TCPIP_AF_INET6: 8 block to 16bit separated by character ":" in hexadecimal format	
<b>Multiplicity</b>	1..1	
<b>Type</b>	STRING	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.8.1.40. TcplpNvmBlock

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

<b>Parameter Name</b>	<b>TcplpNvmBlockDescriptorRef</b>	
<b>Description</b>	Reference to the Nvm block description in the Nvm module configuration.  Nvm block must have NvMSelectBlockForReadAll and NvMSelectBlockForWriteAll enabled, NvMRamBlockDataAddress set to Tcplp_Memory_NvM_Ip_Memory and "Tcplp.h" added as one of the entries in NvMIUserHeader list.	
<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>TcplpNvmBlockSize</b>	
<b>Description</b>	This parameter defines the size of internal Tcplp cache dedicated to interaction with NvM [in units of four bytes].	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	1	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.8.1.41. TcplpPhysAddrConfig

Containers included		
Container name	Multiplicity	Description
<a href="#">TcplpPhysAddrChgHandler</a>	0..1	This container is a subcontainer of TcplpPhysAddrConfig

### 5.8.1.42. TcplpPhysAddrChgHandler

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

<b>Parameter Name</b>	<b>TcplpPhysAddrChgHandlerHeaderFileName</b>	
<b>Description</b>	This parameter specifies the name of the header file containing the definition of the physical address change handler function.	
<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>TcplpPhysAddrChgHandlerName</b>	
<b>Description</b>	This parameter defines the name of the physical address change function <Up>_PhysAddrTableChg.	



	Syntax: void Up_PhysAddrTableChg ( uint8 CtrlIdx, const Tcplp_SockAddrType* IpAddrPtr, const uint8* PhysAddrPtr, boolean valid )
<b>Multiplicity</b>	1..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.8.1.43. TcplpSocketOwnerConfig

Containers included		
Container name	Multiplicity	Description
<a href="#">TcplpSocketOwner</a>	1..n	This container is a subcontainer of TcplpSocketOwnerConfig and specifies an upper layer of Tcplp that uses the socket API.

### 5.8.1.44. TcplpSocketOwner

Parameters included	
Parameter name	Multiplicity
<a href="#">TcplpSocketOwnerHeaderFileName</a>	0..1
<a href="#">TcplpSocketOwnerCopyTxDataName</a>	0..1
<a href="#">TcplpSocketOwnerLocallpAddrAssignmentChgName</a>	0..1
<a href="#">TcplpSocketOwnerRxIndicationName</a>	0..1
<a href="#">TcplpSocketOwnerTcpAcceptedName</a>	0..1
<a href="#">TcplpSocketOwnerTcpConnectedName</a>	0..1
<a href="#">TcplpSocketOwnerTxConfirmationName</a>	0..1
<a href="#">TcplpSocketOwnerTcplpEventName</a>	0..1
<a href="#">TcplpSocketOwnerUpperLayerType</a>	1..1

<b>Parameter Name</b>	<b>TcplpSocketOwnerHeaderFileName</b>
<b>Description</b>	This parameter specifies the name of the header file containing the definition of the TcplpSocketOwner module functions. The header file name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD.
<b>Multiplicity</b>	0..1



<b>Type</b>	STRING
<b>Configuration class</b>	<b>Link:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpSocketOwnerCopyTxDataName</b>
<b>Description</b>	This parameter defines the name of the <Up_CopyTxData> function of the TcpIpSocketOwner module. The function name shall only be configurable if TcpIpSocketOwnerUpperLayerType is set to CDD. API.
<b>Multiplicity</b>	0..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>Link:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpSocketOwnerLocallpAddrAssignmentChgName</b>
<b>Description</b>	This parameter defines the name of the <Up_LocallpAddrAssignmentChg> function of the TcpIpSocketOwner module. The function name shall only be configurable if TcpIpSocketOwnerUpperLayerType is set to CDD.
<b>Multiplicity</b>	0..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>Link:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpSocketOwnerRxIndicationName</b>
<b>Description</b>	This parameter defines the name of the <Up_RxIndication> function of the TcpIpSocketOwner module. The function name shall only be configurable if TcpIpSocketOwnerUpperLayerType is set to CDD.
<b>Multiplicity</b>	0..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>Link:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpSocketOwnerTcpAcceptedName</b>
<b>Description</b>	This parameter defines the name of the <Up_TcpAccepted> function of the TcpIpSocketOwner module. The function name shall only be configurable if TcpIpSocketOwnerUpperLayerType is set to CDD.
<b>Multiplicity</b>	0..1



<b>Type</b>	STRING
<b>Configuration class</b>	<b>Link:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpSocketOwnerTcpConnectedName</b>
<b>Description</b>	This parameter defines the name of the <Up_TcpConnected> function of the TcpIpSocketOwner module. The function name shall only be configurable if TcpIpSocketOwnerUpperLayerType is set to CDD.
<b>Multiplicity</b>	0..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>Link:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpSocketOwnerTxConfirmationName</b>
<b>Description</b>	This parameter defines the name of the <Up_TxConfirmation> function of the TcpIpSocketOwner module. The function name shall only be configurable if TcpIpSocketOwnerUpperLayerType is set to CDD.
<b>Multiplicity</b>	0..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>Link:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpSocketOwnerTcplpEventName</b>
<b>Description</b>	This parameter defines the name of the <Up_TcplpEvent> function of the TcpIpSocketOwner module. The function name shall only be configurable if TcpIpSocketOwnerUpperLayerType is set to CDD.
<b>Multiplicity</b>	0..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>Link:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpSocketOwnerUpperLayerType</b>
<b>Description</b>	This parameter specifies the type of the upper layer module.
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION



<b>Range</b>	CDD SOAD
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

### 5.8.1.45. TcplpTcpConfig

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcplpTcpConfigOptionFilter</a>	0..n	This container describes the white list for the filtering of TCP options, i.e. segments containing TCP options not listed here shall be silently dropped
<a href="#">TcplpTcpUnpredictableSequenceNumbers</a>	0..1	Container for unpredictable sequence number parameters. If enabled, unpredictables sequence numbers are generated when upper layer wants to establish a new active connection. Values are generated from remote Ip address, remote port, local address id, local port, domain and a secret key.
<a href="#">TcplpTcpSynCookies</a>	0..1	Container for SYN cookies parameters

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">TcplpDelayedAckTimeout</a>	1..1
<a href="#">TcplpTcpCongestionAvoidanceEnabled</a>	1..1
<a href="#">TcplpTcpFastRecoveryEnabled</a>	1..1
<a href="#">TcplpTcpFastRetransmitEnabled</a>	1..1
<a href="#">TcplpTcpFinWait2Timeout</a>	1..1
<a href="#">TcplpTcpKeepAliveEnabled</a>	1..1
<a href="#">TcplpTcpKeepAliveDefault</a>	1..1
<a href="#">TcplpTcpKeepAliveInterval</a>	1..1
<a href="#">TcplpTcpKeepAliveProbesMax</a>	1..1
<a href="#">TcplpTcpKeepAliveTime</a>	1..1
<a href="#">TcplpTcpMaxRtx</a>	0..1
<a href="#">TcplpTcpMsl</a>	1..1
<a href="#">TcplpTcpNagleEnabled</a>	1..1



#### Parameters included

<a href="#">TcplpTcpReceiveWindowMax</a>	1..1
<a href="#">TcplpTcpRetransmissionTimeout</a>	0..1
<a href="#">TcplpTcpSlowStartEnabled</a>	1..1
<a href="#">TcplpTcpSynMaxRtx</a>	1..1
<a href="#">TcplpTcpSynReceivedTimeout</a>	1..1
<a href="#">TcplpTcpTtl</a>	1..1
<a href="#">TcplpTcpOptionFilterEnabled</a>	1..1
<a href="#">TcplpTcpDupAckTransmitTimeoutSeqNum</a>	1..1
<a href="#">TcplpTcpDupAckTransmitTimeoutAckNum</a>	1..1
<a href="#">TcplpTcpOutOfOrderReceptionEnabled</a>	1..1
<a href="#">TcplpTcpOutOfOrderReceptionBufferCount</a>	1..1
<a href="#">TcplpTcpOutOfOrderReceptionBufferSize</a>	1..1
<a href="#">TcplpTcpOutOfOrderReceptionHoleListSize</a>	1..1
<a href="#">TcplpTcpCopyWindowCheckEnabled</a>	1..1
<a href="#">TcplpTcpTimeWaitRstIgnoreReceptionEnabled</a>	1..1

<b>Parameter Name</b>	<b>TcplpDelayedAckTimeout</b>	
<b>Description</b>	The maximal time an acknowledgment is delayed for transmission in seconds.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.1	
<b>Range</b>	>=0 <=0.5	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>TcplpTcpCongestionAvoidanceEnabled</b>	
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>	
	Enables or disables support of TCP congestion avoidance algorithm according to IETF RFC 5681.	
	<ul style="list-style-type: none"> <li>▶ true: Congestion avoidance algorithm enabled</li> </ul>	



	▶ false: Congestion avoidance algorithm 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>TcpIpTcpFastRecoveryEnabled</b>
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Enables or disables support of TCP Fast Recovery according to IETF RFC 5681.</p> <ul style="list-style-type: none"> <li>▶ true: Fast recovery algorithm enabled</li> <li>▶ false: Fast recovery algorithm 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>TcpIpTcpFastRetransmitEnabled</b>
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Enables or disables support of TCP Fast Retransmission according to IETF RFC 5681.</p> <ul style="list-style-type: none"> <li>▶ true: Fast retransmissions algorithm enabled</li> <li>▶ false: Fast retransmissions algorithm 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>TcpIpTcpFinWait2Timeout</b>
-----------------------	--------------------------------



<b>Description</b>	Timeout in [s] to receive a FIN from the remote node (after this node has initiated connection termination), i.e. maximum time waiting in FINWAIT-2 for a connection termination request from the remote TCP.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	20.0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>TcpIpTcpKeepAliveEnabled</b>	
<b>Description</b>	Enables or disables TCP Keep Alive Probes according to IETF RFC 1122 chapter 4.2.3.6	
	<ul style="list-style-type: none"> <li>▶ true: Keep alive probes enabled</li> <li>▶ false: Keep alive probes 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>TcpIpTcpKeepAliveDefault</b>	
<b>Description</b>	Enables or disables TCP Keep Alive Probes for all sockets by default.	
<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>TcpIpTcpKeepAliveInterval</b>	
<b>Description</b>	Specifies the interval in [s] between subsequent keepalive probes.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	7200	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild



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

<b>Parameter Name</b>	<b>TcpIpTcpKeepAliveProbesMax</b>	
<b>Description</b>	Maximum number of times that a TCP Keep Alive is retransmitted before the connection is closed.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	10	
<b>Range</b>	>=1 <=255	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>TcpIpTcpKeepAliveTime</b>	
<b>Description</b>	Specifies the time in [s] between the last data packet sent (simple ACKs are not considered data) and the first keepalive probe. Note: Setting this configuration parameter to a value smaller or equal to the value of TcpIpMainFunctionPeriod results in the transmission of keep alive probes within every MainFunction cycle.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	7200	
<b>Range</b>	>=0.1 <=65535	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>TcpIpTcpMaxRtx</b>	
<b>Description</b>	Maximum number of times that a TCP segment is retransmitted before the TCP connection is closed. This parameter is only valid if TcpIpTcpRetransmission-Timeout is configured.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	12	
<b>Range</b>	<=255	



	>=0
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpTcpMsl</b>
<b>Description</b>	Maximum segment lifetime in [s]. (Note: TIME-WAIT = 2 x TcpIpTcpMsl - to ensure that the remote node received the acknowledgment to its connection termination request.)
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	120.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpTcpNagleEnabled</b>
<b>Description</b>	Enables or disables support of Nagle's algorithm according to IETF RFC 896. If enabled the Nagle's algorithm is activated per default for all TCP sockets, but can be deactivated via Tcplp_ChangeParameter() API. <ul style="list-style-type: none"> <li>▶ true: Nagle's algorithm enabled</li> <li>▶ false: Nagle's algorithm 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>TcplpTcpReceiveWindowMax</b>
<b>Description</b>	Default value of maximum receive window in bytes.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	4380
<b>Range</b>	>=1 <=65535
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild



<b>Origin</b>	AUTOSAR_ECUC
<b>Parameter Name</b>	<b>TcpIpTcpRetransmissionTimeout</b>
<b>Description</b>	Timeout in [s] before an unacknowledged TCP segment is sent again.
<b>Multiplicity</b>	0..1
<b>Type</b>	FLOAT
<b>Default value</b>	3.5
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC
<b>Parameter Name</b>	<b>TcpIpTcpSlowStartEnabled</b>
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Enables or disables support of TCP slow start algorithm according to IETF RFC 5681.</p> <ul style="list-style-type: none"> <li>▶ true: Slow start algorithm enabled</li> <li>▶ false: Slow start algorithm 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>TcpIpTcpSynMaxRtx</b>
<b>Description</b>	Maximum number of times that a TCP SYN is retransmitted. Note: SYN will be retried after TcplpTcpRetransmissionTimeout. The connection will be dropped if no matching connection request has been received after the last TCP SYN has been sent and TcplpTcpRetransmissionTimeout has been expired.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	4
<b>Range</b>	<=255 >=0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild



<b>Origin</b>	AUTOSAR_ECUC
<b>Parameter Name</b>	<b>TcpIpTcpSynReceivedTimeout</b>
<b>Description</b>	Timeout in [s] to complete a remotely initiated TCP connection establishment, i.e. maximum time waiting in SYN_RECEIVED for a confirming connection request acknowledgment after having both received and sent a connection request.
<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>TcpIpTcpTtl</b>
<b>Description</b>	Default Time-to-live value of outgoing TCP packets
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	255
<b>Range</b>	>=1 <=255
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC
<b>Parameter Name</b>	<b>TcpIpTcpOptionFilterEnabled</b>
<b>Description</b>	Enables (TRUE) or disables (FALSE) filtering of TCP options
<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>TcpIpTcpDupAckTransmitTimeoutSeqNum</b>
<b>Description</b>	Timeout in [s] to rate-limit the duplicate Acknowledgement that are sent in response to incoming packets with out-of-window sequence number.  Value 0 turns off the rate limiting.



<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0
<b>Range</b>	<=1.0 >=0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpTcpDupAckTransmitTimeoutAckNum</b>
<b>Description</b>	Timeout in [s] to rate-limit the duplicate Acknowledgement that are sent in response to incoming packets with out-of-window acknowledgement number.  Value 0 turns off the rate limiting.
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0
<b>Range</b>	<=1.0 >=0.0
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpTcpOutOfOrderReceptionEnabled</b>
<b>Description</b>	Enables (TRUE) or disables (FALSE) support for out-of-order reception of TCP segments.
<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>TcpIpTcpOutOfOrderReceptionBufferCount</b>
<b>Description</b>	Sets the number of buffers that can be used for the reception of out-of-order TCP segments, i.e. the maximum number of sockets that can use out-of-order reception simultaneously.
<b>Multiplicity</b>	1..1



<b>Type</b>	INTEGER
<b>Default value</b>	1
<b>Range</b>	>=1 <=32764
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpTcpOutOfOrderReceptionBufferSize</b>
<b>Description</b>	Sets the size in octets per buffer that can be used for the reception of out-of-order TCP segments.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	1024
<b>Range</b>	>=1 <=65535
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpTcpOutOfOrderReceptionHoleListSize</b>
<b>Description</b>	Sets the hole list size per buffer that can be used for the reception of out-of-order TCP segments, i.e. the maximum number of holes that can be used per buffer.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	8
<b>Range</b>	>=1 <=254
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpTcpCopyWindowCheckEnabled</b>
<b>Description</b>	Determines the behavior of TcpIp_TcpTransmit when it is called with the parameter forceRetrieve set to FALSE. When copying data from the upper layer to be transmitted over TCP, TcpIp selects the amount of data to be copied based on:



	<ul style="list-style-type: none"> <li>▶ true: the minimum of the window size of the remote host and the available buffer signaled by the upper layer.</li> <li>▶ false: the available buffer signaled by the upper layer.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	true
<b>Configuration class</b>	<b>TcpIpTcpConfigOptionFilterEntry</b> : VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpTcpTimeWaitRstIgnoreReceptionEnabled</b>	
<b>Description</b>	Enables (TRUE) or disables (FALSE) support for ignoring the reception of TCP RST segments.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>TcpIpTcpConfigOptionFilterEntry</b> : VariantPostBuild	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.8.1.46. **TcpIpTcpConfigOptionFilter**

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

<b>Parameter Name</b>	<b>TcpIpTcpConfigOptionFilterEntry</b>	
<b>Description</b>	TCP option kind allowed by this filter	
<b>Multiplicity</b>	1..n	
<b>Type</b>	INTEGER	
<b>Range</b>	>=0 <=255	
<b>Configuration class</b>	<b>TcpIpTcpConfigOptionFilter</b> : PostBuild	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	



<b>Parameter Name</b>	<b>TcplpTcpConfigOptionFilterId</b>	
<b>Description</b>	Identification of the TCP option filter	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Range</b>	>=0 <=255	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.8.1.47. TcplpTcpUnpredictableSeqNumbers

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

<b>Parameter Name</b>	<b>TcplpTcpUnpredictableSeqNumbersKeyGenerateJobId</b>	
<b>Description</b>	Reference to Csm_MacGenerate Job which is used to generate unpredictable value for initial TCP sequence number secret.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>TcplpTcpUnpredictableSeqNumbersKeyResetTime</b>	
<b>Description</b>	Time period for resetting the key used for generating unpredictable TCP initial sequence numbers in seconds.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Range</b>	<=1000 >=1	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	



### 5.8.1.48. TcplpTcpSynCookies

Parameters included	
Parameter name	Multiplicity
<a href="#">TcplpTcpSynCookiesKey1GenerateJobId</a>	1..1
<a href="#">TcplpTcpSynCookiesKey2GenerateJobId</a>	1..1
<a href="#">TcplpTcpSynCookiesTimeResetKeys</a>	1..1
<a href="#">TcplpTcpSynCookiesAcceptAckOverflowTime</a>	1..1
<a href="#">TcplpTcpSynCookiesTimebaseRef</a>	1..1

Parameter Name	<b>TcplpTcpSynCookiesKey1GenerateJobId</b>	
Description	Reference to Csm_MacGenerate Job which is used to generate pseudo random value for SYN cookie secret 1.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>TcplpTcpSynCookiesKey2GenerateJobId</b>	
Description	Reference to Csm_MacGenerate Job which is used to generate pseudo random value for SYN cookie secret 2.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	<b>TcplpTcpSynCookiesTimeResetKeys</b>	
Description	Time period in seconds [s] after which keys for SYN cookies must be reset.	
Multiplicity	1..1	
Type	FLOAT	
Default value	1.0	
Range	<=1000.0 >=0.0	
Configuration class	<b>VariantPostBuild:</b>	VariantPostBuild



<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

<b>Parameter Name</b>	<b>TcpIpTcpSynCookiesAcceptAckOverflowTime</b>	
<b>Description</b>	Time (in seconds [s]) after SYN reception overflow (no free entry for creating new SYN-RECEIVED socket) after which ACK can be accepted.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	1	
<b>Range</b>	<=128 >=0	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>TcpIpTcpSynCookiesTimebaseRef</b>	
<b>Description</b>	A reference to an StbM time base to generate syn cookie. This time base has to return the absolute calendar time (i.e. seconds passed since 1970-01-01 00:00) when StbM_GetCurrentTime is called.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

#### 5.8.1.49. TcplpUdpConfig

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

<b>Parameter Name</b>	<b>TcplpUdpTtl</b>	
<b>Description</b>	Default Time-to-live value of outgoing UDP packets.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	255	



<b>Range</b>	<=255 >=1
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU

<b>Parameter Name</b>	<b>TcpIpUdpMayReFragment</b>
<b>Description</b>	Allows or disallows re-fragmentation along the path for outgoing IP packets by enabling / disabling the corresponding IP header flag.  Enabling this globally will severely limit the allowed bandwidth for Ipv4, as the 16 bit ID counter has to be incremented for every packet on a channel (source, dest, ID, protocol), if (re-)fragmentation is possible within any gateway on the path to the destination. The problem is further aggravated by long reassembly timeouts (RFC1122 suggests 60-120 seconds) If the ID is only incremented for fragmented packets, the associated increase of the MTU size from ~1500 to 64kB makes the 16bit counter a much lesser problem for the bandwith.  ▶ true: UDP Datagram may be fragmented ▶ false: UDP Datagram may not be fragmented
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	true
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.8.1.50. TcplpDuplicateAddressDetectionConfig

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

<b>Parameter Name</b>	<b>TcplpDuplicateAddressDetectionCalloutName</b>
<b>Description</b>	This parameter defines the name of the DAD callout function Up_DADAddress-Conflict.
<b>Multiplicity</b>	1..1



Type	FUNCTION-NAME	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplpDuplicateAddressDetectionHeaderFileName	
Description	This parameter specifies the name of the header file containing the definition of the DAD callout function	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

### 5.8.1.51. TcplpRxPolicyCheckIngressHandlerConfig

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

Parameter Name	TcplpRxPolicyCheckIngressHandlerName	
Description	This parameter defines the name of the RxPolicyCheckIngressHandler callout function.	
Multiplicity	1..1	
Type	FUNCTION-NAME	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpRxPolicyCheckIngressHandlerHeaderFileName	
Description	This parameter specifies the header file name containing the definition of RxPolicyCheckIngressHandler callout function.	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	Elektrobit Automotive GmbH
--------	----------------------------

### 5.8.1.52. TcplpRxPolicyCheckRoutingHandlerConfig

Parameters included	
Parameter name	Multiplicity
<a href="#">TcplpRxPolicyCheckRoutingHandlerName</a>	1..1
<a href="#">TcplpRxPolicyCheckRoutingHandlerHeaderFileName</a>	1..1
Parameter Name	<b>TcplpRxPolicyCheckRoutingHandlerName</b>
Description	This parameter defines the name of the RxPolicyCheckRoutingHandler callout function.
Multiplicity	1..1
Type	FUNCTION-NAME
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH
Parameter Name	<b>TcplpRxPolicyCheckRoutingHandlerHeaderFileName</b>
Description	This parameter specifies the header file name containing the definition of RxPolicyCheckRoutingHandler callout function.
Multiplicity	1..1
Type	STRING
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

### 5.8.1.53. TcplpRxPacketPostProcessHandlerConfig

Parameters included	
Parameter name	Multiplicity
<a href="#">TcplpRxPacketPostProcessHandlerName</a>	1..1
<a href="#">TcplpRxPacketPostProcessHandlerHeaderFileName</a>	1..1
Parameter Name	<b>TcplpRxPacketPostProcessHandlerName</b>
Description	This parameter defines the name of the RxPacketPostProcessHandler callout function.



<b>Multiplicity</b>	1..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpRxPacketPostProcessHandlerHeaderFileName</b>
<b>Description</b>	This parameter specifies the header file name containing the definition of Rx-PacketPostProcessHandler callout function.
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.8.1.54. TcpIpRxPacketDropHandlerConfig

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

<b>Parameter Name</b>	<b>TcpIpRxPacketDropHandlerName</b>
<b>Description</b>	This parameter defines the name of the RxPacketDropHandler callout function.
<b>Multiplicity</b>	1..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpRxPacketDropHandlerHeaderFileName</b>
<b>Description</b>	This parameter specifies the header file name containing the definition of Rx-PacketDropHandler callout function.
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild



Origin	Elektrobit Automotive GmbH
--------	----------------------------

### 5.8.1.55. TcplpMemoryConfig

Containers included		
Container name	Multiplicity	Description
<a href="#">TcplpMemoryPool</a>	0..255	This container describes a memory pool. A memory pool consists of the memory block size and how many blocks of this size are available to store data for e.g. TCP retransmissions.

### 5.8.1.56. TcplpMemoryPool

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

Parameter Name	TcplpMemoryBlockSize
Description	This parameter defines the size of a memory block in bytes.
Multiplicity	1..1
Type	INTEGER
Configuration class	<a href="#">VariantPostBuild:</a> VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplpMemoryBlockCount
Description	This parameter defines the number of available memory blocks with the size specified by TcplpMemoryBlockSize.
Multiplicity	1..1
Type	INTEGER
Range	<=65535 >=1
Configuration class	<a href="#">VariantPostBuild:</a> VariantPostBuild
Origin	Elektrobit Automotive GmbH



### 5.8.1.57. TcpllpSecConfig

Containers included		
Container name	Multiplicity	Description
<a href="#">TcpllpSecGeneral</a>	1..1	<p>General pre-compile configuration parameters for IpSec</p> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</li> </ul>
<a href="#">TcpllpSecConnections</a>	1..255	This container describes the relationship between a local and a remote host. An entry must be created for every host the ECU wants to communicate with. Transmissions and Receptions will be blocked for hosts which are not listed.
<a href="#">TcpllpSecSecurityAssociationCfg</a>	1..n	This container contains the configuration of the IPsec security associations.
<a href="#">TcpllpSecReportErrorHandler</a>	0..1	This container is a subcontainer of TcpllpSecReportError

### 5.8.1.58. TcpllpSecGeneral

Parameters included	
Parameter name	Multiplicity
<a href="#">TcpllpSecGmacEnable</a>	1..1
<a href="#">TcplpMaxNumIpsecConnections</a>	1..1
<a href="#">TcplpMaxNumIpsecSecurityAssociation</a>	1..1
<a href="#">TcplpMaxNumIpsecSecurityAssociationConfigurations</a>	1..1
<a href="#">TcplpIcvSizeMax</a>	1..1
<a href="#">TcpllpSecRemotePhysAddrCheckEnable</a>	1..1

<b>Parameter Name</b>	<b>TcpllpSecGmacEnable</b>
<b>Description</b>	Enables/Disables AES GMAC. <ul style="list-style-type: none"> <li>▶ true: AES GMAC is enabled.</li> </ul>



	▶ false: AES GMAC is 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>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpMaxNumIPsecConnections</b>
<b>Description</b>	Maximal number of IPSec connections
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpMaxNumIPsecSecurityAssociations</b>
<b>Description</b>	Maximal number of IPSec security associations
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpMaxNumIPsecSecurityAssociationConfigurations</b>
<b>Description</b>	Maximal number of IPSec security association configurations
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpIcvSizeMax</b>
<b>Description</b>	Indicates the maximal size of a generated ICV in bytes. It should be equal to the maximal size of the ICV generated by the Integrity algorithm.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	32



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	
<b>Parameter Name</b>	<b>TcpIpIpSecRemotePhysAddrCheckEnable</b>	
<b>Description</b>	Enables (FALSE) or disables (TRUE) filtering by remote MAC address.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.8.1.59. TcpIpIpSecConnections

Containers included		
Container name	Multiplicity	Description
<a href="#">IpSecRemoteAddrConfig</a>	1..1	This parameter is responsible to indicate the remote address configuration for the IpSec connection
<a href="#">TcpIpIpSecRemotePhysAddr</a>	0..1	Specifies the range of destination MAC Address for filtering.
<a href="#">TcpIpIpSecSecurityPolicies</a>	1..n	The current implementation only support one inbound and one outbound security policy.  This container describes the policies between the two parties. Communication can either be bypassed or secured. Transmissions and receptions which do not match a policy are dropped.

Parameters included	
Parameter name	Multiplicity
<a href="#">TcpIpIpSecConId</a>	1..1
<a href="#">TcpIpIpSecDomainType</a>	1..1
<a href="#">TcpIpIpSecLocalAddrRef</a>	1..1
<a href="#">TcpIpIpSecRemoteAddrType</a>	1..1

<b>Parameter Name</b>	<b>TcpIpIpSecConId</b>
<b>Description</b>	IP secured connection table identifier assigned by TCP/IP stack.
<b>Multiplicity</b>	1..1



Type	INTEGER	
Default value	0	
Range	<=255 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpllpSecDomainType	
Description	Address family <ul style="list-style-type: none"> <li>▶ TCPIP_AF_INET: IPv4 address</li> <li>▶ TCPIP_AF_INET6: IPv6 address</li> </ul>	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	TCPIP_AF_INET	
Range	TCPIP_AF_INET TCPIP_AF_INET6	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpllpSecLocalAddrRef	
Description	Reference to a configured Local IP Address	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpllpSecRemoteAddrType	
Description	Remote Address type <ul style="list-style-type: none"> <li>▶ TCPIP_UNICAST: Unicast address</li> <li>▶ TCPIP_MULTICAST: Multicast address</li> </ul>	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	TCPIP_UNICAST	



<b>Range</b>	TCPIP_MULTICAST
	TCPIP_UNICAST
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.8.1.60. IpSecRemoteAddrConfig

Containers included		
Container name	Multiplicity	Description
<a href="#">SingleAddress</a>	1..1	Configuration for a remote host using a single IP address.
<a href="#">AddressRange</a>	1..1	Configuration for a remote host using a an IP address range.

### 5.8.1.61. SingleAddress

Parameters included	
Parameter name	Multiplicity
<a href="#">TcplplpSecRemoteAddr</a>	1..1
<b>Parameter Name</b>	<b>TcplplpSecRemoteAddr</b>
<b>Description</b>	Remote IP Address with following notation dependent on parameter TcplplpSecSADomainType: <ul style="list-style-type: none"> <li>▶ TCPIP_AF_INET: Four octets separated by character "." in decimal format</li> <li>▶ TCPIP_AF_INET6: 8 block to 16bit separated by character ":" in hexadecimal format</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.8.1.62. AddressRange

Parameters included	
Parameter name	Multiplicity



#### Parameters included

<a href="#">TcpllpSecStartRemoteAddr</a>	1..1
<a href="#">TcpllpSecEndRemoteAddr</a>	1..1

Parameter Name	TcpllpSecStartRemoteAddr
<b>Description</b>	Remote IP Address with following notation dependent on parameter TcpllpSecSADomainType:
	<ul style="list-style-type: none"> <li>▶ TCPIP_AF_INET: Four octets separated by character "." in decimal format</li> <li>▶ TCPIP_AF_INET6: 8 block to 16bit separated by character ":" in hexadecimal format</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

Parameter Name	TcpllpSecEndRemoteAddr
<b>Description</b>	Remote IP Address with following notation dependent on parameter TcpllpSecSADomainType:
	<ul style="list-style-type: none"> <li>▶ TCPIP_AF_INET: Four octets separated by character "." in decimal format</li> <li>▶ TCPIP_AF_INET6: 8 block to 16bit separated by character ":" in hexadecimal format</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	STRING
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

#### 5.8.1.63. TcpllpSecRemotePhysAddr

Parameters included	
Parameter name	Multiplicity
<a href="#">TcpllpSecStartRemotePhysAddr</a>	1..1
<a href="#">TcpllpSecEndRemotePhysAddr</a>	1..1
Parameter Name	TcpllpSecStartRemotePhysAddr



<b>Description</b>	Start range of destination MAC Address	
<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>TcpIpIpSecEndRemotePhysAddr</b>	
<b>Description</b>	End range of destination MAC Address	
<b>Multiplicity</b>	1..1	
<b>Type</b>	STRING	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.8.1.64. TcpIpIpSecSecurityPolicies

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcplpIpSecSecurityRule</a>	2..n	This container describes the ports and protocols for which the policy applies.

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

<b>Parameter Name</b>	<b>TcplpIpSecSecurityPolicyMechanism</b>
<b>Description</b>	Indicates if datagram shall be bypassed or secured <ul style="list-style-type: none"> <li>▶ BYPASSED: Datagram is not protected by IpSec</li> <li>▶ SECURED: Datagram is protected by IpSec</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Range</b>	SECURED BYPASSED



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>TcpIpIpSecSecurityPolicyDirection</b>	
<b>Description</b>	Security policy direction.  ▶ INBOUND: Describes the handling for incoming messages. ▶ OUTBOUND: Describes the handling of outgoing messages.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Range</b>	INBOUND OUTBOUND	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>TcpIpIpSecSecurityPolicySecurityAssociationRef</b>	
<b>Description</b>	Reference to Security Association for this policy. Note: Reference to Security Association is only required when the communication shall be secured. Note: Inbound and Outbound policy must reference the same Security Association	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.8.1.65. TcpIpIpIpSecSecurityRule

<b>Parameters included</b>		
<b>Parameter name</b>		<b>Multiplicity</b>
<a href="#">TcpIpIpSecSecurityPolicyDirection</a>		1..1
<a href="#">TcpIpIpSecSecurityPolicyUpperLayer</a>		1..1
<a href="#">TcpIpIpSecSecurityPolicyStartPort</a>		1..1
<a href="#">TcpIpIpSecSecurityPolicyEndPort</a>		1..1
<b>Parameter Name</b>	<b>TcpIpIpSecSecurityPolicyDirection</b>	



<b>Description</b>	Direction of the rule <ul style="list-style-type: none"> <li>▶ LOCAL:</li> <li>▶ REMOTE:</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Range</b>	LOCAL <hr/> REMOTE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

Parameter Name	TcpIplpSecSecurityPolicyUpperLayer
<b>Description</b>	Upper layer protocol <ul style="list-style-type: none"> <li>▶ ICMP: Internet Control Message Protocol for the Internet Protocol Version 4 and 6.</li> <li>▶ UDP: User Datagram Protocol.</li> <li>▶ TCP: Transport Control Protocol.</li> <li>▶ ANY: An arbitrary protocol.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	ANY
<b>Range</b>	UDP <hr/> TCP <hr/> ICMP <hr/> ANY
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

Parameter Name	TcpIplpSecSecurityPolicyStartPort
<b>Description</b>	Start of the port range for which the policy is valid
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild



<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

<b>Parameter Name</b>	<b>TcpIpIpSecSecurityPolicyEndPort</b>	
<b>Description</b>	End of the port range for which the policy is valid	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.8.1.66. TcpIpIpSecSecurityAssociationCfg

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcplpIpSecSecurityProposal</a>	1..n	This container contains a list of IPsec SAs Proposals.

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">TcplpIpSecSecurityAssociationKeyExchangeMethod</a>	1..1
<a href="#">TcplpIpSecSecurityAssociationInboundSpi</a>	1..1
<a href="#">TcplpIpSecSecurityAssociationOutboundSpi</a>	1..1
<a href="#">TcplpIpSecSecurityAssociationProtocol</a>	1..1

<b>Parameter Name</b>	<b>TcplpIpSecSecurityAssociationKeyExchangeMethod</b>	
<b>Description</b>	Indicates if the security association is either configured manually or dynamically through IKEv2.  ▶ MANUAL: Manual static configuration. ▶ DYNAMIC: Dynamic configuration through IKEv2.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	MANUAL	
<b>Range</b>	MANUAL  DYNAMIC	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild



<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

<b>Parameter Name</b>	<b>TcpIpIpSecSecurityAssociationInboundSpi</b>	
<b>Description</b>	Note: This parameter shall only be configured for Security Association with manual <code>TcpIpIpSecSecurityAssociationKeyExchangeMethod</code> , dynamic ones are selected during runtime.  Indicates the Security Policy Index for inbound Security Association.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>TcpIpIpSecSecurityAssociationOutboundSpi</b>	
<b>Description</b>	Note: This parameter shall only be configured for Security Association with manual <code>TcpIpIpSecSecurityAssociationKeyExchangeMethod</code> , dynamic ones are selected during runtime.  Indicates the Security Policy Index for outbound Security Association.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>TcpIpIpSecSecurityAssociationProtocol</b>	
<b>Description</b>	<i>The current implementation supports AH only.</i>  Defines the used security protocol.  ▶ AH: Authentication Header ▶ ESP: Encapsulating Security Payload	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	AH	
<b>Range</b>	AH  ESP	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild

<b>Origin</b>	Elektrobit Automotive GmbH
---------------	----------------------------

### 5.8.1.67. TcpllpSecSecurityProposal

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcpllpSecSecurityAssociation</a>	1..n	This container contains a list of IPsec SAs which contain references to the cryptographic jobs.

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

<b>Parameter Name</b>	<b>TcpllpSecSecurityAssociationIntegrityAlgorithm</b>	
<b>Description</b>	Supported algorithms for Integrity:	
	<ul style="list-style-type: none"> <li>▶ AES_CMAC_96</li> <li>▶ HMAC_SHA2_256_128</li> <li>▶ AES_GMAC_128</li> <li>▶ AES_GMAC_256</li> </ul> <p><b>Note: If HMAC_SHA2_256_128 is used to secure IPv4 traffic with a Linux host, the respective transform state i.e. security association needs to be configured with the align4 flag. For more information please refer to man ip-xfrm.</b></p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Range</b>	AES_CMAC_96 HMAC_SHA2_256_128 AES_GMAC_128 AES_GMAC_256	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	



<b>Parameter Name</b>	<b>TcpIplpSecSecurityExtendedSequenceNumberEnabled</b>	
<b>Description</b>	<p>Note: This parameter is disabled for manual TcpIplpSecSecurityAssociation-KeyExchangeMethod, for dynamic ones it is enabled.</p> <p>Indicates the support for Extended Sequence Number (ESN) handling.</p>	
<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>TcpIplpSecProposalPriority</b>	
<b>Description</b>	<p>Priority of the security association proposal. If Dynamic key exchange is configured multiple security association proposals can be exchanged during the negotiation. security association with lower priority value (index) will be preferred during the negotiation If manual key exchange is configured only one proposal is allowed</p>	
<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>	Elektrobit Automotive GmbH	

### 5.8.1.68. TcpIplpSecSecurityAssociation

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

<b>Parameter Name</b>	<b>TcpIplpSecSecurityAssociationInboundIntegrityJobRef</b>	
<b>Description</b>	<p>Reference to a job with key which is used for:</p> <p>Verification of MACs.</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	
<b>Parameter Name</b>	<b>TcpIplpSecSecurityAssociationOutboundIntegrityJobRef</b>	
<b>Description</b>	Reference to a job with key which is used for:  Generation of MACs.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	REFERENCE	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

### 5.8.1.69. TcpIplpSecReportErrorHandler

Parameters included		
<b>Parameter name</b>		<b>Multiplicity</b>
<a href="#">TcpIplpSecReportErrorHandlerFileName</a>		1..1
<a href="#">TcpIplpSecReportErrorHandlerName</a>		1..1
<b>Parameter Name</b>	<b>TcpIplpSecReportErrorHandlerFileName</b>	
<b>Description</b>	This parameter specifies the name of the header file containing the definition of the report error handler function.	
<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>TcpIplpSecReportErrorHandlerName</b>	
<b>Description</b>	This parameter defines the name of the physical address change function <Up>_TcpIplpSecReportError. This function is called when the following errors occur:  - no secured or bypassed policy could be found in the SPD (frame is dropped)  - AH ICV validation failed  Syntax: void Up_TcpIplpSecReportError ( uint8 ctrlIdx, P2CONST(TcpIplp_SockAddrType, AUTOMATIC, TCPIP_APPL_DATA) localSockAddPtr,	



	P2CONST(Tcplp_SockAddrType, AUTOMATIC, TCPIP_APPL_DATA) remoteSockAddPtr, uint8 errorCode )
<b>Multiplicity</b>	1..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

### 5.8.1.70. TcplpDefensiveProgramming

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">TcplpDefProgEnabled</a>	1..1
<a href="#">TcplpPrecondAssertEnabled</a>	1..1
<a href="#">TcplpPostcondAssertEnabled</a>	1..1
<a href="#">TcplpStaticAssertEnabled</a>	1..1
<a href="#">TcplpUnreachAssertEnabled</a>	1..1
<a href="#">TcplpInvariantAssertEnabled</a>	1..1

<b>Parameter Name</b>	<b>TcplpDefProgEnabled</b>
<b>Label</b>	Enable Defensive Programming
<b>Description</b>	<p>Enables or disables the defensive programming feature for the module Tcplp.</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>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcplpPrecondAssertEnabled</b>
-----------------------	----------------------------------



<b>Label</b>	Enable Precondition Assertions
<b>Description</b>	<p>Enables handling of precondition assertion checks reported from the module Tcplp.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>TcpIpDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>TcpIpDefProgEnabled</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>TcplpPostcondAssertEnabled</b>
<b>Label</b>	Enable Postcondition Assertions
<b>Description</b>	<p>Enables handling of postcondition assertion checks reported from the module Tcplp.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>TcpIpDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>TcpIpDefProgEnabled</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>TcplpStaticAssertEnabled</b>
<b>Label</b>	Enable Static Assertions
<b>Description</b>	<p>Enables handling of static assertion checks reported from the module Tcplp.</p> <p>Dependency on parameter(s):</p>



	<ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>TcpIpDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>TcpIpDefProgEnabled</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>TcpIpUnreachAssertEnabled</b>
<b>Label</b>	Enable Unreachable Code Assertions
<b>Description</b>	<p>Enables handling of unreachable code assertion checks reported from the module Tcplp.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>TcpIpDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>TcpIpDefProgEnabled</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>TcpIpInvariantAssertEnabled</b>
<b>Label</b>	Enable Invariant Assertions
<b>Description</b>	<p>Enables handling of invariant assertion checks reported from functions of the module Tcplp.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>TcpIpDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>TcpIpDefProgEnabled</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

### 5.8.1.71. TcplpGeneral

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">TcplpV4General</a>	1..1	This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack for IPv4.
<a href="#">TcplpV6General</a>	1..1	This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack for IPv6.

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">TcplpBufferMemory</a>	1..1
<a href="#">TcplpNumMemoryBlocks</a>	1..1
<a href="#">TcplpDevErrorDetect</a>	1..1
<a href="#">TcplpDhcpServerEnabled</a>	1..1
<a href="#">TcplpMainFunctionPeriod</a>	1..1
<a href="#">TcplpResetIpAssignmentApi</a>	1..1
<a href="#">TcplpScalabilityClass</a>	1..1
<a href="#">TcplpTcpEnabled</a>	1..1
<a href="#">TcplpTcpSocketMax</a>	1..1
<a href="#">TcplpTcpV6SocketMax</a>	1..1
<a href="#">TcplpUdpEnabled</a>	1..1
<a href="#">TcplpUdpMaxMulticastRxInstances</a>	1..1
<a href="#">TcplpUdpSocketMax</a>	1..1
<a href="#">TcplpUdpV6SocketMax</a>	1..1
<a href="#">TcplpUdpAllowImplicitBroadcastReception</a>	1..1



#### Parameters included

<a href="#">TcplpVersionInfoApi</a>	1..1
<a href="#">TcplpCtrlMax</a>	1..1
<a href="#">TcplpRelocatablePbcfgEnable</a>	1..1
<a href="#">TcplpTransmitRetriesMax</a>	1..1
<a href="#">TcplpGetAndResetMeasurementDataApi</a>	1..1
<a href="#">TcplpEnableMainFunctionTx</a>	1..1
<a href="#">TcplpDhcpGetStatusApi</a>	1..1
<a href="#">TcplpIcmpTransmitErrorApi</a>	1..1
<a href="#">TcplpSecurityMode</a>	1..1
<a href="#">TcplpCustomCsmlInterfaceHeaderFile</a>	0..1

Parameter Name	TcplpBufferMemory
<b>Description</b>	The functionality related to this parameter is not supported anymore. Memory for TCP sockets can be configured through the TcplpMemoryConfig container by adding a TcplpMemoryPool and setting TcplpMemoryBlockSize to TcplpBufferMemory devided by TcplpNumMemoryBlocks and TcplpMemoryBlockCount to TcplpNumMemoryBlocks  Memory size in bytes reserved for TCP/IP Tx buffers.
	<b>Optimization Effect:</b>  ▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	8320
<b>Range</b>	<=4294967295 >=1
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

Parameter Name	TcplpNumMemoryBlocks
<b>Description</b>	The functionality related to this parameter is not supported anymore. See parameter TcplpBufferMemory for more information.  Defines the number of memory blocks that TcplpBufferMemory is divided by.



<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	1	
<b>Configuration class</b>	VariantPostBuild:	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	TcplpDevErrorDetect	
<b>Description</b>	If true then TCP/IP will enable the error-reporting to the Development Error Tracker (DET).	
<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>	TcplpDhcpServerEnabled	
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>	
	Enables (TRUE) or disables (FALSE) the DHCP (Dynamic Host Configuration Protocol) Server.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	BOOLEAN	
<b>Configuration class</b>	VariantPostBuild:	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	TcplpMainFunctionPeriod	
<b>Description</b>	Allow to configure the time for the MainFunction (in seconds). This configuration value shall be equal to the value in the ScheduleManager module.  <b>Note: 100ms should be dividable by the Mainfunction period without rest, otherwise it would lead to incorrect timeout calculations</b>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.1	
<b>Configuration class</b>	VariantPostBuild:	VariantPostBuild



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

<b>Parameter Name</b>	<b>TcpIpResetIpAssignmentApi</b>	
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Enables/disables the API TcpIp_ResetIpAssignment of a DHCP-client.</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>TcpIpScalabilityClass</b>	
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>In order to customize the TcpIp Stack to the specific needs of the user it can be scaled according to the scalability classes.</p>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	ENUMERATION	
<b>Default value</b>	SC1	
<b>Range</b>	SC1 SC2 SC3	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>TcpIpTcpEnabled</b>	
<b>Description</b>	<p>Enable or disable the TCP support.</p> <ul style="list-style-type: none"> <li>▶ true: Enables protocol TCP.</li> <li>▶ false: Disables protocol TCP.</li> </ul>	
<b>Optimization Effect:</b>	<ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> Disabling this parameter reduces the ROM consumption of the module configuration.</li> </ul>	



	<ul style="list-style-type: none"> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</li> <li>▶ <b>RAM reduction:</b> Disabling this parameter reduces the RAM consumption of the module.</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>TcpIpTcpSocketMax</b>
<b>Description</b>	Maximum number of TCP IPv4 sockets.  <b>Optimization Effect:</b>
	<ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	1
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU

<b>Parameter Name</b>	<b>TcpIpTcpV6SocketMax</b>
<b>Description</b>	Maximum number of TCP IPv6 sockets.  <b>Optimization Effect:</b>
	<ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..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>TcpIpUdpEnabled</b>
-----------------------	------------------------



<b>Description</b>	Enables or disabled support of UDP (User Datagram Protocol)
	<ul style="list-style-type: none"> <li>▶ true: UDP enabled</li> <li>▶ false: UDP 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_ECUC

<b>Parameter Name</b>	<b>TcpIpUdpMaxMulticastRxInstances</b>	
<b>Description</b>	For Udp broadcast or multicast incoming messages may match multiple sockets This configuration parameter specifies the maximum number of sockets elected for reception.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	2	
<b>Range</b>	>=1 <=255	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>TcpIpUdpSocketMax</b>	
<b>Description</b>	Maximum number of UDP IPv4 sockets.  <b>Optimization Effect:</b>	
	<ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	1	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>TcpIpUdpV6SocketMax</b>	



<b>Description</b>	Maximum number of UDP IPv6 sockets.
	<b>Optimization Effect:</b> <ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..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>TcpIpUdpAllowImplicitBroadcastReception</b>
<b>Description</b>	TRUE: Limited broadcast addresses are generated and added automatically. UDP packet received on either Limited broadcast or All nodes multicast addresses can be processed by other sockets bound to the same controller. FALSE: Limited broadcast addresses have to be added manually. UDP packet received on either Limited broadcast or All nodes multicast addresses can't be processed by other sockets bound to the same controller.
<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>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpVersionInfoApi</b>
<b>Description</b>	Switches the API service <code>TcpIp_GetVersionInfo()</code> on or off. <ul style="list-style-type: none"> <li>▶ true: <code>TcpIp_GetVersionInfo()</code> is implemented as function.</li> <li>▶ false: <code>TcpIp_GetVersionInfo()</code> is implemented as preprocessor macro.</li> </ul> <b>Optimization Effect:</b> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</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>TcpIpCtrlMax</b>
<b>Description</b>	<p>This parameter defines the maximum index a EthIf controller referenced via parameter TcplpEthIfCtrlRef is allowed to have.</p> <p>This parameter has impact on the size of the PostBuild RAM.</p> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	4
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpRelocatablePbcfgEnable</b>
<b>Description</b>	<p>Enables/disable 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>
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpTransmitRetriesMax</b>
<b>Description</b>	The maximal number of retries to transmit a packet before a socket connection is reseted.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	3
<b>Range</b>	>=0



	<=255
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>TcpIpGetAndResetMeasurementDataApi</b>	
<b>Description</b>	Enables / Disables the Get and Reset Measurement Data API	
<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>TcpIpEnableMainFunctionTx</b>	
<b>Description</b>	Enables/Disables the availability of TcpIp_MainFunctionTx. This function is used to perform transmission tasks (e.g. transmission of TCP segments after TcpIp_TcpTransmit) immediately without TcpIpMainFunctionPeriod delay. <ul style="list-style-type: none"> <li>▶ true: TcpIp_MainFunctionTx is available.</li> <li>▶ false: TcpIp_MainFunctionTx is 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>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>TcpIpDhcpGetStatusApi</b>	
<b>Description</b>	Enables/Disables the API TcpIp_DhcpGetStatus() This function is used to get the status of the dhcp address assignment <ul style="list-style-type: none"> <li>▶ true: TcpIp_DhcpGetStatus is available.</li> <li>▶ false: TcpIp_DhcpGetStatus is 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>	Elektrobit Automotive GmbH
<b>Parameter Name</b>	<b>TcpIpIcmpTransmitErrorApi</b>
<b>Description</b>	<p>Enables/Disables the API <code>TcpIp_IcmpTransmitError()</code>. This function constructs the ICMP header and payload and invokes the transmission of the ICMP frame over EthIf.</p> <ul style="list-style-type: none"> <li>▶ true: <code>TcpIp_IcmpTransmitError</code> is available.</li> <li>▶ false: <code>TcpIp_IcmpTransmitError</code> is 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>	Elektrobit Automotive GmbH
<b>Parameter Name</b>	<b>TcpIpSecurityMode</b>
<b>Description</b>	<p>Sets the security mode for TcpIp communication</p> <p>Available choices:</p> <ul style="list-style-type: none"> <li>▶ NO_SECURITY: No security applied to TcpIp communication.</li> <li>▶ FIREWALL: Enables the availability to define bypass/discard policies.</li> <li>▶ STATIC_IPSEC: Enables the availability to define bypass/secured/discard policies and manual security associations.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	NO_SECURITY
<b>Range</b>	NO_SECURITY FIREWALL STATIC_IPSEC
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH
<b>Parameter Name</b>	<b>TcpIpCustomCsmInterfaceHeaderFile</b>
<b>Description</b>	Allows to modify the interface to the Csm Module. Specify a c header file which contains the definition of custom Csm functions.
<b>Multiplicity</b>	0..1



Type	STRING
Configuration class	<b>PreCompile:</b> VariantPostBuild
Origin	Elektrobit Automotive GmbH

### 5.8.1.72. TcplpIpV4General

Parameters included	
Parameter name	Multiplicity
<a href="#">TcplpArpEnabled</a>	1..1
<a href="#">TcplpAutolpEnabled</a>	1..1
<a href="#">TcplpDhcpClientEnabled</a>	1..1
<a href="#">TcplpDhcpSimpleClientEnabled</a>	1..1
<a href="#">TcplpIcmpEnabled</a>	1..1
<a href="#">TcplpIpV4Enabled</a>	1..1
<a href="#">TcplpLocalAddrIpv4EntriesMax</a>	1..1
<a href="#">TcplpPathMtuDiscoveryEnabled</a>	1..1

Parameter Name	TcplpArpEnabled
Description	Enables or disabled support of ARP (Address Resolution Protocol). <ul style="list-style-type: none"> <li>▶ true: ARP enabled</li> <li>▶ false: ARP disabled</li> </ul>
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	<b>VariantPostBuild:</b> VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcplpAutolpEnabled
Description	Enable or disable the Auto-IP support. <ul style="list-style-type: none"> <li>▶ true: Enables Auto-IP.</li> <li>▶ false: Disables Auto-IP.</li> </ul>
Optimization Effect:	



	<ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> Disabling this parameter reduces the ROM consumption of the module configuration.</li> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</li> <li>▶ <b>RAM reduction:</b> Disabling this parameter reduces the RAM consumption of the module.</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

Parameter Name	TcpIpDhcpClientEnabled
<b>Description</b>	Enable or disable the DHCP client support. <ul style="list-style-type: none"> <li>▶ true: Enables DHCP client.</li> <li>▶ false: Disables DHCP client.</li> </ul> <b>Optimization Effect:</b> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> Disabling this parameter reduces the ROM consumption of the module configuration.</li> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</li> <li>▶ <b>RAM reduction:</b> Disabling this parameter reduces the RAM consumption of the module.</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

Parameter Name	TcpIpDhcpSimpleClientEnabled
<b>Description</b>	Enable or disable the DHCP Simple client support. The IP address is assigned through an exchange of 2 messages with the DHCP server. Client sends a DHCPDISCOVER with XID set to the lower 4 bytes of the MAC address. If server responds with a DHCPOFFER with the XID set to the client's MAC address, the client sets its own IP address to that given in the YIADDR field.



	<ul style="list-style-type: none"> <li>▶ true: Enables DHCP client.</li> <li>▶ false: Disables DHCP client.</li> </ul> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> Disabling this parameter reduces the ROM consumption of the module configuration.</li> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</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>TcpIpIcmpEnabled</b>
<b>Description</b>	Enables or disabled support of ICMP (Internet Control Message Protocol). <ul style="list-style-type: none"> <li>▶ true: ICMP enabled</li> <li>▶ false: ICMP 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_ECUC

<b>Parameter Name</b>	<b>TcpIpIplpV4Enabled</b>
<b>Description</b>	Enables (TRUE) or disables (FALSE) support of IPv4 (Internet Protocol version 4).
<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>TcpIpLocalAddrIpv4EntriesMax</b>
<b>Description</b>	Maximum number of IPv4 address table entries in PostBuild configuration.



	<p>This parameter has impact on the size of the PostBuild RAM.</p> <p><b>Optimization Effect:</b></p> <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> Selecting a small value for this parameter reduces the ROM consumption of the module configuration.</li> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	10
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>TcpIpPathMtuDiscoveryEnabled</b>
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Enables (TRUE) or disables (FALSE) the discovery of the maximum transmission unit on a path according to IETF Rfc 1191.</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

### 5.8.1.73. TcpIpV6General

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">TcplpDhcpV6ClientEnabled</a>	1..1
<a href="#">TcplpV6ExtensionHeaderFilterEnabled</a>	1..1
<a href="#">TcplpV6Enabled</a>	1..1
<a href="#">TcplpV6PathMtuDiscoveryEnabled</a>	1..1
<a href="#">TcplpLocalAddrIpv6EntriesMax</a>	1..1
<a href="#">TcplpNdpAddressResolutionUnreachabilityDetectionEnabled</a>	1..1



#### Parameters included

<a href="#">TcplpNdpPrefixAndRouterDiscoveryEnabled</a>	1..1
<a href="#">TcplpSlaacLinkLocalEnabled</a>	1..1
<a href="#">TcplpSlaacLinkLocalEntriesMax</a>	1..1
<a href="#">TcplpIpV6TrafficClassFilterEnabled</a>	1..1
<a href="#">TcplpIpV6FlowLabelFilterEnabled</a>	1..1

<b>Parameter Name</b>	<b>TcplpDhcpV6ClientEnabled</b>	
<b>Description</b>	Enables (TRUE) or disables (FALSE) the DHCPv6 (Dynamic Host Configuration Protocol for IPv6) Client.	
<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>TcplpIpV6ExtensionHeaderFilterEnabled</b>	
<b>Description</b>	Enables (TRUE) or disables (FALSE) filtering of IPv6 extension headers <ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> Disabling this parameter reduces the ROM consumption of the module configuration.</li> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</li> <li>▶ <b>RAM reduction:</b> Disabling this parameter reduces the RAM consumption of the module.</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>TcplpIpV6Enabled</b>	
<b>Description</b>	Enables (TRUE) or disables (FALSE) support of IPv6 (Internet Protocol version 6).	
<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>TcpIpIpV6PathMtuDiscoveryEnabled</b>	
<b>Description</b>	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Enables (TRUE) or disables (FALSE) Path MTU Discovery support for IPv6 according to IETF RFC 1981.</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>TcpIpLocalAddrIpv6EntriesMax</b>	
<b>Description</b>	Maximum number of LocalAddr table entries for IPv6	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Range</b>	<p>&lt;=255</p> <p>&gt;=0</p>	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	
<b>Parameter Name</b>	<b>TcpIpNdpAddressResolutionUnreachabilityDetectionEnabled</b>	
<b>Description</b>	<p>Note: Address Resoultion and Neighbor Unreachability Detetion is always turned on per default and cannot be turned off.</p> <p>Enables (TRUE) or disables (FALSE) support of Address Resoultion and Neighbor Unreachability Detetion via NDP.</p>	
<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>TcpIpNdpPrefixAndRouterDiscoveryEnabled</b>	
<b>Description</b>	Enables (TRUE) or disables (FALSE) support of Prefix and Router Discovery via NDP.	
<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>TcpIpSlaacLinkLocalEnabled</b>	
<b>Description</b>	Enable or disable the Stateless Address Auto Configuration of IPv6 Link Local Addresses.  ▶ true: Enables Local Slaac. ▶ false: Disables Local Slaac.	
<b>Optimization Effect:</b>	<ul style="list-style-type: none"> <li>▶ <b>ROM reduction (config):</b> Disabling this parameter reduces the ROM consumption of the module configuration.</li> <li>▶ <b>ROM reduction (code):</b> Disabling this parameter reduces the ROM consumption of the module code.</li> <li>▶ <b>RAM reduction:</b> Disabling this parameter reduces the RAM consumption of the module.</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>TcpIpSlaacLinkLocalEntriesMax</b>	
<b>Description</b>	This parameter defines the maximum number of interface which are allowed to have an entry in TcplpLocalAddr assigned with TcplpAssignmentMethod set to TCPIP_LINKLOCAL (Ipv6).  This parameter has impact on the size of the PostBuild RAM.	

	<b>Optimization Effect:</b> <ul style="list-style-type: none"> <li>▶ <b>RAM reduction:</b> Selecting a small value for this parameter reduces the RAM consumption of the module.</li> </ul>
<b>Multiplicity</b>	1..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>Description</b>	Enables Traffic Class filter for received IPv6 packets. When enabled Tcplp will discard received packets with Traffic Class field value different than zero. <ul style="list-style-type: none"> <li>▶ true: value filtering enabled</li> <li>▶ false: value filtering 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>	Elektrobit Automotive GmbH
<b>Parameter Name</b>	
<b>Description</b>	Enables Flow Label filter for received IPv6 packets. When enabled Tcplp will discard received packets with Flow Label field value different than zero. <ul style="list-style-type: none"> <li>▶ true: value filtering enabled</li> <li>▶ false: value filtering 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>	Elektrobit Automotive GmbH

## 5.8.2. Recommended configurations



### 5.8.2.1. TcplpRecConfiguration

Containers included	
Container name	Container definition
<a href="#">TcplpConfig</a>	<a href="#">TcplpConfig</a>

Parameters included	
Parameter name	Value

#### 5.8.2.1.1. TcplpConfig

Containers included	
Container name	Container definition
<a href="#">TcplpSocketOwnerConfig</a>	<a href="#">TcplpConfig TcplpSocketOwnerConfig</a>
<a href="#">TcplpIpConfig</a>	<a href="#">TcplpConfig TcplpIpConfig</a>

Parameters included	
Parameter name	Value

#### 5.8.2.1.2. TcplpSocketOwnerConfig

Containers included	
Container name	Container definition
<a href="#">TcplpSocketOwner</a>	<a href="#">TcplpSocketOwner</a>

Parameters included	
Parameter name	Value

#### 5.8.2.1.3. TcplpSocketOwner

Parameters included	
Parameter name	Value
<a href="#">TcplpSocketOwnerHeaderFileName</a>	(DISABLED)
<a href="#">TcplpSocketOwnerCopyTxDataName</a>	(DISABLED)



#### Parameters included

<a href="#">TcplpSocketOwnerLocallpAddrAssignmentChg-Name</a>	(DISABLED)
<a href="#">TcplpSocketOwnerRxIndicationName</a>	(DISABLED)
<a href="#">TcplpSocketOwnerTcpAcceptedName</a>	(DISABLED)
<a href="#">TcplpSocketOwnerTcpConnectedName</a>	(DISABLED)
<a href="#">TcplpSocketOwnerTxConfirmationName</a>	(DISABLED)
<a href="#">TcplpSocketOwnerUpperLayerType</a>	SOAD

#### 5.8.2.1.4. TcplpIpConfig

##### Containers included

Container name	Container definition
<a href="#">TcplpIpV4Config</a>	<a href="#">TcplpIpConfig</a> <a href="#">TcplpV4Config</a>

##### Parameters included

Parameter name	Value

#### 5.8.2.1.5. TcplpIpV4Config

##### Containers included

Container name	Container definition
<a href="#">TcplpArpConfig</a>	<a href="#">TcplpArpConfig</a>

##### Parameters included

Parameter name	Value

#### 5.8.2.1.6. TcplpArpConfig

##### Parameters included

Parameter name	Value
<a href="#">TcplpArpPacketQueueEnabled</a>	false
<a href="#">TcplpArpTableSizeMax</a>	5
<a href="#">TcplpArpNumGratuitousARPonStartup</a>	0



#### Parameters included

<a href="#">TcplpArpTableEntryTimeout</a>	16.0
---	------

## 5.8.3. Application programming interface (API)

### 5.8.3.1. Type definitions

#### 5.8.3.1.1. Tcplp\_ArpCacheEntryType

<b>Purpose</b>	Specifies an entry in the ARP cache.	
<b>Type</b>	struct	
<b>Members</b>	uint32 InetAddr	IPv4 address in network byte order.
	uint8 PhysAddr	physical address in network byte order
	uint8 State	state of the address entry

#### 5.8.3.1.2. Tcplp\_EventType

<b>Purpose</b>	Socket state machine events signalized to upper layer.
<b>Type</b>	uint8

#### 5.8.3.1.3. Tcplp\_IPsecStateType

<b>Purpose</b>	Specifies the IPsec state for a specific IPsec connection.
<b>Type</b>	uint8

#### 5.8.3.1.4. Tcplp\_IpAddrAssignmentType

<b>Purpose</b>	Specification of IP address assignment policy.
<b>Type</b>	uint8



#### 5.8.3.1.5. Tcplp\_IpAddrStateType

Purpose	Specifies the state for a specific IP address assignment.
Type	uint8

#### 5.8.3.1.6. Tcplp\_Ip\_HandleIdType

Purpose	Handle identifier.
Type	uint16
Description	Type definition of the handle identifier used in the CopyData functor interface.

#### 5.8.3.1.7. Tcplp\_Ip\_RxReturnType

Purpose	Return type used for processing of received datagrams inside Tcplp_RxIndication.
Type	uint8

#### 5.8.3.1.8. Tcplp\_NdpCacheEntryType

Purpose	Specifies an entry in the NDP cache.	
Type	struct	
Members	uint32 Inet6Addr	IPv6 address in network byte order.
	uint8 PhysAddr	physical address in network byte order
	uint8 State	state of the address entry

#### 5.8.3.1.9. Tcplp\_ParamIdType

Purpose	Type for the specification of all supported Parameter IDs.
Type	uint8

#### 5.8.3.1.10. Tcplp\_ProtocolType

Purpose	Protocol.
Type	uint8



Description	Type definition of the used protocol
-------------	--------------------------------------

#### 5.8.3.1.11. Tcplp\_ReturnType

Purpose	Tcplp specific return type.
Type	uint8

#### 5.8.3.1.12. Tcplp\_SockAddrPtrType

Purpose	Pointer to structure of type <a href="#">Tcplp_SockAddrType</a> .
Type	<a href="#">TcpIp_SockAddrType</a> *

#### 5.8.3.1.13. Tcplp\_SockAddrType

Purpose	Generic address structure.	
Type	struct	
Members	TcpIp_DomainType domain	This is the code for the address format of this address.
	uint32 data	Secures alignment and shall not be used.
Description	This structure defines a generic address type to pass on the API. This generic type abstract IPv4 and IPv6 addresses and shall be used to typecast Tcplp_SockAddrNetType and Tcplp_SockAddrNet6Type. The value of domain defines the underlying structure. Note: The member data secures the correct alignment of this structure and shall not be used in any way.	

#### 5.8.3.1.14. Tcplp\_SocketIdType

Purpose	Socket identifier.
Type	<a href="#">TcpIp_Ip_HandleIdType</a>
Description	Type definition of socket identifier

#### 5.8.3.1.15. Tcplp\_StateType

Purpose	Specifies the Tcplp state for a specific EthIf controller.
---------	--



Type	uint8
------	-------

### 5.8.3.2. Macro constants

#### 5.8.3.2.1. TCPIP\_AF\_INET

Purpose	Value identifies IP version 4 for Tcplp_DomainType.
Value	0x02U

#### 5.8.3.2.2. TCPIP\_AF\_INET6

Purpose	Value identifies IP version 6 for Tcplp_DomainType.
Value	0x1cU

#### 5.8.3.2.3. TCPIP\_AF\_UNSPEC

Purpose	Value identifies unspecified address family for Tcplp_DomainType.
Value	0x00U

#### 5.8.3.2.4. TCPIP\_DHCPV6\_OPTION\_FQDN

Purpose	Option TCPIP_DHCPV6_OPTION_FQDN on API <a href="#">Tcplp_DhcpReadOption()</a> and <a href="#">Tcplp_DhcpWriteOption()</a> .
Value	39U
Description	This option requests to read or write the Domain Name of the DHCPV6 client. Precondition: none ParameterValue: Points to underlying type of size uint8.

#### 5.8.3.2.5. TCPIP\_DHCP\_OPTION\_FQDN

Purpose	Option TCPIP_DHCP_OPTION_FQDN on API <a href="#">Tcplp_DhcpReadOption()</a> and <a href="#">Tcplp_DhcpWriteOption()</a> .
Value	81U



<b>Description</b>	This option requests to read or write the Domain Name of the DHCP client. Precondition: none ParameterValue: Points to underlying type of size uint8.
--------------------	---

#### 5.8.3.2.6. TCPIP\_E\_ADDRINUSE

<b>Purpose</b>	Det error Id TCPIP_E_ADDRINUSE.
<b>Value</b>	0x09U

#### 5.8.3.2.7. TCPIP\_E\_ADDRNOTASSIGNED

<b>Purpose</b>	operation failed, no local address assigned
<b>Value</b>	6U

#### 5.8.3.2.8. TCPIP\_E\_ADDRNOTAVAIL

<b>Purpose</b>	Det error Id TCPIP_E_ADDRNOTAVAIL.
<b>Value</b>	0x0aU

#### 5.8.3.2.9. TCPIP\_E\_AFNOSUPPORT

<b>Purpose</b>	Det error Id TCPIP_E_AFNOSUPPORT.
<b>Value</b>	0x0eU

#### 5.8.3.2.10. TCPIP\_E\_DESTADDRREQ

<b>Purpose</b>	Det error Id TCPIP_E_DESTADDRREQ.
<b>Value</b>	0x06U

#### 5.8.3.2.11. TCPIP\_E\_DROP

<b>Purpose</b>	operation failed, ignore the packet
<b>Value</b>	3U



#### 5.8.3.2.12. TCPIP\_E\_INV\_ARG

<b>Purpose</b>	Det error Id TCPIP_E_INV_ARG.
<b>Value</b>	0x03U

#### 5.8.3.2.13. TCPIP\_E\_INV\_SOCKADDR

<b>Purpose</b>	Det error Id TCPIP_E_INV_SOCKADDR.
<b>Value</b>	0x05U

#### 5.8.3.2.14. TCPIP\_E\_ISCONN

<b>Purpose</b>	Det error Id TCPIP_E_ISCONN.
<b>Value</b>	0x0bU

#### 5.8.3.2.15. TCPIP\_E\_MEMORY

<b>Purpose</b>	operation failed, out of memory
<b>Value</b>	5U

#### 5.8.3.2.16. TCPIP\_E\_MSGSIZE

<b>Purpose</b>	Det error Id TCPIP_E_MSGSIZE.
<b>Value</b>	0x07U

#### 5.8.3.2.17. TCPIP\_E\_NOBUFS

<b>Purpose</b>	Det error Id TCPIP_E_NOBUFS.
<b>Value</b>	0x04U

#### 5.8.3.2.18. TCPIP\_E\_NOPROTOOPT

<b>Purpose</b>	Det error Id TCPIP_E_NOPROTOOPT.
----------------	----------------------------------



<b>Value</b>	0x0dU
--------------	-------

#### 5.8.3.2.19. TCPIP\_E\_NOTCONN

<b>Purpose</b>	Det error Id TCPIP_E_NOTCONN.
<b>Value</b>	0x0cU

#### 5.8.3.2.20. TCPIP\_E\_NOTINIT

<b>Purpose</b>	Det error Id TCPIP_E_NOTINIT.
<b>Value</b>	0x01U

#### 5.8.3.2.21. TCPIP\_E\_NOT\_OK

<b>Purpose</b>	operation failed
<b>Value</b>	1U

#### 5.8.3.2.22. TCPIP\_E\_NOT\_PERMITTED

<b>Purpose</b>	operation failed, illegal request
<b>Value</b>	4U

#### 5.8.3.2.23. TCPIP\_E\_PARAM\_POINTER

<b>Purpose</b>	Det error Id TCPIP_E_PARAM_POINTER.
<b>Value</b>	0x02U

#### 5.8.3.2.24. TCPIP\_E\_PENDING

<b>Purpose</b>	operation failed temporarily, dependent operation not yet concluded
<b>Value</b>	7U



#### 5.8.3.2.25. TCPIP\_E\_PHYS\_ADDR\_MISS

Purpose	operation failed because of an ARP cache miss
Value	2U

#### 5.8.3.2.26. TCPIP\_E\_PROTOTYPE

Purpose	Det error Id TCPIP_E_PROTOTYPE.
Value	0x08U

#### 5.8.3.2.27. TCPIP\_IPADDR\_ASSIGNMENT\_ALL

Purpose	all configured methods with manual trigger
Value	5U

#### 5.8.3.2.28. TCPIP\_IPADDR\_ASSIGNMENT\_DHCP

Purpose	address obtained via DHCP
Value	3U

#### 5.8.3.2.29. TCPIP\_IPADDR\_ASSIGNMENT\_IPV6\_ROUTER

Purpose	address obtained via IPV6 ROUTER
Value	4U

#### 5.8.3.2.30. TCPIP\_IPADDR\_ASSIGNMENT\_LINKLOCAL

Purpose	address obtained via LINK-LOCAL
Value	1U

#### 5.8.3.2.31. TCPIP\_IPADDR\_ASSIGNMENT\_LINKLOCAL\_DOIP

Purpose	address obtained via LINK-LOCAL or DO-IP
---------	--



<b>Value</b>	2U
--------------	----

#### 5.8.3.2.32. TCPIP\_IPADDR\_ASSIGNMENT\_STATIC

<b>Purpose</b>	Static IPv4 address assignment.
<b>Value</b>	0U

#### 5.8.3.2.33. TCPIP\_IPADDR\_STATE\_ASSIGNED

<b>Purpose</b>	IP address assignment in use.
<b>Value</b>	0U

#### 5.8.3.2.34. TCPIP\_IPADDR\_STATE\_ONHOLD

<b>Purpose</b>	IP address assignment in use, but link lost.
<b>Value</b>	1U

#### 5.8.3.2.35. TCPIP\_IPADDR\_STATE\_UNASSIGNED

<b>Purpose</b>	IP address assignment unused.
<b>Value</b>	2U

#### 5.8.3.2.36. TCPIP IPPROTO\_TCP

<b>Purpose</b>	Protocol TCP.
<b>Value</b>	0x06U

#### 5.8.3.2.37. TCPIP IPPROTO\_UDP

<b>Purpose</b>	Protocol UDP.
<b>Value</b>	0x11U



#### **5.8.3.2.38. TCPIP\_IPSEC\_ACTIVE**

<b>Purpose</b>	IPsec is activated for this connection.
<b>Value</b>	0U

#### **5.8.3.2.39. TCPIP\_IPSEC\_INACTIVE**

<b>Purpose</b>	IPsec is deactivated for this connection.
<b>Value</b>	1U

#### **5.8.3.2.40. TCPIP\_IPSEC\_INVALID\_ICV**

<b>Purpose</b>	
<b>Value</b>	2U

#### **5.8.3.2.41. TCPIP\_IP\_RX\_DEST\_UNREACHABLE**

<b>Purpose</b>	Destination address not known to host.
<b>Value</b>	0x10U

#### **5.8.3.2.42. TCPIP\_IP\_RX\_EXT\_HDR\_OK**

<b>Purpose</b>	extension header successfully processed
<b>Value</b>	0x05U

#### **5.8.3.2.43. TCPIP\_IP\_RX\_FRAGMENT\_HEADER\_SHORT**

<b>Purpose</b>	Fragment header too short.
<b>Value</b>	0x08U

#### **5.8.3.2.44. TCPIP\_IP\_RX\_FRAGMENT\_LENGTH**

<b>Purpose</b>	Length problem during fragment reassembly.
----------------	--



<b>Value</b>	0x0AU
--------------	-------

#### 5.8.3.2.45. TCPIP\_IP\_RX\_FRAGMENT\_OFFSET

<b>Purpose</b>	Offset problem during fragment reassembly.
<b>Value</b>	0x0BU

#### 5.8.3.2.46. TCPIP\_IP\_RX\_FRAGMENT\_OUT\_OF\_BUFFERS

<b>Purpose</b>	data was not delivered to transport protocol - no more buffers to handle
<b>Value</b>	0x07U

#### 5.8.3.2.47. TCPIP\_IP\_RX\_FRAGMENT\_OVERLAP

<b>Purpose</b>	Overlap problem during fragment reassembly.
<b>Value</b>	0x0CU

#### 5.8.3.2.48. TCPIP\_IP\_RX\_FRAGMENT\_RESERVED

<b>Purpose</b>	Reserved flag usage problem during fragment reassembly.
<b>Value</b>	0x0DU

#### 5.8.3.2.49. TCPIP\_IP\_RX\_FRAGMENT\_TIMEOUT

<b>Purpose</b>	Timeout during fragment reassembly.
<b>Value</b>	0x09U

#### 5.8.3.2.50. TCPIP\_IP\_RX\_INVALID\_HEADER

<b>Purpose</b>	Basic header check failed.
<b>Value</b>	0x0FU



#### 5.8.3.2.51. TCPIP\_IP\_RX\_NOT\_OK

Purpose	data was not accepted
Value	0x01U

#### 5.8.3.2.52. TCPIP\_IP\_RX\_OK

Purpose	data was accepted
Value	0x00U

#### 5.8.3.2.53. TCPIP\_IP\_RX\_PORT\_UNREACHABLE

Purpose	data was not accepted by transport protocol - port unknown
Value	0x03U

#### 5.8.3.2.54. TCPIP\_IP\_RX\_PROTOCOL\_UNREACHABLE

Purpose	data was not delivered to transport protocol - protocol unknown
Value	0x02U

#### 5.8.3.2.55. TCPIP\_IP\_RX\_PROTOCOL\_UNREACHABLE\_FRAG

Purpose	data was not delivered to transport protocol - protocol unknown
Value	0x04U

#### 5.8.3.2.56. TCPIP\_IP\_RX\_RES\_POLICYCHECK

Purpose	Reserved values for policy check handlers start here.
Value	0x60U

#### 5.8.3.2.57. TCPIP\_IP\_RX\_SRCADDR\_INVALID

Purpose	Data not accepted because of invalid source address.
---------	--



<b>Value</b>	0x11U
--------------	-------

#### 5.8.3.2.58. TCPIP\_IP\_RX\_UNRECOGNIZED\_OPTION

<b>Purpose</b>	data was not accepted because of unrecognized ipv6 option code
<b>Value</b>	0x06U

#### 5.8.3.2.59. TCPIP\_IP\_RX\_UNRECOGNIZED\_ROUTING\_TYPE

<b>Purpose</b>	Reserved flag Segments Left is non-zero with unrecognized Routing Type.
<b>Value</b>	0x0EU

#### 5.8.3.2.60. TCPIP\_LOCALADDRID\_ANY

<b>Purpose</b>	Any local IP address.
<b>Value</b>	254U
<b>Description</b>	This macro defines a value to select any local IP address (instead of a specific one) used for API service <a href="#">Tcplp_Bind()</a> .

#### 5.8.3.2.61. TCPIP\_OK

<b>Purpose</b>	operation completed successfully
<b>Value</b>	0U

#### 5.8.3.2.62. TCPIP\_PARAMID\_DSCP

<b>Purpose</b>	The 6-bit Differentiated Service Code Point according to IETF RFC 2474.
<b>Value</b>	0x0bU

#### 5.8.3.2.63. TCPIP\_PARAMID\_FLOWLABEL

<b>Purpose</b>	The 20-bit Flow Label according to IETF RFC 6437.
----------------	---



<b>Value</b>	0x0aU
--------------	-------

#### 5.8.3.2.64. TCPIP\_PARAMID\_FRAMEPrio

<b>Purpose</b>	Specifies the frame priority for outgoing frames on the socket.
<b>Value</b>	0x01U

#### 5.8.3.2.65. TCPIP\_PARAMID\_PATHMTU\_ENABLE

<b>Purpose</b>	Specifies if the Path MTU Discovery shall be performed on the related socket.
<b>Value</b>	0x09U

#### 5.8.3.2.66. TCPIP\_PARAMID\_TCP\_KEEPALIVE

<b>Purpose</b>	Specifies if TCP Keep Alive Probes are sent on the socket connection.
<b>Value</b>	0x03U

#### 5.8.3.2.67. TCPIP\_PARAMID\_TCP\_KEEPALIVE\_INTERVAL

<b>Purpose</b>	Specifies the interval between subsequent keepalive probes.
<b>Value</b>	0x07U

#### 5.8.3.2.68. TCPIP\_PARAMID\_TCP\_KEEPALIVE\_PROBES\_MAX

<b>Purpose</b>	Specifies the maximum number of times that a keepalive probe is retransmitted.
<b>Value</b>	0x06U

#### 5.8.3.2.69. TCPIP\_PARAMID\_TCP\_KEEPALIVE\_TIME

<b>Purpose</b>	Specifies the time between the last data packet sent and the first keepalive probe.
----------------	---



<b>Value</b>	0x05U
--------------	-------

#### 5.8.3.2.70. TCPIP\_PARAMID\_TCP\_NAGLE

<b>Purpose</b>	Specifies if the Nagle Algorithm according to IETF RFC 896 is enabled or not.
<b>Value</b>	0x02U

#### 5.8.3.2.71. TCPIP\_PARAMID\_TCP\_OPTIONFILTER

<b>Purpose</b>	Specifies which TCP option filter shall be applied on the related socket.
<b>Value</b>	0x08U

#### 5.8.3.2.72. TCPIP\_PARAMID\_TCP\_RXWND\_MAX

<b>Purpose</b>	Specifies the maximum TCP receive window for the socket.
<b>Value</b>	0x00U

#### 5.8.3.2.73. TCPIP\_PARAMID\_TTL

<b>Purpose</b>	Specifies the time to live value for outgoing frames on the socket.
<b>Value</b>	0x04U

#### 5.8.3.2.74. TCPIP\_PARAMID\_UDP\_CHECKSUM

<b>Purpose</b>	Specifies if UDP checksum handling shall be enabled (TRUE) or skipped (FALSE) on the related socket.
<b>Value</b>	0x0cU

#### 5.8.3.2.75. TCPIP\_PARAMID\_UDP\_UNSPECIFIED\_IP

<b>Purpose</b>	Specifies if unspecified ip address shall be used for transmission as source.
----------------	---



<b>Value</b>	0x81U
--------------	-------

#### 5.8.3.2.76. TCPIP\_PHYS\_ADDR\_ENTRY\_STATE\_DELAY

<b>Purpose</b>	
<b>Value</b>	0x10U

#### 5.8.3.2.77. TCPIP\_PHYS\_ADDR\_ENTRY\_STATE\_FREE

<b>Purpose</b>	
<b>Value</b>	0x40U

#### 5.8.3.2.78. TCPIP\_PHYS\_ADDR\_ENTRY\_STATE\_INCOMPLETE

<b>Purpose</b>	
<b>Value</b>	0x02U

#### 5.8.3.2.79. TCPIP\_PHYS\_ADDR\_ENTRY\_STATE\_PROBE

<b>Purpose</b>	
<b>Value</b>	0x20U

#### 5.8.3.2.80. TCPIP\_PHYS\_ADDR\_ENTRY\_STATE\_REACHABLE

<b>Purpose</b>	
<b>Value</b>	0x04U

#### 5.8.3.2.81. TCPIP\_PHYS\_ADDR\_ENTRY\_STATE\_STALE

<b>Purpose</b>	
<b>Value</b>	0x08U



#### 5.8.3.2.82. TCPIP\_PHYS\_ADDR\_ENTRY\_STATE\_STATIC

Purpose	
Value	0x01U

#### 5.8.3.2.83. TCPIP\_PORT\_ANY

Purpose	Any port.
Value	0U
Description	This macro defines a value to select any port (instead of a specific one) used in <a href="#">Tcplp_SockAddrType</a> .

#### 5.8.3.2.84. TCPIP\_SOCKETID\_INVALID

Purpose	Represents the invalid value for function arguments of type Tcplp_SocketIdType.
Value	0xFFFFFU

#### 5.8.3.2.85. TCPIP\_STATE\_OFFLINE

Purpose	TCP/IP stack state for a specific EthIf controller is OFFLINE, i.e. no communication is possible.
Value	2U

#### 5.8.3.2.86. TCPIP\_STATE\_ONHOLD

Purpose	TCP/IP stack state for a specific EthIf controller is ONHOLD, i.e. no communication is currently possible (e.g. link down).
Value	1U

#### 5.8.3.2.87. TCPIP\_STATE\_ONLINE

Purpose	TCP/IP stack state for a specific EthIf controller is ONLINE, i.e. communication via at least one IP address is possible.
---------	---



<b>Value</b>	0U
--------------	----

#### 5.8.3.2.88. TCPIP\_STATE\_SHUTDOWN

<b>Purpose</b>	TCP/IP stack state for a specific EthIf controller is SHUTDOWN, i.e. release of resources using the EthIf controller, release of IP address assignment.
<b>Value</b>	4U

#### 5.8.3.2.89. TCPIP\_STATE\_STARTUP

<b>Purpose</b>	TCP/IP stack state for a specific EthIf controller is STARTUP, i.e. IP address assignment in progress or ready for manual start, communication is currently not possible.
<b>Value</b>	3U

#### 5.8.3.2.90. TCPIP\_TCP\_CLOSED

<b>Purpose</b>	TCP connection reached state 'CLOSE', all resources released.
<b>Value</b>	0x01U

#### 5.8.3.2.91. TCPIP\_TCP\_FIN\_RECEIVED

<b>Purpose</b>	TCP segment with set FIN flag received.
<b>Value</b>	0x02U

#### 5.8.3.2.92. TCPIP\_TCP\_RESET

<b>Purpose</b>	TCP connection reseted, all resources released.
<b>Value</b>	0x00U

#### 5.8.3.2.93. TCPIP\_UDP\_CLOSED

<b>Purpose</b>	UDP socket and all related resources have been released.
<b>Value</b>	0x03U



### 5.8.3.3. Functions

#### 5.8.3.3.1. TcpIp\_Bind

<b>Purpose</b>	Binds a TCP/UDP socket to a local address/port pair.	
<b>Synopsis</b>	<code>Std_ReturnType TcpIp_Bind ( TcpIp_SocketIdType SocketId ,      TcpIp_LocalAddrIdType LocalAddrId , uint16 * PortPtr );</code>	
<b>Service ID</b>	0x05	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SocketId	
<b>Parameters (in)</b>	SocketId	identifying the local socket resource.
	LocalAddrId	IP address identifier representing the local IP address and EthIf controller to bind the socket to. Note: to listen on all available EthIf controller an additional (specific) IpAddr entry can be configured in the TcpIp module configuration. The remote IP address of an incoming packet has no effect then. In case the socket shall be used as client socket, the IP address and EthIf controller represented by IpAddrId is used for transmission. Note: to use the default route an additional (specific) IpAddr entry has to be configured in the TcpIp module configuration. The IP address given by parameter localAddrPtr has not effect then.
<b>Parameters (in,out)</b>	PortPtr	port to which the socket shall be bound. In case the socket shall be used as listen socket TCPIP_PORT_ANY accepts incoming packets regardless of the destination port. In case the socket shall be used as client socket, TCPIP_PORT_ANY let the TCP/IP stack choose the local port automatically and write it back to this parameter.
<b>Return Value</b>	Result of operation	
	E_OK	The request has been accepted



	<code>E_NOT_OK</code>	The request has not been accepted (e.g. address in use)
<b>Description</b>	By this API service the TCP/IP stack is requested to bind a UDP or TCP socket to a local resource, specifying its IP address (via <code>IpAddrId</code> ) and port number. Sockets that shall be switched in a listening state later on must be bound to a local resource. Optionally this API can be used to specify the local IP address and port used by later calls of <a href="#">Tcplp_TcpConnect()</a> or <a href="#">Tcplp_Transmit()</a> .	

#### 5.8.3.3.2. Tcplp\_ChangeParameter

<b>Purpose</b>	Changes socket configuration.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>TcpIp_ChangeParameter</b> ( TcpIp_SocketIdType SocketId , TcpIp_ParamIdType ParameterId , const uint8 * ParameterValue );</pre>	
<b>Service ID</b>	0x0f	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SocketId	
<b>Parameters (in)</b>	SocketId	identifying the local socket resource.
	ParameterId	Identifier of the parameter to be changed
	ParameterValue	Points to memory location where new parameter value is stored.
<b>Return Value</b>	Result of operation	
	<code>E_OK</code>	The request has been accepted
	<code>E_NOT_OK</code>	The request has not been accepted
<b>Description</b>	By this API service the TCP/IP stack is requested to change a connection or socket parameter. Nagle algorithm: the Nagle algorithm may be controlled by this API. mem-Limit UDP: maximum size of UDP receive data queue in bytes, 0 for unlimited mem-Limit TCP: size of the maximum TCP receive window in bytes, 0 for default window size value	

#### 5.8.3.3.3. Tcplp\_Close

<b>Purpose</b>	Closes a socket.
<b>Synopsis</b>	<pre>Std_ReturnType <b>TcpIp_Close</b> ( TcpIp_SocketIdType SocketId , boolean Abort );</pre>



<b>Service ID</b>	0x04	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant for different SocketId	
<b>Parameters (in)</b>	SocketId	identifying the local socket resource.
	Abort	Terminate immediately?
<b>Return Value</b>	Result of operation	
	E_OK	The request has been accepted
	E_NOT_OK	The request has not been accepted (e.g. the TCP connection was not established before).
<b>Description</b>	By this API service the TCP/IP stack is requested to close connection and frees all resources. All buffers assigned to the socket are released regardless of UDP or TCP sockets. In case of a UDP socket the socket resources are released immediately and are available again for socket allocation via Tcplp_GetSocket(). In case of TCP, the 4-way handshake for closing a TCP socket starts. After completion the socket resources are released and the SoAd gets informed via SoAd_TcplpEvent().	

#### 5.8.3.3.4. Tcplp\_DhcpGetStatus

<b>Purpose</b>	Read DHCP status.	
<b>Synopsis</b>	Std_ReturnType <b>TcpIp_DhcpGetStatus</b> ( TcpIp_LocalAddrIdType LocalAddressId , uint8 * StatusPtr );	
<b>Parameters (in)</b>	LocalAddressId	Address Id of the DHCP address assignment
	StatusPtr	Pointer to a status variable. Contains the state information.
<b>Return Value</b>	Std_ReturnType	
	E_OK	Status was successfully retrieved
	E_NOT_OK	No Dhcp address assignment was found
<b>Description</b>	<p>This function returns the status of the Dhcp address assignment</p> <p>StatusPtr value DhcpV4 DhcpV6 0 INACTIVE INACTIVE port closed 1 START_DELAY START_DELAY waiting for limited broadcast address(v4)/Link local address (v6) 2 INIT INIT Startup state 3 SELECTING SOLICIT Searching for server (DISCOVER/SOLICITATIONS are sent) 4 REQUESTING REQUESTING Requesting specific lease (REQUESTS are sent) 5 BOUND BOUND Lease has been assigned, is in use 6 RENEWING RENEWING Renewing a lease 7 REBINDING REBINDING Requesting</p>	



	lease extension from any server 8 / REASSIGNING (DhcpV6 only) Requested specific lease in response to NoBinding status code 9
--	---

#### 5.8.3.3.5. Tcplp\_DhcpReadOption

<b>Purpose</b>	Retrieves value of a DHCP option.	
<b>Synopsis</b>	<code>Std_ReturnType TcpIp_DhcpReadOption ( TcpIp_LocalAddrIdType LocalIpAddrId , uint8 Option , uint8 * DataLength , uint8 * DataPtr );</code>	
<b>Service ID</b>	0x0D	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	LocalIpAddrId	IP address identifier representing the local IP address and EthIf controller for which the DHCP option shall be read.
	Option	DHCP option
	DataPtr	Pointer to memory containing DHCP option data
<b>Parameters (in,out)</b>	DataLength	in: contains the length of the provided data buffer. out: length of the actual data.
<b>Return Value</b>	Result of operation	
	E_OK	requested data retrieved successfully.
	E_NOT_OK	requested data could not be retrieved
<b>Description</b>	By this API service the TCP/IP stack retrieves DHCP option data identified by parameter option for already received DHCP options.	

#### 5.8.3.3.6. Tcplp\_DhcpV6ReadOption

<b>Purpose</b>	Retrieves value of a DHCPv6 option.	
<b>Synopsis</b>	<code>Std_ReturnType TcpIp_DhcpV6ReadOption ( TcpIp_LocalAddrIdType LocalIpAddrId , uint16 Option , uint16 * DataLength , uint8 * DataPtr );</code>	
<b>Service ID</b>	0x19	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	



<b>Parameters (in)</b>	LocalIpAddrId	IP address identifier representing the local IP address and EthIf controller for which the DHCP option shall be read.
	Option	DHCP option
	DataPtr	Pointer to memory containing DHCP option data
<b>Parameters (in,out)</b>	DataLength	in: contains the length of the provided data buffer. out: length of the actual data.
<b>Return Value</b>	Result of operation	
	E_OK	requested data retrieved successfully.
	E_NOT_OK	requested data could not be retrieved
<b>Description</b>	By this API service the TCP/IP stack retrieves DHCPv6 option data identified by parameter option for already received DHCPv6 options.	

#### 5.8.3.3.7. Tcplp\_DhcpV6WriteOption

<b>Purpose</b>	Write value of a DHCPv6 option.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>TcpIp_DhcpV6WriteOption</b> ( TcpIp_LocalAddrIdType LocalIpAddrId , uint16 Option , uint16 DataLength , const uint8 * DataPtr );</pre>	
<b>Service ID</b>	0x1a	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	None Reentrant	
<b>Parameters (in)</b>	LocalIpAddrId	IP address identifier representing the local IP address and EthIf controller for which the DHCP option shall be written.
	Option	DHCP option, e.g. Host Name
	DataLength	length of DHCP option data
	DataPtr	Pointer to memory containing DHCP option data
<b>Return Value</b>	Result of operation	
	E_OK	no error occurred.
	E_NOT_OK	DHCP option data could not be written.
<b>Description</b>	By this API service the TCP/IP stack writes the DHCPv6 option data identified by parameter option.	



### 5.8.3.3.8. Tcplp\_DhcpWriteOption

<b>Purpose</b>	Write value of a DHCP option.		
<b>Synopsis</b>	<pre>Std_ReturnType <b>TcpIp_DhcpWriteOption</b> ( TcpIp_LocalAddrIdType LocalIpAddrId , uint8 Option , uint8 DataLength , const uint8 * DataPtr );</pre>		
<b>Service ID</b>	0x0E		
<b>Sync/Async</b>	Synchronous		
<b>Reentrancy</b>	None Reentrant		
<b>Parameters (in)</b>	LocalIpAddrId	IP address identifier representing the local IP address and EthIf controller for which the DHCP option shall be written.	
	Option	DHCP option, e.g. Host Name	
	DataLength	length of DHCP option data	
	DataPtr	Pointer to memory containing DHCP option data	
<b>Return Value</b>	Result of operation		
	E_OK	no error occurred.	
	E_NOT_OK	DHCP option data could not be written.	
<b>Description</b>	By this API service the TCP/IP stack writes the DHCP option data identified by parameter option.		

### 5.8.3.3.9. Tcplp\_GetAndResetMeasurementData

<b>Purpose</b>	Reads and resets measurement data for diagnostic purposes.		
<b>Synopsis</b>	<pre>Std_ReturnType <b>TcpIp_GetAndResetMeasurementData</b> ( TcpIp_MeasurementIdxType MeasurementIdx , boolean MeasurementResetNeeded , uint32 * MeasurementDataPtr );</pre>		
<b>Parameters (in)</b>	MeasurementIdx	Index to select specific measurement data: TCPIP_MEAS_DROP_TCP (0x01) - Measurement index of dropped PDUs caused by invalid destination TCP-Port. TCPIP_MEAS_DROP_UDP (0x02) - Measurement index of dropped PDUs caused by invalid destination UDP-Port.	



		TCPIP_MEAS_DROP_IPV4 (0x03) - Measurement index of dropped datagrams caused by invalid IPv4 address TCPIP_MEAS_DROP_IPV6 (0x04) - Measurement index of dropped datagrams caused by invalid IPv6 address TCPIP_MEAS_RESERVED_1 (0x05-0x7F) - Reserved by AUTOSAR. TCPIP_MEAS_RESERVED_2 (0x80-0xEF) - Vendor specific range. TCPIP_MEAS_RESERVED_3 (0xF0-0xFE) - Reserved by AUTOSAR (future use). TCPIP_MEAS_ALL (0xFF) - Represents all measurement indexes.
	MeasurementResetNeeded	Flag to trigger a reset of the measurement data.
<b>Parameters (out)</b>	MeasurementDataPtr	Pointer to data buffer, where to copy measurement data.
<b>Return Value</b>	Std_ReturnType	
	E_OK	The function has been successfully executed.
	E_NOT_OK	The function could not be successfully executed.
<b>Description</b>	This service allows to read and reset detailed measurement data for diagnostic purposes. Get all MeasurementIdx's at once is not supported. TCPIP_MEAS_ALL shall only be used to reset all MeasurementIdx's at once. A NULL_PTR shall be provided for MeasurementDataPtr in this case.	

#### 5.8.3.3.10. TcpIp\_GetArpCacheEntries

<b>Purpose</b>	Retrieve all valid physical addresses from ARP cache.	
<b>Synopsis</b>	<code>Std_ReturnType TcpIp_GetArpCacheEntries ( uint8 ctrlIdx , uint32 * numberOfElements , TcpIp_ArpCacheEntryType * entryListPtr );</code>	
<b>Service ID</b>	0x1d	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	ctrlIdx	EthIf controller index to identify the related ARP table.



<b>Parameters (in,out)</b>	numberOfElements	In: Maximum number of entries that can be stored in output entryListPtr. Out: Number of entries written to output entryListPtr (Number of all entries in the cache if input value is 0).
<b>Parameters (out)</b>	entryListPtr	Pointer to memory where the list of cache entries shall be stored.
<b>Return Value</b>	Result of operation	
	E_OK	physical address cache could be read.
	E_NOT_OK	physical address cache could not be read (i.e. no IPv4 instance active on this controller)
<b>Description</b>	This service copies entries from the physical address cache of the IPv4 instance that is active on the EthIf controller specified by ctrlIdx into a user provided buffer. The function will copy all or numberOfElements into the output list. If input value of numberOfElements is 0 the function will not copy any data but only return the number of valid entries in the cache. EntryListPtr may be NULL_PTR in this case.	

#### 5.8.3.3.11. Tcplp\_GetCtrlIdx

<b>Purpose</b>	Obtain controller index.	
<b>Synopsis</b>	Std_ReturnType <b>TcpIp_GetCtrlIdx</b> ( TcpIp_LocalAddrIdType LocalAddrId , uint8 * CtrlIdxPtr );	
<b>Service ID</b>	0x17	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	LocalAddrId	Local address identifier implicitly specifying the EthIf controller that shall be returned
<b>Parameters (out)</b>	CtrlIdxPtr	Pointer to the memory where the index of the controller related to LocalAddrId is stored
<b>Return Value</b>	Result of operation	
	E_OK	the request was successful
	E_NOT_OK	the request was not successful.
<b>Description</b>	Obtains the index of the controller related to LocalAddrId.	



### 5.8.3.3.12. `TcpIp_GetIpAddr`

<b>Purpose</b>	Obtain local IP address.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>TcpIp_GetIpAddr</b> ( TcpIp_LocalAddrIdType Local- AddrId , TcpIp_SockAddrType * IpAddrPtr , uint8 * NetmaskPtr , TcpIp_SockAddrType * DefaultRouterPtr );</pre>	
<b>Service ID</b>	0x10	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	LocalAddrId	Local address identifier referring to the local IP address which shall be obtained.
<b>Parameters (out)</b>	IpAddrPtr	Pointer to a struct where the IP address is stored. Struct members not related to the IP address are of arbitrary value and shall not be used.
	NetmaskPtr	Pointer to a struct where the netmask is stored. Struct members not related to the IP address are of arbitrary value and shall not be used.
	DefaultRouterPtr	Pointer to a struct where the Gateway IP address is stored. Struct members not related to the IP address are of arbitrary value and shall not be used.
<b>Return Value</b>	Result of operation	
	E_OK	The request was successful
	E_NOT_OK	The request was not successful
<b>Description</b>	Obtains the local IP address actually used by LocalAddrId.	

### 5.8.3.3.13. `TcpIp_GetNdpCacheEntries`

<b>Purpose</b>	Retrieve all valid physical addresses from NDP cache.
<b>Synopsis</b>	<pre>Std_ReturnType <b>TcpIp_GetNdpCacheEntries</b> ( uint8 ctrlIdx , uint32 * numberOfElements , TcpIp_NdpCacheEntryType * en- tryListPtr );</pre>
<b>Service ID</b>	0x1d



<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	ctrlIdx	EthIf controller index to identify the related NDP table.
<b>Parameters (in,out)</b>	numberOfElements	In: Maximum number of entries that can be stored in output entryListPtr. Out: Number of entries written to output entryListPtr (Number of all entries in the cache if input value is 0).
<b>Parameters (out)</b>	entryListPtr	Pointer to memory where the list of cache entries shall be stored.
<b>Return Value</b>	Result of operation	
	E_OK	physical address cache could be read.
	E_NOT_OK	physical address cache could not be read (i.e. no IPv6 instance active on this controller)
<b>Description</b>	This service copies entries from the physical address cache of the IPv6 instance that is active on the EthIf controller specified by ctrlIdx into a user provided buffer. The function will copy all or numberOfElements into the output list. If input value of numberOfElements is 0 the function will not copy any data but only return the number of valid entries in the cache. EntryListPtr may be NULL_PTR in this case.	

#### 5.8.3.3.14. TcpIp\_GetPhysAddr

<b>Purpose</b>	Obtain local physical address.	
<b>Synopsis</b>	Std_ReturnType <b>TcpIp_GetPhysAddr</b> ( TcpIp_LocalAddrIdType LocalAddrId , uint8 * PhysAddrPtr );	
<b>Service ID</b>	0x11	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	LocalAddrId	Local address identifier implicitly specifying the EthIf controller for which the physical address shall be obtained.
<b>Parameters (out)</b>	PhysAddrPtr	Pointer to the memory where the physical source address (MAC address) in network byte order is stored



<b>Return Value</b>	Result of operation	
	E_OK	The request was successful
	E_NOT_OK	The request was not successful, e.g. no unique Ctrl specified via IpAddrId.
<b>Description</b>	Obtains the physical source address used by the EthIf controller implicitly specified via LocalAddrId.	

#### 5.8.3.3.15. Tcplp\_GetRemotePhysAddr

<b>Purpose</b>	Retrieve physical address from remote host.	
<b>Synopsis</b>	<pre>TcpIp_ReturnType <b>TcpIp_GetRemotePhysAddr</b> ( uint8 CtrlIdx , const TcpIp_SockAddrType * IpAddrPtr , uint8 * PhysAddrPtr , boolean initRes );</pre>	
<b>Service ID</b>	0x16	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	CtrlIdx	EthIf controller index to identify the related ARP/NDP table.
	IpAddrPtr	specifies the IP address for which the physical address shall be retrieved
	initRes	specifies if the address resolution shall be initiated (TRUE) or not (FALSE) in case the physical address related to the specified IP address is currently unknown.
<b>Parameters (out)</b>	PhysAddrPtr	Pointer to the memory where the physical address (MAC address)
<b>Return Value</b>	Result of operation	
	TCPIP_E_OK	specified IP address resolved, physical address provided via PhysAddrPtr
	TCPIP_E_NOT_OK	The request was not successful, e.g. invalid controller index
	TCPIP_E_PHYS_ADDR_MISS	physical address currently unknown (address resolution initiated if initRes set to TRUE)
<b>Description</b>	This service queries the IP/physical address translation table specified by CtrlIdx and returns the physical address related to the IP address specified by IpAddrPtr.	



### 5.8.3.3.16. `Tcplp_GetVersionInfo`

<b>Purpose</b>	Get version information of the Tcplp module.	
<b>Synopsis</b>	<pre>void <b>TcpIp_GetVersionInfo</b> ( Std_VersionInfoType * versioninfo ) ;</pre>	
<b>Service ID</b>	0x02	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (out)</b>	versioninfo	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>	

### 5.8.3.3.17. `Tcplp_IcmpTransmit`

<b>Purpose</b>	Requests to transmit an ICMP message.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>TcpIp_IcmpTransmit</b> ( TcpIp_LocalAddrIdType LocalIpAddrId , const TcpIp_SockAddrType * RemoteAddrPtr , uint8 Ttl , uint8 Type , uint8 Code , uint16 DataLength , const uint8 * DataPtr ) ;</pre>	
<b>Service ID</b>	0x0C	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	LocalIpAddrId	Id of Local Address used for transmission.
	RemoteAddrPtr	IP address of the remote host to transmit to.
	Ttl	Time-to-Live used for Ip header
	Type	ICMP Type to write into ICMP header
	Code	ICMP Code to write into ICMP header
	DataLength	Data length of the data to be transmitted.



	DataPtr	DataPtr points to a linear buffer of DataLength bytes.
<b>Return Value</b>	Result of operation	
	E_OK	UDP message transmission was successful.
	E_NOT_OK	UDP message transmission failed (E.g. caused by ARP cache miss).
<b>Description</b>	This service transmits an ICMP message	

#### 5.8.3.3.18. TcpIp\_IcmpTransmitError

<b>Purpose</b>	Requests to transmit an ICMP message based on a (EthIf) CtrlIdx.																	
<b>Synopsis</b>	<pre>Std_ReturnType <b>TcpIp_IcmpTransmitError</b> ( uint8 CtrlIdx , const Tcplp_SockAddrType * RemoteAddrPtr , uint8 Ttl , uint8 Type , uint8 Code , uint16 DataLength , const uint8 * DataPtr , const uint32 * SpecificValue );</pre>																	
<b>Service ID</b>	0xF2																	
<b>Sync/Async</b>	Synchronous																	
<b>Reentrancy</b>	Non-Reentrant																	
<b>Parameters (in)</b>	<table border="1"> <tr> <td>CtrlIdx</td> <td>Index of the ethernet controller (EthIfCtrlIdx).</td> </tr> <tr> <td>RemoteAddrPtr</td> <td>Pointer to remote IP address.</td> </tr> <tr> <td>Ttl</td> <td>Time-to-Live/Hop Limit used for Ip header</td> </tr> <tr> <td>Type</td> <td>ICMP Message Type to write into ICMP header</td> </tr> <tr> <td>Code</td> <td>ICMP Error Code to write into ICMP header</td> </tr> <tr> <td>DataLength</td> <td>Data Length of payload to transmit.</td> </tr> <tr> <td>DataPtr</td> <td>Points to the received data that contains the payload of the Icmp message</td> </tr> <tr> <td>SpecificValue</td> <td>Pointer to value set before the payload.</td> </tr> </table>		CtrlIdx	Index of the ethernet controller (EthIfCtrlIdx).	RemoteAddrPtr	Pointer to remote IP address.	Ttl	Time-to-Live/Hop Limit used for Ip header	Type	ICMP Message Type to write into ICMP header	Code	ICMP Error Code to write into ICMP header	DataLength	Data Length of payload to transmit.	DataPtr	Points to the received data that contains the payload of the Icmp message	SpecificValue	Pointer to value set before the payload.
CtrlIdx	Index of the ethernet controller (EthIfCtrlIdx).																	
RemoteAddrPtr	Pointer to remote IP address.																	
Ttl	Time-to-Live/Hop Limit used for Ip header																	
Type	ICMP Message Type to write into ICMP header																	
Code	ICMP Error Code to write into ICMP header																	
DataLength	Data Length of payload to transmit.																	
DataPtr	Points to the received data that contains the payload of the Icmp message																	
SpecificValue	Pointer to value set before the payload.																	
<b>Return Value</b>	<table border="1"> <tr> <td colspan="2">Result of operation</td></tr> <tr> <td>E_OK</td><td>Transmission request has been successfully performed.</td></tr> <tr> <td>E_NOT_OK</td><td>Transmission request failed.</td></tr> </table>		Result of operation		E_OK	Transmission request has been successfully performed.	E_NOT_OK	Transmission request failed.										
Result of operation																		
E_OK	Transmission request has been successfully performed.																	
E_NOT_OK	Transmission request failed.																	



<b>Description</b>	This function constructs the ICMP header and payload and and invokes the transmission of the Icmp frame over Ethlf.
--------------------	---

#### 5.8.3.3.19. Tcplp\_Init

<b>Purpose</b>	Initializes the Tcplp stack.	
<b>Synopsis</b>	<code>void TcpIp_Init ( const TcpIp_ConfigType * ConfigPtr );</code>	
<b>Service ID</b>	0x01	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	ConfigPtr	Address of the post-build time configuration data of the Tcplp module.
<b>Description</b>	This service initializes the TCP/IP Stack. The call of this service is mandatory before using the Tcplp instance for further processing.	

#### 5.8.3.3.20. Tcplp\_IsConnectionReady

<b>Purpose</b>	Checks if physical address is known and IpSec SA is established. If physical address is not known address resolution shall be triggered.	
<b>Synopsis</b>	<code>TcpIp_ReturnType TcpIp_IsConnectionReady ( TcpIp_SocketIdType SocketId , const TcpIp_SockAddrType * RemoteAddrPtr );</code>	
<b>Service ID</b>	0xF5	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SocketId	
<b>Parameters (in)</b>	SocketId	ID of the socket over which data shall be transmitted
	RemoteAddrPtr	UDP: Pointer to address and port of the remote host to which data shall be transmitted. TCP: NULL_PTR, for TCP a remote host does not need to be provided because address is already known by Tcplp.
<b>Return Value</b>	Indicates if data can be transmitted.	
	TCPIP_OK	Physical address known and IpSec SA exists, data can be sent



	TCPPIP_E_PENDING	Physical address not known or IpSec SA does not exist
	TCPPIP_E_NOT_OK	<ul style="list-style-type: none"> <li>- Transmission not allowed according to policy table OR</li> <li>▶ API called with invalid parameters if development error detection is enabled OR</li> <li>▶ An error occurs during processing of the request.</li> </ul>
<b>Description</b>	<p>By this API service Tcplp stack shall check if a connection is ready for transmission. This is ensured by: 1. If IPsec is enabled, checking that an IpSec Security Association (SA) is established if the connection shall be secured. 2. Checking that the physical address corresponding to the remote IP address is known (i.e. it is present in the ARP cache for IPv4 or in the neighbor cache for IPv6). If this is not the case an address resolution (i.e. sending an ARP request for IPv4 or a neighbor solicitation message) request is triggered.</p>	

#### 5.8.3.3.21. Tcplp\_IsValidConfig

<b>Purpose</b>	Checks compatibility of the post-build-time configuration.	
<b>Synopsis</b>	Std_ReturnType <b>TcpIp_IsValidConfig</b> ( const void * voidConfigPtr );	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	voidConfigPtr	Pointer to the configuration data of the Tcplp module.
<b>Return Value</b>	Result of compatibility check	
	E_OK	Provided configuration is compatible.
	E_NOT_OK	Provided configuration is not compatible.
<b>Description</b>	This service checks the compatibility of the post-build-time configuration against the source code.	

#### 5.8.3.3.22. Tcplp\_MainFunction

<b>Purpose</b>	Tcplp main function.
----------------	----------------------



<b>Synopsis</b>	<code>void TcpIp_MainFunction ( void );</code>
<b>Service ID</b>	0x15
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant
<b>Description</b>	This function performs the periodic actions of the Tcplp module (e.g. timer handling, state machine triggering).

#### 5.8.3.3.23. `Tcplp_MainFunctionTx`

<b>Purpose</b>	MainFunction for socket transmission.
<b>Synopsis</b>	<code>void TcpIp_MainFunctionTx ( void );</code>
<b>Description</b>	This function performs transmission tasks. e.g. transmission of TCP segments.

#### 5.8.3.3.24. `Tcplp_ReleaseIpAddrAssignment`

<b>Purpose</b>	Release an IP address assignment.	
<b>Synopsis</b>	<code>Std_ReturnType TcpIp_ReleaseIpAddrAssignment ( TcpIp_LocalAddrIdType LocalAddrId );</code>	
<b>Service ID</b>	0x0b	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	LocalAddrId	IP address index specifying the IP address for which an assignment shall be released.
<b>Return Value</b>	Result of operation	
	E_OK	The request has been accepted
	E_NOT_OK	The request has not been accepted
<b>Description</b>	By this API service the local IP address assignment for the IP address specified by LocalAddrId shall be released.	

#### 5.8.3.3.25. `Tcplp_RequestComMode`

<b>Purpose</b>	Request to change state of communication network.
----------------	---



<b>Synopsis</b>	<code>Std_ReturnType TcpIp_RequestComMode ( uint8 CtrlIdx , TcpIp_-StateType State );</code>	
<b>Service ID</b>	0x09	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	CtrlIdx	EthIf controller index to identify the communication network where the Tcplp state is requested.
	State	Requested Tcplp state.
<b>Return Value</b>	Result of operation	
	E_OK	Service accepted
	E_NOT_OK	Service denied
<b>Description</b>	By this API service the TCP/IP stack is requested to change the Tcplp state of the communication network identified by EthIf controller index.	

### 5.8.3.3.26. `Tcplp_RequestIpAddrAssignment`

<b>Purpose</b>	Request an IP address assignment.	
<b>Synopsis</b>	<code>Std_ReturnType TcpIp_RequestIpAddrAssignment ( TcpIp_LocalAddrIdType LocalAddrId , TcpIp_IpAddrAssignmentType Type , const TcpIp_SockAddrType * LocalIpAddrPtr );</code>	
<b>Service ID</b>	0x0a	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	LocalAddrId	IP address index specifying the IP address for which an assignment shall be requested.
	Type	Type of IP address assignment which shall be initiated.
	LocalIpAddrPtr	Pointer to structure containing the IP address which shall be assigned to the EthIf controller indirectly specified via LocalAddrId. Note: This parameter is only used in case the parameters Type is set to TCPIP_IPADDR_ASSIGNMENT_STATIC.



<b>Return Value</b>	Result of operation	
	E_OK	The request has been accepted
	E_NOT_OK	The request has not been accepted
<b>Description</b>	By this API service the local IP address assignment type for the IP address specified by LocalAddrId shall be requested.	

#### 5.8.3.3.27. Tcplp\_RequestIpSecMode

<b>Purpose</b>	Request to change state of communication security.	
<b>Synopsis</b>	Std_ReturnType <b>TcpIp_RequestIpSecMode</b> ( uint8 conId , TcpIp_-IPsecStateType reqState );	
<b>Service ID</b>	0xF2	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	conId	IpSec connection index to identify the communication where the Tcplp state is requested.
	reqState	Requested Tcplp state.
<b>Return Value</b>	Result of operation	
	E_OK	Service accepted
	E_NOT_OK	Service denied
<b>Description</b>	By this API service the TCP/IP stack is requested to change the IpSec state of the communication network identified by IpSec connection index.	

#### 5.8.3.3.28. Tcplp\_RxIndication

<b>Purpose</b>	Tcplp receive interface.
<b>Synopsis</b>	void <b>TcpIp_RxIndication</b> ( uint8 CtrlIdx , Eth_FrameType FrameType , boolean IsBroadcast , uint8 * PhysAddrPtr , uint8 * DataPtr , uint16 LenByte );
<b>Service ID</b>	0x14
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non-Reentrant



<b>Parameters (in)</b>	CtrlIdx	Index of the ethernet controller (EthIfCtr-Idx)
	FrameType	Value of ethernet header type-field.
	IsBroadcast	Indicates if the target MAC address is a broadcast address
	PhysAddrPtr	Points to MAC address of remote host (source MAC)
	DataPtr	Points to the received data. The data contains the payload of the Ethernet protocol (which excludes the Ethernet header but includes headers of higher layers).
	LenByte	Length of received data in units of bytes.
<b>Description</b>	This is the receive interface of the Tcplp stack. All received data must be passed to the Tcplp module using this API function.	

#### 5.8.3.3.29. Tcplp\_SetRemotePhysAddr

<b>Purpose</b>	Set physical address of remote host.	
<b>Synopsis</b>	<pre>TcpIp_ReturnType <b>TcpIp_SetRemotePhysAddr</b> ( uint8 CtrlIdx , const TcpIp_SockAddrType * IpAddrPtr , const uint8 * PhysAddrPtr , uint8 State );</pre>	
<b>Service ID</b>	0xF0	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	CtrlIdx	EthIf controller index to identify the related ARP/NDP table.
	IpAddrPtr	specifies the IP address for which the physical address shall be set in the ARP/NDP table
	PhysAddrPtr	specifies the physical address which shall be set in the ARP/NDP table
	State	TCPIP_PHYS_ADDR_ENTRY_STATE_FREE: remove the specified entry in the ARP/NDP table TCPIP_PHYS_ADDR_ENTRY_STATE_REACHABLE: add/update the specified entry in the ARP/NDP table



		NDP table TCPIP_PHYS_ADDR_ENTRY_STATE_STATIC: set the specified entry in the ARP/NDP table to static (entry will remain in the cache until it is deleted with Tcplp_SetRemotePhysAddr or Tcplp is reinitialized)
<b>Return Value</b>	Result of operation	
	TCPIP_OK	physical address successfully added/updated/removed
	TCPIP_E_NOT_OK	The request was not successful, e.g. invalid controller index
	TCPIP_E_PHYS_ADDR_MISS	physical address currently unknown (in case the entry shall be removed)
<b>Description</b>	This service adds, updates or removes a physical address from a remote host in the ARP/NDP table	

#### 5.8.3.3.30. Tcplp\_Socket\_getSocket

<b>Purpose</b>	Allocates resources for a new TCP/UDP socket.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>TcpIp_Socket_getSocket</b> ( TcpIp_DomainType Domain, , TcpIp_ProtocolType Protocol , TcpIp_SocketIdType * SocketIdP- tr , uint8 sockOwnerId );</pre>	
<b>Service ID</b>	0x03	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	Domain	IP address family. Possible values are AF_INET (IPv4) and AF_INET6 (IPv6) which is not supported at the moment.
	Protocol	Socket protocol as sub-family of parameter type. Possible values are IPPROTO_TCP and IPPROTO_UDP.
	sockOwnerId	Id of the socket owner (specifies call-back functions)
<b>Parameters (out)</b>	SocketIdPtr	Pointer to Tcplp_SocketIdType representing the requested SocketId. This SocketId must be provided for all further API calls which requires a SocketId. Note: SocketId



		only contains a valid value if return value is E_OK.
<b>Return Value</b>	Result of operation	
	E_OK	The request has been accepted
	E_NOT_OK	The request has not been accepted: no free socket resources found.
<b>Description</b>	By this API service the TCP/IP stack is requested to allocate a new socket. The configuration parameters TcplpUdpSocketMax and TcplpTcpSocketMax determines the maximum number of sockets. Note: Each accepted incoming TCP connection also allocates a socket resource.	

#### 5.8.3.3.31. Tcplp\_TcpConnect

<b>Purpose</b>	Trigger the TCP connection establishment to a remote host.	
<b>Synopsis</b>	Std_ReturnType <b>Tcplp_TcpConnect</b> ( TcpIp_SocketIdType SocketId , const TcpIp_SockAddrType * RemoteAddrPtr );	
<b>Service ID</b>	0x06	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
<b>Parameters (in)</b>	SocketId	Socket identifier of the related local socket resource.
	RemoteAddrPtr	IP address and port of the remote host to connect to.
<b>Return Value</b>	Result of operation	
	E_OK	The request has been accepted
	E_NOT_OK	The request has not been accepted, e.g. connection is already established or no route to destination specified by remoteAddrPtr found.
<b>Description</b>	By this API service the TCP/IP stack is requested to establish a TCP connection to the configured peer.	

#### 5.8.3.3.32. Tcplp\_TcpListen

<b>Purpose</b>	Start to listen for incoming TCP connection requests.
----------------	---



<b>Synopsis</b>	<code>Std_ReturnType TcpIp_TcpListen ( TcpIp_SocketIdType SocketId , uint16 MaxChannels );</code>	
<b>Service ID</b>	0x07	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
<b>Parameters (in)</b>	SocketId	Socket identifier of the related local socket resource.
	MaxChannels	Maximum number of new parallel connections established on this listen connection.
<b>Return Value</b>	Result of operation	
	E_OK	The request has been accepted
	E_NOT_OK	The request has not been accepted, e.g. connection is already established or no route to destination specified by remoteAddrPtr found.
<b>Description</b>	By this API service the TCP/IP stack is requested to listen on the TCP socket specified by the socket identifier.	

### 5.8.3.3.33. TcpIp\_TcpReceived

<b>Purpose</b>	Confirm the reception of data.	
<b>Synopsis</b>	<code>Std_ReturnType TcpIp_TcpReceived ( TcpIp_SocketIdType SocketId , uint32 Length );</code>	
<b>Service ID</b>	0x08	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
<b>Parameters (in)</b>	SocketId	Socket identifier of the related local socket resource.
	Length	Number of bytes finally consumed by the upper layer.
<b>Return Value</b>	Result of operation	
	E_OK	The request has been accepted
	E_NOT_OK	The request has not been accepted, e.g. invalid socket id.
<b>Description</b>	By this API service the reception of socket data is confirmed to the TCP/IP stack.	



#### 5.8.3.3.34. Tcplp\_TcpTransmit

<b>Purpose</b>	Requests to transmit data to remote destination via TCP protocol.		
<b>Synopsis</b>	<pre>Std_ReturnType Tcplp_TcpTransmit ( TcpIp_SocketIdType SocketId , const uint8 * DataPtr , uint32 AvailableLength , boolean ForceRetrieve );</pre>		
<b>Service ID</b>	0x13		
<b>Sync/Async</b>	Asynchronous		
<b>Reentrancy</b>	Reentrant for different SocketIds. Non reentrant for the same SocketId.		
<b>Parameters (in)</b>	SocketId	Socket identifier of the related local socket resource.	
	DataPtr	Pointer to a linear buffer of AvailableLength bytes containing the data to be transmitted. In case DataPtr is a NULL_PTR, Tcplp shall retrieve data from SoAd via callback SoAd_CopyTxData().	
	AvailableLength	Available data for transmission in bytes.	
	ForceRetrieve	This parameter is only valid if DataPtr is a NULL_PTR. Indicates how the TCP/IP stack retrieves data from SoAd if DataPtr is a NULL_PTR. TRUE: the whole data indicated by availableLength shall be retrieved from the upper layer via one or multiple SoAd_CopyTxData() calls within the context of this transmit function. FALSE: The TCP/IP stack may retrieve up to availableLength data from the upper layer. It is allowed to retrieve less than availableLength bytes. Note: Not retrieved data will be provided by SoAd with the next call to Tcplp_TcpTransmit (along with new data if available).	
<b>Return Value</b>	Result of operation		
	E_OK	The request has been accepted	
	E_NOT_OK	The request has not been accepted, e.g. due to a lack of buffer space or the socket is not connected.	



<b>Description</b>	This service requests transmission of data via TCP to a remote node. The transmission of the data is decoupled. Note: The TCP segment(s) are sent dependent on runtime factors (e.g. receive window) and configuration parameter (e.g. Nagle algorithm).
--------------------	--

#### 5.8.3.3.35. Tcplp\_UdpTransmit

<b>Purpose</b>	Requests to transmit data to a remote destination via UDP protocol.	
<b>Synopsis</b>	<code>TcpIp_ReturnType Tcplp_UdpTransmit ( TcpIp_SocketIdType socketId , const uint8 * dataPtr , const TcpIp_SockAddrType * remoteAddrPtr , uint16 totalLength );</code>	
<b>Service ID</b>	0x12	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SocketId	
<b>Parameters (in)</b>	socketId	identifying the local socket resource.
	dataPtr	In case DataPtr is a NULL_PTR, Tcplp shall retrieve data from SoAd via callback SoAd_CopyTxData(). In case DataPtr is a valid pointer Tcplp must not retrieve data via SoAd_CopyTxData(). Then DataPtr points to a linear buffer of TotalLength bytes. Tcplp shall send the whole data indicated by DataPtr and TotalLength within the context of this transmit function.
	remoteAddrPtr	IP address and port of the remote host to transmit to.
	totalLength	Total length of the data to be transmitted.
	<b>Return Value</b>	
<b>Return Value</b>	Result of operation	
	E_OK	UDP message transmission was successful.
	E_NOT_OK	UDP message transmission failed.
	TCPIP_E_PHYS_ADDR_MISS	UDP message could not be sent because of an ARP cache miss, ARP request has been sent and upper layer may retry transmission by calling this function later again.
<b>Description</b>	This service transmits UDP data to a remote node. The transmission of the data is immediately performed with this function call by forwarding it to Ethlf. The parame-	



ter TotalLength indicates the payload size of the UDP datagram which shall be sent by the TCP/IP stack within the context of this transmit function call. If DataPtr is a NULL\_PTR the payload must be retrieved via SoAd\_CopyTxData(). If DataPtr is a valid pointer, it points to a linear buffer of TotalLength bytes containing the payload. If the socket has not been bound to a local resource via a previous call to [Tcplp\\_Bind\(\)](#) the local IP address and port used for transmission is selected by the Tcplp stack.

## 5.8.4. Integration notes

### 5.8.4.1. Exclusive areas

This section describes the exclusive areas used by the TcpIp module.

#### 5.8.4.1.1. SCHM\_TCPIP\_EXCLUSIVE\_AREA\_0

<b>Protected data structures</b>	All Tcplp socket 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 Mapping exclusive areas in the basic software modules in the Integration notes section for details.

#### 5.8.4.1.2. SCHM\_TCPIP\_EXCLUSIVE\_AREA\_1

<b>Protected data structures</b>	All IPv4 ARP table data that shall be protected from mutual access.
<b>Recommended locking mechanism</b>	This exclusive area must be protected by a locking mechanism if preemptions between API functions <code>TcpIp_Mainfunction()</code> , <code>TcpIp_UdpTransmit()</code> , <code>TcpIp_IcmpTransmit()</code> and <code>TcpIp_RxIndication()</code> are possible. 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.



#### 5.8.4.1.3. SCHM\_TCPIP\_EXCLUSIVE\_AREA\_2

<b>Protected data structures</b>	All state machine data that shall be protected from mutual access.
<b>Recommended locking mechanism</b>	This exclusive area must be protected by a locking mechanism if preemptions between API functions <code>TcpIp_Mainfunction()</code> , <code>TcpIp_RequestIpAddrAssignment()</code> and <code>TcpIp_ReleaseIpAddrAssignment()</code> are possible. 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.

#### 5.8.4.1.4. SCHM\_TCPIP\_EXCLUSIVE\_AREA\_3

<b>Protected data structures</b>	All state machine locking data that shall be protected from mutual access.
<b>Recommended locking mechanism</b>	This exclusive area must be protected by a locking mechanism if API function <code>TcpIp_Mainfunction()</code> is able to preempt API functions <code>TcpIp_UdpTransmit()</code> or <code>TcpIp_IcmpTransmit()</code> . 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.

#### 5.8.4.2. Production errors

Production errors are not reported by the `TcpIp` module.

#### 5.8.4.3. 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:

<b>Memory section</b>
-----------------------



VAR\_CLEARED\_UNSPECIFIED

CODE

VAR\_INIT\_32

CONFIG\_DATA\_UNSPECIFIED

CONST\_UNSPECIFIED

VAR\_INIT\_8

VAR\_CLEARED\_16

CONST\_8

CONST\_16

CONST\_32

VAR\_CLEARED\_32

VAR\_CLEARED\_8

VAR\_INIT\_UNSPECIFIED

VAR\_INIT\_16

#### 5.8.4.4. 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.

##### 5.8.4.4.1. Tcplp.EB\_INTREQ\_Tcplp\_0001

<b>Description</b>	The reinitialization process shall not interrupt other module functions. If reinitialization of the module is required, the call of Tcplp_Init() shall not interrupt other module functions.
<b>Rationale</b>	The reinitialization process resets all internal variables. Continuing an interrupted module function after reinitialization can lead to undefined module behavior.

##### 5.8.4.4.2. Tcplp.EB\_INTREQ\_Tcplp\_0002

<b>Description</b>	Tcplp_Init() shall not be preempted by any other module API calls. It needs to be ensured that the function call Tcplp_Init() is not preempted by any other module API calls.
--------------------	---



<b>Rationale</b>	During the call of Tcplp_Init() global variables and pointers get initialized. It is easy for the integrator to avoid this preemption, thus no data protection mechanism has been implemented for function Tcplp_Init().
------------------	--

#### 5.8.4.4.3. Tcplp.EB\_INTREQ\_Tcplp\_0003

<b>Description</b>	Tcplp_MainFunction must not preempt or be preempted by EthIf_MainFunctionRx(). The integrator must assure that EthIf_MainFunctionRx() does not preempt Tcplp_MainFunction(). The integrator also must assure that Tcplp_MainFunction() does not preempt EthIf_MainFunctionRx().
<b>Rationale</b>	This limitation reduces code size and execution time by eliminating the need for extensive use of exclusive areas.

#### 5.8.4.4.4. Tcplp.EB\_INTREQ\_Tcplp\_0004

<b>Description</b>	Eth and EthIf must not use receive or transmit interrupts. The Integrator must assure that neither Eth nor EthIf use interrupts. That is, EthIfEnableRxInterrupt, EthCtrlEnableRxInterrupt, EthIfEnableTxInterrupt and EthCtrlEnableTxInterrupt shall be set to false. This prevents Tcplp_RxIndication() from interrupting Tcplp_MainFunction().
<b>Rationale</b>	This limitation reduces code size and execution time by eliminating the need for extensive use of exclusive areas.

#### 5.8.4.4.5. Tcplp.EB\_INTREQ\_Tcplp\_0005

<b>Description</b>	Tcplp_MainFunctionTx() shall have the same preemption constraints as Tcplp_MainFunction().
--------------------	--

#### 5.8.4.4.6. Tcplp.EB\_INTREQ\_Tcplp\_0006

<b>Description</b>	The Integrator must assure that the hardware buffer returned by EthIf_provideTxBuffer is always greater than or equal to the requested buffer size if EthIf_provideTxBuffer returns BUFREQ_OK. If requirement SWS_Eth_00079 is implemented by the Ethernet driver the buffer is always greater or equal when EthIf_provideTxBuffer returns BUFREQ_OK. (If a buffer is requested with Eth_ProvideTxBuffer that is larger than the available buffer length, the buffer shall not be locked but return the available length and BUFREQ_E_OVFL.)
--------------------	--



#### 5.8.4.4.7. Tcplp.EB\_INTREQ\_Tcplp\_0007

Description	The Integrator must assure that the Crypto modules are initialized before the initialization of Tcplp.
-------------	--

#### 5.8.4.4.8. Tcplp.EB\_INTREQ\_Tcplp\_0008

Description	The Integrator must assure that StbM module is initialized before the initialization of Tcplp.
-------------	--

#### 5.8.4.4.9. Tcplp.EB\_INTREQ.uisnSecretKeyLen

Description	Key used to generate the pseudo random value used in unpredictable TCP sequence numbers shall be 128 bits long.
-------------	---

#### 5.8.4.4.10. Tcplp.EB\_INTREQ.secretHashOutLen

Description	The hash value produced by Csm_MacGenerate used to create the TCP sequence number (unpredictable sequence numbers and syn cookies) shall be 4 octets long.
-------------	--

#### 5.8.4.4.11. Tcplp.EB\_INTREQ.storingIpAddresses

Description	BswM and NvM should be initialized before Tcplp when NvM storing of Ip addresses is used. BswM will read the addresses during startup which will then be available for Tcplp to use in internal memory. ComM_RequestComMode(NO_COMMUNICATION) needs to be called to properly turn Tcplp off and signalize NvM that writing can be done. BswM will write the addresses during shutdown which will then be copied from internal memory to NvM.
-------------	--

#### 5.8.4.4.12. Tcplp.EB\_INTREQ.ASILD\_EthIf\_ProvideTxBuffer

Description	If the function EthIf_ProvideTxBuffer returns BUFREQ_OK then Eth layer has already ensured that the buffer provided by argument BufPtr is larger or equal as the argument LenBytePtr.
-------------	---



#### 5.8.4.4.13. Tcplp.EB\_INTREQ.ASILD\_EthIf\_GetPhysAddr

<b>Description</b>	Upon calling the function EthIf_GetPhysAddr the Eth layer SHALL ensure that it does not write more than 6 bytes beyond argument PhysAddrPtr (i.e. there is no write access to PhysAddrPtr[6] or later).
--------------------	---

#### 5.8.4.4.14. Tcplp.EB\_INTREQ.ASILD\_Tcplp\_GetRemotePhysAddr

<b>Description</b>	The 3rd argument (PhysAddrPtr) passed to Tcplp_GetRemotePhysAddr has to point to at least 6 bytes of allocated memory.
--------------------	--

## 5.9. UdpNm

### 5.9.1. Configuration parameters

<b>Containers included</b>		
<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="#">PublishedInformation</a>	1..1	<b>Label:</b> EB Published Information Additional published parameters not covered by CommonPublishedInformation container.
<a href="#">UdpNmGeneral</a>	1..1	
<a href="#">UdpNmGlobalConfig</a>	1..1	This container contains all global configuration parameters of UDP NM configured from the NM Module perspective.
<a href="#">UdpNmDefensiveProgramming</a>	1..1	<b>Label:</b> Defensive Programming Options Parameters for defensive programming

<b>Parameters included</b>	
<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>	VariantPostBuild
<b>Range</b>	VariantPostBuild
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild

### 5.9.1.1. 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>	3
<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>	3
<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>	2
<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>	9
<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>	33
<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	

### 5.9.1.2. 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 UdpNm 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

### 5.9.1.3. UdpNmGeneral

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">UdpNmMultiCoreSupport</a>	1..1
<a href="#">UdpNmPnSupported</a>	1..1
<a href="#">UdpNmASR412APICompatibility</a>	1..1
<a href="#">UdpNmRelocatablePbcfgEnable</a>	1..1
<a href="#">UdpNmMaxPn</a>	0..1

<b>Parameter Name</b>	<b>UdpNmMultiCoreSupport</b>
<b>Label</b>	UdpNm 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>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH	

<b>Parameter Name</b>	<b>UdpNmPnSupported</b>	
<b>Label</b>	Support for Partial Network Cluster (PNC)	
<b>Description</b>	<p>Enables or disables support of partial networking.</p> <ul style="list-style-type: none"> <li>▶ False: Partial Networking is disabled</li> <li>▶ True: Partial Networking is 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>UdpNmASR412APICompatibility</b>	
<b>Label</b>	UdpNm_SoAdIfRxIndication ASR412 Compatibility	
<b>Description</b>	<p>UdpNm_SoAdIfRxIndication compatibility with ASR 4.1.2.</p> <ul style="list-style-type: none"> <li>▶ False: UdpNm_SoAdIfRxIndication ASR 4.1.3 compatibility</li> <li>▶ True: UdpNm_SoAdIfRxIndication ASR 4.1.2 compatibility</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>UdpNmRelocatablePbcfgEnable</b>	
<b>Label</b>	UdpNmRelocatablePbcfgEnable	
<b>Description</b>	Enables/disables support for relocatable postbuild 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>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	UdpNmMaxPn
<b>Label</b>	UdpNmMaxPn
<b>Description</b>	The maximum number of Partial Networking Clusters that can be configured.
<b>Multiplicity</b>	0..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.9.1.4. UdpNmGlobalConfig

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">UdpNmChannelConfig</a>	1..n	<b>Label:</b> Channel Configuration This container holds the channel specific configuration parameter of the UdpNm.
<a href="#">UdpNmPnInfo</a>	0..1	PN information configuration.

<b>Parameters included</b>	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">UdpNmActiveWakeupBitEnabled</a>	1..1
<a href="#">UdpNmBusSynchronizationEnabled</a>	1..1
<a href="#">UdpNmComControlEnabled</a>	1..1
<a href="#">UdpNmNodIdCallback</a>	0..1
<a href="#">UdpNmNodIdCallbackHeader</a>	1..1
<a href="#">UdpNmComUserDataSupport</a>	1..1
<a href="#">UdpNmCoordinatorSyncSupport</a>	1..1
<a href="#">UdpNmDevErrorDetect</a>	1..1
<a href="#">UdpNmImmediateRestartEnabled</a>	1..1
<a href="#">UdpNmMainFunctionPeriod</a>	1..1

**Parameters included**

<a href="#">UdpNmNumberOfChannels</a>	1..1
<a href="#">UdpNmPostBuildRamSize</a>	1..1
<a href="#">UdpNmPassiveModeEnabled</a>	1..1
<a href="#">UdpNmPduRxIndicationEnabled</a>	1..1
<a href="#">UdpNmPnPnEiraCalcEnabled</a>	0..1
<a href="#">UdpNmPnPnResetTime</a>	0..1
<a href="#">UdpNmRemoteSleepIndEnabled</a>	1..1
<a href="#">UdpNmStateChangeIndEnabled</a>	1..1
<a href="#">UdpNmUserDataEnabled</a>	1..1
<a href="#">UdpNmVersionInfoApi</a>	1..1
<a href="#">UdpNmPnPnEiraRxNSduRef</a>	0..1

**Parameter Name****UdpNmActiveWakeupBitEnabled****Label**

Active Wakeup Bit Enable

**Description**

Enables/Disables the handling of the Active Wakeup Bit in the UdpNm module.

**Multiplicity**

1..1

**Type**

BOOLEAN

**Default value**

false

**Configuration class****VariantPostBuild:** VariantPostBuild**Origin**

Elektrobit Automotive GmbH

**Parameter Name****UdpNmBusSynchronizationEnabled****Label**

Bus Synchronization

**Description**

Pre-processor switch for enabling bus synchronization support. This feature is required for gateway nodes only. It must not be defined if UDPNM\_PASSIVE\_MODE\_ENABLED is defined. This parameter shall be derived from NM\_BUS\_SYNCHRONIZATION\_ENABLED.

The following API function is provided:

- ▶ `UdpNm_RequestBusSynchronization()`

Dependencies:

- ▶ The value of this parameter has to be synchronized with the value of respective parameter in the Nm module.



	▶ Passive Mode must be 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>UdpNmComControlEnabled</b>
<b>Label</b>	Communication Control
<b>Description</b>	<p>Pre-processor switch for enabling the Communication Control support. This parameter shall be derived from NM_COM_CONTROL_ENABLED.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ The value of this parameter has to be synchronized with the value of respective parameter in the Nm module.</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>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>UdpNmNodeIdCallback</b>
<b>Label</b>	UdpNm Node Id callback
<b>Description</b>	Name of the callback function to be called if UdpNmNodeIdCallback is enabled.
<b>Multiplicity</b>	0..1
<b>Type</b>	FUNCTION-NAME
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>UdpNmNodeIdCallbackHeader</b>
<b>Label</b>	Node Id callback header
<b>Description</b>	The name of a header file that will be included to obtain the external declaration of the callback function.
	Dependencies:



	▶ This parameter is only available if UdpNmNodeIdCallback is enabled.
<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>UdpNmComUserDataSupport</b>
<b>Description</b>	Enable/disable the user data support.
<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>UdpNmCoordinatorSyncSupport</b>
<b>Description</b>	<i>The functionality related to this parameter is not supported by the current implementation.</i>  Enables/disables the coordinator synchronisation support.
<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>UdpNmDevErrorDetect</b>
<b>Label</b>	Enable Development Error Detection
<b>Description</b>	Pre-processor switch for enabling development error detection support.
<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>UdpNmImmediateRestartEnabled</b>
-----------------------	-------------------------------------



<b>Label</b>	Immediate Restart
<b>Description</b>	<p>Enabling the asynchronous transmission of a NM PDU upon bus communication request in Prepare-Bus-Sleep mode.</p> <p>Dependencies:</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>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>UdpNmMainFunctionPeriod</b>
<b>Label</b>	Main Function Period [s]
<b>Description</b>	Call cycle in seconds of UdpNm_MainFunction
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.001
<b>Range</b>	<=0.255 >=0.001
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>UdpNmNumberOfChannels</b>
<b>Label</b>	Number Of Channels
<b>Description</b>	Maximum number of Udp NM channels allowed within one ECU.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	1
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>UdpNmPostBuildRamSize</b>
<b>Label</b>	UdpNmPostBuildRamSize



<b>Description</b>	Number of bytes for TX and RX buffers
	Value should be set as: the sum of the first RxPdu lengths on each channel multiplied with 2 (in case passive mode is disabled). Size should be big enough to hold eventual changes of PDU lengths at postbuild time
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	96
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>UdpNmPassiveModeEnabled</b>
<b>Label</b>	Passive Mode
<b>Description</b>	Pre-processor switch for enabling support of the Passive Mode. This parameter shall be derived from NM_PASSIVE_MODE_ENABLED.  In passive mode, the UdpNm will not be able to wake up the bus and will not send NM messages. It will only listen to the NM messages and silently monitor the bus.
<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>UdpNmPduRxIndicationEnabled</b>
<b>Label</b>	PDU Receive Indication
<b>Description</b>	Pre-processor switch for enabling the PDU Rx Indication. This parameter shall be derived from NM_PDU_RX_INDICATION_ENABLED. If a NM message is received the function  Dependencies:  ▶ The value of this parameter has to be synchronized with the value of respective parameter in the Nm module.
<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>UdpNmPnEiraCalcEnabled</b>
<b>Description</b>	Specifies if UdpNm calculates the PN request information for internal and external requests(EIRA). <ul style="list-style-type: none"> <li>▶ true: PN request are calculated</li> <li>▶ false: PN request are not calculated</li> </ul>
<b>Multiplicity</b>	0..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>PreCompile:</b>
<b>Origin</b>	AUTOSAR_ECU

<b>Parameter Name</b>	<b>UdpNmPnResetTime</b>
<b>Description</b>	Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA. The value shall be the same for every channel.
<b>Multiplicity</b>	0..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.01
<b>Range</b>	<=65.535 >=0.0010
<b>Configuration class</b>	<b>PreCompile:</b>
<b>Origin</b>	AUTOSAR_ECU

<b>Parameter Name</b>	<b>UdpNmRemoteSleepIndEnabled</b>
<b>Label</b>	Remote Sleep Indication
<b>Description</b>	Pre-processor switch for enabling remote sleep indication support. This feature is required for gateway nodes only. It must not be defined if UDPNM_PASSIVE_MODE_ENABLED is defined. This parameter shall be derived from NM_REMOTE_SLEEP_IND_ENABLED.  Therefore the following API is provided: <ul style="list-style-type: none"> <li>▶ <code>UdpNm_CheckRemoteSleepIndication()</code></li> </ul>



	<p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ The value of this parameter has to be synchronized with the value of respective parameter in the Nm module.</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>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>UdpNmStateChangeIndEnabled</b>
<b>Label</b>	State Change Indication
<b>Description</b>	Pre-processor switch for enabling the UDP NM state change notification. This parameter shall be derived from NM_STATE_CHANGE_IND_ENABLED.
<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>UdpNmUserDataEnabled</b>
<b>Label</b>	User Data
<b>Description</b>	<p>Pre-processor switch for enabling user data support. This parameter shall be derived from NM_USER_DATA_ENABLED.</p> <p>Therefore the following API functions are provided:</p> <ul style="list-style-type: none"> <li>▶ UdpNm_GetUserData()</li> <li>▶ UdpNm_SetUserData() (<i>Only if Passive Mode Support is disabled</i>)</li> </ul> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ The value of this parameter has to be synchronized with the value of respective parameter in the Nm module.</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>UdpNmVersionInfoApi</b>
<b>Label</b>	Enable Version Info API
<b>Description</b>	Pre-processor switch for enabling version info API support. Provide API function for retrieving version information: ▶ UdpNm_GetVersionInfo ()
<b>Multiplicity</b>	1..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	AUTOSAR_ECU

<b>Parameter Name</b>	<b>UdpNmPnEiraRxNSduRef</b>
<b>Description</b>	Reference to a Pdu in the COM-Stack. Only one SduRef is required for UdpNm because the EIRA is the aggregation over all Ethernet Channels.
<b>Multiplicity</b>	0..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>PreCompile:</b>
<b>Origin</b>	AUTOSAR_ECU

### 5.9.1.5. UdpNmChannelConfig

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>
<a href="#">UdpNmRxPdu</a>	1..n	<b>Label:</b> Receive PDU ID and Reference  This container holds the <code>UdpNmRxPduId</code> and the <code>UdpNmRxPduRef</code> .
<a href="#">UdpNmTxPdu</a>	0..1	<b>Label:</b> Transmit PDU Reference  This container contains the <code>UdpNmTxConfirmationPduId</code> and the <code>UdpNmTxPduRef</code> .
<a href="#">UdpNmUserDataTxPdu</a>	0..1	<b>Label:</b> User Data Transmission PDU



### Containers included

		This optional container is used to configure the UserNm PDU. This container is only available if <code>UdpNmComUserDataSupport</code> is enabled.
<a href="#">UdpNmUserDataRxPdu</a>	0..1	<p><b>Label:</b> User Data Reception PDUs</p> <p>This optional container is used to configure the UserNm PDU. This container is only available if <code>UdpNmComUserDataSupport</code> is enabled.</p>

### Parameters included

Parameter name	Multiplicity
<a href="#">UdpNmNodeIdEnabled</a>	1..1
<a href="#">UdpNmRepeatMsgIndEnabled</a>	1..1
<a href="#">UdpNmNodeDetectionEnabled</a>	1..1
<a href="#">UdpNmAllNmMessagesKeepAwake</a>	1..1
<a href="#">UdpNmCarWakeUpBitPosition</a>	1..1
<a href="#">UdpNmCarWakeUpBytePosition</a>	1..1
<a href="#">UdpNmCarWakeUpFilterEnabled</a>	1..1
<a href="#">UdpNmCarWakeUpFilterNodeId</a>	1..1
<a href="#">UdpNmCarWakeUpRxEnabled</a>	1..1
<a href="#">UdpNmPnEnabled</a>	0..1
<a href="#">UdpNmPnEraCalcEnabled</a>	0..1
<a href="#">UdpNmPnHandleMultipleNetworkRequests</a>	0..1
<a href="#">UdpNmPnEraRxNSduRef</a>	0..1
<a href="#">UdpNmImmediateNmCycleTime</a>	1..1
<a href="#">UdpNmImmediateNmTransmissions</a>	1..1
<a href="#">UdpNmRetryFirstMessageRequest</a>	1..1
<a href="#">UdpNmMsgCycleOffset</a>	1..1
<a href="#">UdpNmMsgCycleTime</a>	1..1
<a href="#">UdpNmMsgTimeoutTime</a>	1..1
<a href="#">UdpNmNodeId</a>	1..1
<a href="#">UdpNmPduCbvPosition</a>	1..1
<a href="#">UdpNmPduNidPosition</a>	1..1
<a href="#">UdpNmRemoteSleepIndTime</a>	1..1

**Parameters included**

<a href="#">UdpNmRepeatMessageTime</a>	1..1
<a href="#">UdpNmTimeoutTime</a>	1..1
<a href="#">UdpNmWaitBusSleepTime</a>	1..1
<a href="#">UdpNmComMNetworkHandleRef</a>	1..1

<b>Parameter Name</b>	<b>UdpNmNodeIdEnabled</b>	
<b>Label</b>	Node Identifier	
<b>Description</b>	<p>Enable support for sending of Node Ids in NM messages and provide functions for retrieving the node identifier from the most recently received NM PDU and the local node identifier.</p> <p>Therefore the following API functions are provided:</p> <ul style="list-style-type: none"> <li>▶ <code>Nm_GetNodeIdentifier()</code></li> <li>▶ <code>Nm_GetLocalNodeIdentifier()</code></li> </ul> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ The value of this parameter has to be synchronized with the value of the respective parameter in the Nm module.</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>UdpNmRepeatMsgIndEnabled</b>	
<b>Label</b>	Repeat Message Indication	
<b>Description</b>	<p>Enable/disable the notification that a RepeatMessageRequest bit has been received. This parameter shall be derived from NM_REPEAT_MSG_IND_ENABLED.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ Node Detection 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_ECU	

<b>Parameter Name</b>	<b>UdpNmNodeDetectionEnabled</b>	
<b>Label</b>	Node Detection	
<b>Description</b>	<p>Pre-processor switch for enabling the node detection support. This parameter shall be derived from NM_NODE_DETECTION_ENABLED.</p> <p>This parameter shall only be enabled if UDPNM_NODE_ID_ENABLED is defined. If(UdpNmPduCbvPosition != UDPNM_PDU_OFF) then Equal(NmNodeDetectionEnabled) else Equal(False).</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>▶ UdpNm_RepeatMessageRequest ()</li> </ul> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ The value of this parameter has to be synchronized with the value of the respective parameter in the Nm module.</li> <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>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>UdpNmAllNmMessagesKeepAwake</b>	
<b>Label</b>	UdpNmAllNmMessagesKeepAwake	
<b>Description</b>	Specifies if UdpNm drops irrelevant NM PDUs	
<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>UdpNmCarWakeUpBitPosition</b>	
-----------------------	----------------------------------	--



<b>Description</b>	Specifies the Bit position of the CWU within the UdpNmCarWakeUpBytePosition.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Range</b>	<=7 >=0	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>UdpNmCarWakeUpBytePosition</b>	
<b>Description</b>	Specifies the Byte position of the CWU within the NM-Message.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	2	
<b>Configuration class</b>	<b>PreCompile:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>UdpNmCarWakeUpFilterEnabled</b>	
<b>Description</b>	If CWU filtering is supported, only the CWU bit within the NM PDU message with source node identifier UdpNmCarWakeUpFilterNodId is considered as CWU request.	
<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_ECU	

<b>Parameter Name</b>	<b>UdpNmCarWakeUpFilterNodId</b>	
<b>Description</b>	Source node identifier for CWU filtering. If CWU filtering is supported, only the CWU bit within the NM message with source node identifier UdpNmCarWakeUpFilterNodId is considered as CWU request.	
<b>Multiplicity</b>	1..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>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>UdpNmCarWakeUpRxEnabled</b>	
<b>Description</b>	Enables or disables support of CarWakeUp bit evaluation in received NM messages.	
<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>UdpNmPnEnabled</b>	
<b>Description</b>	Enables or disables support of partial networking.  ▶ true: Partial networking supported ▶ false: Partial networking not supported	
<b>Multiplicity</b>	0..1	
<b>Type</b>	BOOLEAN	
<b>Default value</b>	false	
<b>Configuration class</b>	<b>PostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>UdpNmPnEraCalcEnabled</b>	
<b>Description</b>	Specifies if UdpNm calculates the PN request information for external requests.(ERA)	
<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>UdpNmPnHandleMultipleNetworkRequests</b>	
-----------------------	---	--



<b>Description</b>	<ul style="list-style-type: none"> <li>▶ true: UdpNm_NetworkRequest triggers a change from NO to RM.</li> <li>▶ false: UdpNm_NetworkRequest is ignored in NO.</li> </ul> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ Support for Partial Networks must be enabled for this channel</li> </ul>
<b>Multiplicity</b>	0..1
<b>Type</b>	BOOLEAN
<b>Default value</b>	false
<b>Configuration class</b>	<b>PostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>UdpNmPnEraRxNSduRef</b>
<b>Description</b>	Reference to a Pdu in the COM-Stack. The SduRef is required for every UdpNm Channel, because ERA is reported per channel.
<b>Multiplicity</b>	0..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>UdpNmImmediateNmCycleTime</b>
<b>Label</b>	Immediate NM PDU cycle time
<b>Description</b>	Defines the immediate NM PDU cycle time in seconds which is used for UdpNmImmediateNmTransmissions NM PDU transmissions. This parameter is only valid if UdpNmImmediateNmTransmissions is greater one  Dependencies: <ul style="list-style-type: none"> <li>▶ This parameter is active only if UdpNmImmediateNmTransmissions greater than one</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.001
<b>Configuration class</b>	<b>PreCompile:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>UdpNmImmediateNmTransmissions</b>
-----------------------	--------------------------------------



<b>Label</b>	Number of immediate NM PDUs
<b>Description</b>	<p>Defines the number of immediate NM PDUs which shall be transmitted. If the value is zero no immediate NM PDUs are transmitted. The cycle time of immediate NM PDUs is defined by <code>UdpNmImmediateNmCycleTime</code>.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ <code>UdpNmImmediateNmCycleTime</code> is active only if this parameter greater than zero</li> </ul>
<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_ECU

<b>Parameter Name</b>	<b>UdpNmRetryFirstMessageRequest</b>
<b>Label</b>	UdpNmRetryFirstMessageRequest
<b>Description</b>	If <code>UdpNmRetryFirstMessageRequest</code> is enabled and if the first transmit request after transition into NetworkMode is not accepted by SoAd, the message request shall be repeated in the next main function until one transmit request is accepted by SoAd. Note: This feature can be used in case of partial network wakeup filter to avoid a blocking of all messages in case of passive start-up and first message request is not accepted by SoAd due to EthSM could not enable transmission path fast enough (e.g. in case of asynchronous transceiver handling).
<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>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>UdpNmMsgCycleOffset</b>
<b>Label</b>	Message Cycle Offset [s]
<b>Description</b>	<p>Time offset in the periodic transmission node. It determines the start delay of the transmission. &amp;lt; UDPNM_MSG_CYCLE_TIME This parameter is only valid if UDPNM_PASSIVE_MODE_ENABLED is disabled.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ If Passive Mode is enabled this parameter is ignored.</li> </ul>



	<ul style="list-style-type: none"> <li>▶ The Message Cycle Offset must be smaller than the Message Cycle Time.</li> <li>▶ The value must be multiple of the Main Function Period.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.001
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC
<b>Parameter Name</b>	<b>UdpNmMsgCycleTime</b>
<b>Label</b>	Message Cycle Time [s]
<b>Description</b>	<p>Period of a NM-message. It determines the periodic rate and is the basis for transmit scheduling. NM_TIMEOUT_TIME = n * UDPNM_MSG_CYCLE_TIME</p> <p>This parameter is only valid if UDPNM_PASSIVE_MODE_ENABLED is disabled.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ If Passive Mode is enabled this parameter is ignored.</li> <li>▶ The value must be multiple of the Main Function Period.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	FLOAT
<b>Default value</b>	0.002
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC
<b>Parameter Name</b>	<b>UdpNmMsgTimeoutTime</b>
<b>Label</b>	Message Timeout Time [s]
<b>Description</b>	<p>Transmission Timout of NM-message. If there is no transmission confirmation by the UDP Interface within this timeout, the UDPNM module shall gibe an error notification. This parameter is only valid if UDPNM_PASSIVE_MODE_ENABLED is disabled. UDPNM_MSG_TIMEOUT_TIME should be a multiple of UDPNM_MSG_CYCLE_TIME.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ If Passive Mode is enabled this parameter is ignored.</li> <li>▶ Value must be a multiple of the Main Function Period.</li> <li>▶ Value must be less than the Message Cycle Time.</li> </ul>
<b>Multiplicity</b>	1..1



Type	FLOAT
Default value	0.002
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmNodeId
Label	Node Identifier
Description	<p>Node identifier of local node. This parameter is only valid if UDPNM_PASSIVE_MODE_ENABLED is set to OFF and UDPNM_NODE_DETECTION_ENABLED is set to ON.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ This parameter is only valid if UdpNmPassiveModeEnabled = False</li> <li>▶ If the Node Identifier Position is set to UDPNM_PDU_OFF this parameter is ignored.</li> </ul>
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<p>&lt;=255</p> <p>&gt;=0</p>
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmPduCbvPosition
Label	PDU Control Bit Vector Position
Description	<p>Defines the position of the control bit vector within the NM PACKET. The value of the parameter represents the location of the control bit vector in the NM PACKET (UDPNM_PDU_BYTE_0 means byte 0, UDPNM_PDU_BYTE_1 means byte 1, UDPNM_PDU_OFF means the control bit vector is not part of the NM PACKET) See also UDPNM_PDU_NID_POSITION if (UDPNM_PDU_CBV_POSITION != UDPNM_PDU_OFF &amp;amp; UDPNM_PDU_NID_POSITION != UDPNM_PDU_OFF) then UDPNM_PDU_CBV_POSITION != UDPNM_PDU_NID_POSITION if (UDPNM_PDU_CBV_POSITION != UDPNM_PDU_OFF &amp;amp; UDPNM_PDU_NID_POSITION == UDPNM_PDU_OFF) then UDPNM_PDU_CBV_POSITION = UDPNM_PDU_BYTE0</p> <p>Dependencies:</p>



	<ul style="list-style-type: none"> <li>▶ If Node Detection support is disabled this parameter is ignored.</li> <li>▶ The Control Bit Vector must not occupy the same byte as the Node Id.</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	UDPNM_PDU_BYTE_1
<b>Range</b>	UDPNM_PDU_BYTE_0 UDPNM_PDU_BYTE_1 UDPNM_PDU_OFF
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU

<b>Parameter Name</b>	<b>UdpNmPduNidPosition</b>
<b>Label</b>	PDU Node Identifier Position
<b>Description</b>	<p>Defines the position of the source node identifier within the NM PACKET. ImplementationType: UdpNm_PduPositionType The value of the parameter represents the location of the source node identifier in the NM PACKET (UDPNM_PDU_BYTE_0 means byte 0, UDPNM_PDU_BYTE_1 means byte 1, UDPNM_PDU_OFF means source node identifier is not part of the NM PACKET) See also UDPNM_PDU_CBV_POSITION if (UDPNM_PDU_NID_POSITION != UDPNM_PDU_OFF &amp;&amp; UDPNM_PDU_CBV_POSITION != UDPNM_PDU_OFF) then UDPNM_PDU_NID_POSITION != UDPNM_PDU_CBV_POSITION if (UDPNM_PDU_NID_POSITION != UDPNM_PDU_OFF &amp;&amp; UDPNM_PDU_CBV_POSITION == UDPNM_PDU_OFF) then UDPNM_PDU_IND_POSITION = UDPNM_PDU_BYTE0</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ If Node Id support is disabled this parameter is ignored.</li> <li>▶ The Node Id must not occupy the same byte as the Control Bit Vector</li> </ul>
<b>Multiplicity</b>	1..1
<b>Type</b>	ENUMERATION
<b>Default value</b>	UDPNM_PDU_BYTE_0
<b>Range</b>	UDPNM_PDU_BYTE_0 UDPNM_PDU_BYTE_1 UDPNM_PDU_OFF
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild



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

<b>Parameter Name</b>	<b>UdpNmRemoteSleepIndTime</b>	
<b>Label</b>	Remote Sleep Indication Time [s]	
<b>Description</b>	<p>Timeout for Remote Sleep Indication. It defines the time in [s] how long it shall take to recognize that all other nodes are ready to sleep. Typically it should be equal to: <math>n * \text{UDPNM\_MSG\_CYCLE\_TIME}</math>, where <math>n</math> denotes the number of NM packets that are normally sent before Remote Sleep Indication is detected. The value of <math>n</math> decremented by one determines the amount of lost NM packets that can be tolerated by the Remote Sleep Indication procedure.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ <math>\text{UdpNmRemoteSleepIndTime} \geq \text{UdpNmMsgCycleTime}</math></li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.001	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

<b>Parameter Name</b>	<b>UdpNmRepeatMessageTime</b>	
<b>Label</b>	Repeat Message Time [s]	
<b>Description</b>	<p>Timeout for Repeat Message State. It defines the time in [s] how long the NM shall stay in the Repeat Message State. Typically it should be equal to: <math>n * \text{UDPNM\_MSG\_CYCLE\_TIME}</math>, where <math>n</math> denotes the number of NM packets that are normally sent in the Repeat Message State. The value of <math>n</math> decremented by one determines the amount of lost NM packets that can be tolerated by the node detection procedure. The value 0 denotes that no Repeat Message State is configured. It means that Repeat Message State is transient what implicates that it is left immediately after entrance and in result no start-up stability is guaranteed and no node detection procedure is possible.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ If Passive Mode is enabled this parameter is ignored.</li> <li>▶ Value must be a multiple of the Main Function Period.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.001	



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>UdpNmTimeoutTime</b>	
<b>Label</b>	Timeout Time [s]	
<b>Description</b>	<p>Network Timeout for NM packets. It denotes the time in [s] how long the NM shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place. It shall be equal for all nodes in the cluster. It shall be greater than UDPNM_MSG_CYCLE_TIME. Typically, it should be equal to: <math>x * UDPNM\_MSG\_CYCLE\_TIME</math>, where n denotes the number of NM PACKET cycle times in the Ready Sleep State before transition into the Bus-Sleep Mode is initiated. The value of n decremented by one determines the amount of lost NM packets that can be tolerated by the coordination algorithm.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ Value must be a multiple of the Main Function Period.</li> <li>▶ The UdpNmTimeoutTime must be greater than UdpNmPnResetTime.</li> <li>▶ The Timeout Time must be greater than Message cycle time.</li> <li>▶ The Timeout Time must be a multiple of the Main Function Period.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	
<b>Default value</b>	0.004	
<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU	

<b>Parameter Name</b>	<b>UdpNmWaitBusSleepTime</b>	
<b>Label</b>	Wait Bus Sleep Time [s]	
<b>Description</b>	<p>Timeout for bus calm down phase. It denotes the time in [s] how long the NM shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place. It shall be equal for all nodes in the cluster. It shall be long enough to empty all Tx buffer empty.</p> <p>Dependencies:</p> <ul style="list-style-type: none"> <li>▶ Value must be a multiple of the Main Function Period.</li> </ul>	
<b>Multiplicity</b>	1..1	
<b>Type</b>	FLOAT	



<b>Default value</b>	0.004
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>UdpNmComMNetworkHandleRef</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>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.9.1.6. UdpNmRxPdu

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

<b>Parameter Name</b>	<b>UdpNmRxPduld</b>
<b>Label</b>	Receive PDU ID
<b>Description</b>	ID of the RxPdu that will be used by a RxIndication of the lower layer.
<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>UdpNmRxPduRef</b>
<b>Label</b>	Receive PDU Reference
<b>Description</b>	The reference to a PDU in the global PDU structure described in the AUTOSAR ECU Configuration Specification. This reference will be used by the UdpNm module to derive the PDU Id.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE



<b>Configuration class</b>	<b>VariantPostBuild:</b>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

### 5.9.1.7. UdpNmTxPdu

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

<b>Parameter Name</b>	<b>UdpNmTxConfirmationPduld</b>
<b>Description</b>	Id of the TxPdu that will be used by a TxConfirmation from the lower layer.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	0
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>UdpNmTxPduRef</b>
<b>Label</b>	Transmit PDU Reference
<b>Description</b>	The reference to a PDU in the global PDU structure described in the AUTOSAR ECU Configuration Specification. This reference will be used by the UdpNm module to derive the PDU Id.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b>
<b>Origin</b>	AUTOSAR_ECUC

### 5.9.1.8. UdpNmUserDataTxPdu

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



#### Parameters included

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

<b>Parameter Name</b>	<b>UdpNmTxUserDataPduld</b>
<b>Description</b>	This parameter defines the Handle ID of the NM User Data I-PDU.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU

<b>Parameter Name</b>	<b>UdpNmTxUserDataPduRef</b>
<b>Description</b>	Reference to the NM User Data I-PDU in the global PDU collection.
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECU

#### 5.9.1.9. UdpNmUserDataRxPdu

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

<b>Parameter Name</b>	<b>UdpNmRxUserDataPduRef</b>
<b>Description</b>	Reference to the Rx NM User Data I-PDU in the global PDU collection
<b>Multiplicity</b>	1..1
<b>Type</b>	REFERENCE
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

#### 5.9.1.10. UdpNmPnInfo

<b>Containers included</b>		
<b>Container name</b>	<b>Multiplicity</b>	<b>Description</b>



#### Containers included

<a href="#">UdpNmPnFilterMaskByte</a>	0..7	PN information configuration.
---------------------------------------	------	-------------------------------

#### Parameters included

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

<b>Parameter Name</b>	<b>UdpNmPnInfoLength</b>
<b>Description</b>	Specifies the length of the PN request information in the NM message.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	1
<b>Range</b>	<=7 >=1
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

<b>Parameter Name</b>	<b>UdpNmPnInfoOffset</b>
<b>Description</b>	Specifies the offset of the PN request information in the NM message.
<b>Multiplicity</b>	1..1
<b>Type</b>	INTEGER
<b>Default value</b>	1
<b>Range</b>	<=31 >=1
<b>Configuration class</b>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC

#### 5.9.1.11. UdpNmPnFilterMaskByte

##### Parameters included

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



#### Parameters included

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

Parameter Name	<a href="#">UdpNmPnFilterMaskByteIndex</a>	
<b>Description</b>	Index of the filter mask byte. Specifies the position within the filter mask	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Configuration class</b>	<a href="#">VariantPostBuild:</a>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

Parameter Name	<a href="#">UdpNmPnFilterMaskByteValue</a>	
<b>Description</b>	Parameter to configure the filter mask byte.	
<b>Multiplicity</b>	1..1	
<b>Type</b>	INTEGER	
<b>Default value</b>	0	
<b>Range</b>	<=255 >=0	
<b>Configuration class</b>	<a href="#">VariantPostBuild:</a>	VariantPostBuild
<b>Origin</b>	AUTOSAR_ECUC	

#### 5.9.1.12. UdpNmDefensiveProgramming

Parameters included	
<b>Parameter name</b>	<b>Multiplicity</b>
<a href="#">UdpNmDefProgEnabled</a>	1..1
<a href="#">UdpNmPrecondAssertEnabled</a>	1..1
<a href="#">UdpNmPostcondAssertEnabled</a>	1..1
<a href="#">UdpNmStaticAssertEnabled</a>	1..1
<a href="#">UdpNmUnreachAssertEnabled</a>	1..1
<a href="#">UdpNmInvariantAssertEnabled</a>	1..1

Parameter Name	<a href="#">UdpNmDefProgEnabled</a>
----------------	-------------------------------------



<b>Label</b>	Enable Defensive Programming
<b>Description</b>	<p>Enables or disables the defensive programming feature for the module UdpNm.</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>	<b>VariantPostBuild:</b> VariantPostBuild
<b>Origin</b>	Elektrobit Automotive GmbH

<b>Parameter Name</b>	<b>UdpNmPrecondAssertEnabled</b>
<b>Label</b>	Enable Precondition Assertions
<b>Description</b>	<p>Enables handling of precondition assertion checks reported from the module UdpNm.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>UdpNmDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>UdpNmDefProgEnabled</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>UdpNmPostcondAssertEnabled</b>
<b>Label</b>	Enable Postcondition Assertions
<b>Description</b>	<p>Enables handling of postcondition assertion checks reported from the module UdpNm.</p> <p>Dependency on parameter(s):</p>



	<ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>UdpNmDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>UdpNmDefProgEnabled</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>UdpNmStaticAssertEnabled</b>
<b>Label</b>	Enable Static Assertions
<b>Description</b>	<p>Enables handling of static assertion checks reported from the module UdpNm.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>UdpNmDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>UdpNmDefProgEnabled</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>UdpNmUnreachAssertEnabled</b>
<b>Label</b>	Enable Unreachable Code Assertions
<b>Description</b>	<p>Enables handling of unreachable code assertion checks reported from the module UdpNm.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>UdpNmDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>UdpNmDefProgEnabled</code>): must be enabled</li> </ul>
<b>Multiplicity</b>	1..1



Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	
Parameter Name	<b>UdpNmInvariantAssertEnabled</b>	
Label	Enable Invariant Assertions	
Description	<p>Enables handling of invariant assertion checks reported from functions of the module UdpNm.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> <li>▶ Enable Development Error Detection (<code>UdpNmDevErrorDetect</code>): must be enabled</li> <li>▶ Enable Defensive Programming (<code>UdpNmDefProgEnabled</code>): must be enabled</li> </ul>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

## 5.9.2. Application programming interface (API)

### 5.9.2.1. Macro constants

#### 5.9.2.1.1. UDPNM\_AR\_RELEASE\_MAJOR\_VERSION

Purpose	AUTOSAR release major version.
Value	4U

#### 5.9.2.1.2. UDPNM\_AR\_RELEASE\_MINOR\_VERSION

Purpose	AUTOSAR release minor version.
---------	--------------------------------



<b>Value</b>	1U
--------------	----

#### 5.9.2.1.3. UDPNM\_AR\_RELEASE\_REVISION\_VERSION

<b>Purpose</b>	AUTOSAR release revision version.
<b>Value</b>	3U

#### 5.9.2.1.4. UDPNM\_E\_BUSSLEEPMODE

<b>Purpose</b>	Error code for case in which SchM_Call for Nm_BusSleepMode fails.
<b>Value</b>	248U

#### 5.9.2.1.5. UDPNM\_E\_CARWAKEUPINDICATION

<b>Purpose</b>	Error code for case in which SchM_Call for Nm_CarWakeUpIndication fails.
<b>Value</b>	241U

#### 5.9.2.1.6. UDPNM\_E\_INIT\_FAILED

<b>Purpose</b>	UdpNm initialization has failed, e.g. selected configuration set doesn't exist.
<b>Value</b>	0x04U

#### 5.9.2.1.7. UDPNM\_E\_INVALID\_CHANNEL

<b>Purpose</b>	Error Code for Invalid channel.
<b>Value</b>	0x02U

#### 5.9.2.1.8. UDPNM\_E\_INVALID\_PDUID

<b>Purpose</b>	API service called with wrong PDU ID.
----------------	---------------------------------------



<b>Value</b>	0x03U
--------------	-------

#### 5.9.2.1.9. UDPNM\_E\_NETWORKMODE

<b>Purpose</b>	Error code for case in which SchM_Call for Nm_NetworkMode fails.
<b>Value</b>	249U

#### 5.9.2.1.10. UDPNM\_E\_NETWORKSTARTINDICATION

<b>Purpose</b>	Error code for case in which SchM_Call for Nm_NetworkStartIndication fails.
<b>Value</b>	250U

#### 5.9.2.1.11. UDPNM\_E\_NO\_INIT

<b>Purpose</b>	Initialization status before module initilaization.
<b>Value</b>	0x01U

#### 5.9.2.1.12. UDPNM\_E\_NULL\_POINTER

<b>Purpose</b>	Error code for NULL pointers.
<b>Value</b>	0x12U

#### 5.9.2.1.13. UDPNM\_E\_PDURXINDICATION

<b>Purpose</b>	Error code for case in which SchM_Call for Nm_PduRxIndication fails.
<b>Value</b>	244U

#### 5.9.2.1.14. UDPNM\_E\_PREPAREBUSSLEEPMODE

<b>Purpose</b>	Error code for case in which SchM_Call for Nm_PreparesBusSleepMode fails.
----------------	---



<b>Value</b>	247U
--------------	------

#### **5.9.2.1.15. UDPNM\_E\_REMOTE\_SLEEP\_CANCELLATION**

<b>Purpose</b>	Error code for case in which SchM_Call for Nm_RemoteSleepCancellation fails.
<b>Value</b>	245U

#### **5.9.2.1.16. UDPNM\_E\_REMOTE\_SLEEP\_INDICATION**

<b>Purpose</b>	Error code for case in which SchM_Call for Nm_RemoteSleepIndication fails.
<b>Value</b>	246U

#### **5.9.2.1.17. UDPNM\_E\_REPEAT\_MESSAGE\_INDICATION**

<b>Purpose</b>	Error code for case in which SchM_Call for Nm_RepeatMessageIndication fails.
<b>Value</b>	242U

#### **5.9.2.1.18. UDPNM\_E\_STATE\_CHANGE\_NOTIFICATION**

<b>Purpose</b>	Error code for case in which SchM_Call for Nm_StateChangeNotification fails.
<b>Value</b>	243U

#### **5.9.2.1.19. UDPNM\_INSTANCE\_ID**

<b>Purpose</b>	Instance Id of UdpNm.
<b>Value</b>	0U

#### **5.9.2.1.20. UDPNM\_INVALID\_PDU\_INSTANCE\_ID**

<b>Purpose</b>	Instance Id of UdpNm when an invalid PDU is passed.
<b>Value</b>	255U



#### 5.9.2.1.21. UDPNM\_MODULE\_ID

<b>Purpose</b>	AUTOSAR module identification.
<b>Value</b>	33U

#### 5.9.2.1.22. UDPNM\_SERVID\_CHECKREMOTESLEEPINDICATION

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x11U
<b>Description</b>	Definition of UDPNM_SERVID_CHECKREMOTESLEEPINDICATION.

#### 5.9.2.1.23. UDPNM\_SERVID\_DISABLECOMMUNICATION

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x0CU
<b>Description</b>	Definition of UDPNM_SERVID_DISABLECOMMUNICATION.

#### 5.9.2.1.24. UDPNM\_SERVID\_ENABLECOMMUNICATION

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x0DU
<b>Description</b>	Definition of UDPNM_SERVID_ENABLECOMMUNICATION.

#### 5.9.2.1.25. UDPNM\_SERVID\_GETLOCALNODEIDENTIFIER

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x07U
<b>Description</b>	Definition of UDPNM_SERVID_GETLOCALNODEIDENTIFIER.

#### 5.9.2.1.26. UDPNM\_SERVID\_GETNODEIDENTIFIER

<b>Purpose</b>	AUTOSAR API service ID.
----------------	-------------------------



<b>Value</b>	0x06U
<b>Description</b>	Definition of UDPNM_SERVID_GETNODEIDENTIFIER.

#### 5.9.2.1.27. UDPNM\_SERVID\_GETPDUDATA

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x0AU
<b>Description</b>	Definition of UDPNM_SERVID_GETPDUDATA.

#### 5.9.2.1.28. UDPNM\_SERVID\_GETSTATE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x0BU
<b>Description</b>	Definition of UDPNM_SERVID_GETSTATE.

#### 5.9.2.1.29. UDPNM\_SERVID\_GETUSERDATA

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x05U
<b>Description</b>	Definition of UDPNM_SERVID_GETUSERDATA.

#### 5.9.2.1.30. UDPNM\_SERVID\_GETVERSIONINFO

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x09U
<b>Description</b>	Definition of UDPNM_SERVID_GETVERSIONINFO.

#### 5.9.2.1.31. UDPNM\_SERVID\_INIT

<b>Purpose</b>	AUTOSAR API service ID.
----------------	-------------------------



<b>Value</b>	0x01U
<b>Description</b>	Definition of UDPNM_SERVID_INIT.

#### 5.9.2.1.32. UDPNM\_SERVID\_MAINFUNCTION\_X

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x13U
<b>Description</b>	Definition of UDPNM_SERVID_MAINFUNCTION_X.

#### 5.9.2.1.33. UDPNM\_SERVID\_NETWORKGWERAREQUEST

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0xFEU
<b>Description</b>	Definition of UDPNM_SERVID_NETWORKGWERAREQUEST.

#### 5.9.2.1.34. UDPNM\_SERVID\_NETWORKRELEASE

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x03U
<b>Description</b>	Definition of UDPNM_SERVID_NETWORKRELEASE.

#### 5.9.2.1.35. UDPNM\_SERVID\_NETWORKREQUEST

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x02U
<b>Description</b>	Definition of UDPNM_SERVID_NETWORKREQUEST.

#### 5.9.2.1.36. UDPNM\_SERVID\_PASSIVESTARTUP

<b>Purpose</b>	AUTOSAR API service ID.
----------------	-------------------------



<b>Value</b>	0x0eU
<b>Description</b>	Definition of UDPNM_SERVID_PASSIVESTARTUP

#### 5.9.2.1.37. UDPNM\_SERVID\_REPEATMESSAGEREQUEST

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x08U
<b>Description</b>	Definition of UDPNM_SERVID_REPEATMESSAGEREQUEST.

#### 5.9.2.1.38. UDPNM\_SERVID\_REQUESTBUSSYNCHRONIZATION

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x14U
<b>Description</b>	Definition of UDPNM_SERVID_REQUESTBUSSYNCHRONIZATION.

#### 5.9.2.1.39. UDPNM\_SERVID\_RXINDICATION

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x10U
<b>Description</b>	Definition of UDPNM_SERVID_RXINDICATION.

#### 5.9.2.1.40. UDPNM\_SERVID\_SETUSERDATA

<b>Purpose</b>	AUTOSAR API service ID.
<b>Value</b>	0x04U
<b>Description</b>	Definition of UDPNM_SERVID_SETUSERDATA.

#### 5.9.2.1.41. UDPNM\_SERVID\_TXCONFIRMATION

<b>Purpose</b>	AUTOSAR API service ID.
----------------	-------------------------



<b>Value</b>	0x0FU
<b>Description</b>	Definition of UDPNM_SERVID_TXCONFIRMATION.

#### 5.9.2.1.42. UDPNM\_SW\_MAJOR\_VERSION

<b>Purpose</b>	AUTOSAR module major version.
<b>Value</b>	2U

#### 5.9.2.1.43. UDPNM\_SW\_MINOR\_VERSION

<b>Purpose</b>	AUTOSAR module minor version.
<b>Value</b>	9U

#### 5.9.2.1.44. UDPNM\_SW\_PATCH\_VERSION

<b>Purpose</b>	AUTOSAR module patch version.
<b>Value</b>	8U

#### 5.9.2.1.45. UDPNM\_VENDOR\_ID

<b>Purpose</b>	AUTOSAR vendor identification: Elektrobit Automotive GmbH.
<b>Value</b>	1U

### 5.9.2.2. Functions

#### 5.9.2.2.1. UdpNm\_CheckRemoteSleepIndication

<b>Purpose</b>	Check if sleep indication has taken place.
<b>Synopsis</b>	Std_ReturnType <b>UdpNm_CheckRemoteSleepIndication</b> ( NetworkHandleType nmChannelHandle , boolean * nmRemoteSleepIndPtr );



<b>Service ID</b>	0x011	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant (but not for the same NM-Channel)	
<b>Parameters (in)</b>	nmChannelHandle	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/not executed.	
<b>Description</b>	This function checks if remote sleep indication has taken place or not.	

### 5.9.2.2.2. UdpNm\_DisableCommunication

<b>Purpose</b>	Disable NM PDU transmission.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>UdpNm_DisableCommunication</b> ( NetworkHandleType nmChannelHandle );</pre>	
<b>Service ID</b>	0x0c	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant (But not for the same NM Channel)	
<b>Parameters (in)</b>	nmChannelHandle	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/not executed.	
<b>Description</b>	This function disables the NM PDU transmission ability due to a ISO14229 Communication Control (28hex) service.	

### 5.9.2.2.3. UdpNm\_EnableCommunication

<b>Purpose</b>	Enable NM PDU transmission.	
<b>Synopsis</b>	<pre>Std_ReturnType <b>UdpNm_EnableCommunication</b> ( NetworkHandleType nmChannelHandle );</pre>	



<b>Service ID</b>	0x0d	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant (But not for the same NM Channel)	
<b>Parameters (in)</b>	nmChannelHandle	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/not executed.
<b>Description</b>	This function enables the NM PDU transmission ability due to a ISO14229 Communication Control (28hex) service.	

#### 5.9.2.2.4. UdpNm\_GetLocalNodeIdentifier

<b>Purpose</b>	Get Local Node Identifier.	
<b>Synopsis</b>	Std_ReturnType <b>UdpNm_GetLocalNodeIdentifier</b> ( NetworkHandleType nmChannelHandle , uint8 * nmNodeIdPtr );	
<b>Service ID</b>	0x07	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	nmChannelHandle	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.
<b>Description</b>	This function gets the node identifier configured as the local node.	

#### 5.9.2.2.5. UdpNm\_GetNodeIdentifier

<b>Purpose</b>	Get Node Identifier.	
<b>Synopsis</b>	Std_ReturnType <b>UdpNm_GetNodeIdentifier</b> ( NetworkHandleType nmChannelHandle , uint8 * nmNodeIdPtr );	



<b>Service ID</b>	0x06	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	nmChannelHandle	Identification of the NM channel.
<b>Parameters (out)</b>	nmNodeIdPtr	Pointer where node identifier out of the most recently received NM PDU 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 most recently received NM PDU has failed.
<b>Description</b>	This function gets the node identifier out of the most recently received NM PDU.	

#### 5.9.2.2.6. UdpNm\_GetPduData

<b>Purpose</b>	Retrieve the data of the last received NM message.	
<b>Synopsis</b>	Std_ReturnType <b>UdpNm_GetPduData</b> ( NetworkHandleType nmChannelHandle , uint8 * nmPduDataPtr );	
<b>Service ID</b>	0x0a	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	nmChannelHandle	Identification of the NM channel.
	nmPduDataPtr	Pointer where NM PDU data 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.
<b>Description</b>	<p>This function retrieves 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>	



### 5.9.2.2.7. UdpNm\_GetState

<b>Purpose</b>	Get the State and mode of the Network Management.	
<b>Synopsis</b>	<code>Std_ReturnType UdpNm_GetState ( NetworkHandleType nmChannelHandle , Nm_StateType * nmStatePtr , Nm_ModeType * nmModePtr );</code>	
<b>Service ID</b>	0x0b	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	nmChannelHandle	Identification of the NM-channel.
<b>Parameters (out)</b>	nmStatePtr	Pointer to state of network management.
	nmModePtr	Pointer to mode of network management.
<b>Return Value</b>	Standard Return Code E_OK E_NOT_OK	
	E_OK	No Error.
	E_NOT_OK	Getting of NM state has failed.
<b>Description</b>	This function returns the state and the mode of the network management.	

### 5.9.2.2.8. UdpNm\_GetUserData

<b>Purpose</b>	Get User Data from NM messages.	
<b>Synopsis</b>	<code>Std_ReturnType UdpNm_GetUserData ( NetworkHandleType nmChannelHandle , uint8 * nmUserDataPtr );</code>	
<b>Service ID</b>	0x05	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	nmChannelHandle	Identification of the NM channel.
	nmUserDataPtr	Pointer to where user data out of the most recently received NM message shall be copied to.
<b>Return Value</b>	Standard Return Code E_OK E_NOT_OK	
	E_OK	No Error.
	E_NOT_OK	Getting of user data has failed.
<b>Description</b>	This function retrieves the user data from the last received NM message.  Preconditions:	



- The channel handle should be valid and the module should have been initialized for this channel (checked).

#### 5.9.2.2.9. `UdpNm_GetVersionInfo`

<b>Purpose</b>	This service returns the version information of this module.	
<b>Synopsis</b>	<code>void UdpNm_GetVersionInfo ( Std_VersionInfoType * versioninfo );</code>	
<b>Service ID</b>	0x09	
<b>Sync/Async</b>	synchronous	
<b>Reentrancy</b>	reentrant	
<b>Parameters (out)</b>	versioninfo	Pointer to where to store the version information of this module.
<b>Description</b>	This service returns the version information of this module.	

#### 5.9.2.2.10. `UdpNm_Init`

<b>Purpose</b>	Initialization of UdpNm module.	
<b>Synopsis</b>	<code>void UdpNm_Init ( const UdpNm_ConfigType * udpmConfigPtr );</code>	
<b>Service ID</b>	0x01	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	udpmConfigPtr	Pointer to selected configuration structure.
<b>Description</b>	Initialize the complete UdpNm module, i.e. all channels which are activated at configuration time are initialized. A UDP socket shall be set up with the TCP/IP stack.	

#### 5.9.2.2.11. `UdpNm_IsValidConfig`

<b>Purpose</b>	Validate configuration.
<b>Synopsis</b>	<code>Std_ReturnType UdpNm_IsValidConfig ( const void * voidConfigPtr );</code>
<b>Service ID</b>	0x60



<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Reentrant
<b>Return Value</b>	E_OK if the given module configurations is valid otherwise E_NOT_OK.
<b>Description</b>	Checks if the post build configuration fits to the link time configuration part.

#### 5.9.2.2.12. UdpNm\_NetworkGwEraRequest

<b>Purpose</b>	Network Gateway Era Request.	
<b>Synopsis</b>	Std_ReturnType <b>UdpNm_NetworkGwEraRequest</b> ( NetworkHandleType nmChannelHandle );	
<b>Service ID</b>	254	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant (But not for the same NM Channel)	
<b>Parameters (in)</b>	nmChannelHandle	Identification of the NM channel.
<b>Return Value</b>	Standard Return Code E_OK                          No Error. E_NOT_OK                      Requesting of network has failed.	
<b>Description</b>	This function request the network when bus communication is needed. Network state shall be changed to requested. If function is called active wakeup bit is not set	

#### 5.9.2.2.13. UdpNm\_NetworkRelease

<b>Purpose</b>	Release the Network.	
<b>Synopsis</b>	Std_ReturnType <b>UdpNm_NetworkRelease</b> ( NetworkHandleType nmChannelHandle );	
<b>Service ID</b>	0x03	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant (But not for the same NM Channel)	
<b>Parameters (in)</b>	nmChannelHandle	Identification of the NM channel.
<b>Return Value</b>	Standard Return Code E_OK                          No Error. E_NOT_OK                      Releasing of network has failed/not executed.	



<b>Description</b>	This function releases the network, when there is no need for bus communication. Network state shall be changed to released.
--------------------	---

#### 5.9.2.2.14. UdpNm\_NetworkRequest

<b>Purpose</b>	Network Request.	
<b>Synopsis</b>	Std_ReturnType <b>UdpNm_NetworkRequest</b> ( NetworkHandleType nmChannelHandle );	
<b>Service ID</b>	0x02	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant (But not for the same NM Channel)	
<b>Parameters (in)</b>	nmChannelHandle	Identification of the NM channel.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Requesting of network has failed.
<b>Description</b>	This function request the network when bus communication is needed. Network state shall be changed to requested.	

#### 5.9.2.2.15. UdpNm\_PassiveStartUp

<b>Purpose</b>	Passive startup of UdpNm module.	
<b>Synopsis</b>	Std_ReturnType <b>UdpNm_PassiveStartUp</b> ( NetworkHandleType nmChannelHandle );	
<b>Service ID</b>	0x0e	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant (But not for the same NM Channel)	
<b>Return Value</b>	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Passive startup of network management has failed/not executed.
<b>Description</b>	This function performs a passive startup of the AUTOSAR SoAd. It triggers the transition from Bus-Sleep Mode or Prepare Bus Sleep to the Network Mode in Repeat Message State.	



This service has no effect if the current state is not equal to Bus-Sleep Mode or Prepare Bus Sleep. In that case E\_NOT\_OK is returned.

#### 5.9.2.2.16. UdpNm\_RepeatMessageRequest

<b>Purpose</b>	Set the Repeat Message Request Bit.	
<b>Synopsis</b>	Std_ReturnType <b>UdpNm_RepeatMessageRequest</b> ( NetworkHandleType nmChannelHandle );	
<b>Service ID</b>	0x08	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant (but not for the same NM Channel)	
<b>Parameters (in)</b>	nmChannelHandle	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/not executed.
<b>Description</b>	This function sets the Repeat Message Request Bit for NM messages transmitted next on the bus.	

#### 5.9.2.2.17. UdpNm\_RequestBusSynchronization

<b>Purpose</b>	Request Bus Synchronization.	
<b>Synopsis</b>	Std_ReturnType <b>UdpNm_RequestBusSynchronization</b> ( NetworkHandleType nmChannelHandle );	
<b>Service ID</b>	0x014	
<b>Sync/Async</b>	synchronous	
<b>Reentrancy</b>	Reentrant (but not for the same NM Channel)	
<b>Parameters (in)</b>	nmChannelHandle	Identification of the NM-channel.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Requesting of bus synchronization has failed/not executed.
<b>Description</b>	This function requests bus synchronization.	



### 5.9.2.2.18. UdpNm\_SetUserData

<b>Purpose</b>	Set User Data for NM messages.	
<b>Synopsis</b>	<code>Std_ReturnType UdpNm_SetUserData ( NetworkHandleType nmChannelHandle , const uint8 * nmUserDataPtr );</code>	
<b>Service ID</b>	0x04	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non-Reentrant	
<b>Parameters (in)</b>	nmChannelHandle	Identification of the NM channel.
	nmUserDataPtr	Pointer where the user data for the next transmitted NM message shall be copied from.
<b>Return Value</b>	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Setting of user data has failed.
<b>Description</b>	<p>This function sets the user data for the next NM message that is transmitted on the bus.</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>	

### 5.9.2.2.19. UdpNm\_SoAdIfRxIndication

<b>Purpose</b>	Indicates a received transmission.	
<b>Synopsis</b>	<code>void UdpNm_SoAdIfRxIndication ( PduIdType RxPduId , const PduInfoType * PduInfoPtr );</code>	
<b>Parameters (in)</b>	RxPduId	Identification of the network through PDU-ID.
	PduInfoPtr	Contains the length of the received I-PDU and a pointer to a buffer containing the I-PDU.
<b>Description</b>	This function indicates the reception of an NM-message PDU.	



### 5.9.2.2.20. UdpNm\_SoAdIfTxConfirmation

<b>Purpose</b>	Confirms a transmission.	
<b>Synopsis</b>	<code>void UdpNm_SoAdIfTxConfirmation ( PduIdType TxPduId );</code>	
<b>Parameters (in)</b>	TxPduId	Identification of the network through PDU-ID.
<b>Description</b>	<p>This function confirms the transmission of a NM-package.</p> <p>Caveats: - The call context is either on interrupt level (interrupt mode) or on task level (polling mode).</p> <ul style="list-style-type: none"> <li>➤ The UdpNm module is initialized correctly.</li> </ul>	

### 5.9.2.2.21. UdpNm\_Transmit

<b>Purpose</b>	Dummy function.	
<b>Synopsis</b>	<code>Std_ReturnType UdpNm_Transmit ( PduIdType UdpNmTxPduId , const PduInfoType * PduInfoPtr );</code>	
<b>Service ID</b>	0x015	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	UdpNmTxPduId	This parameter contains a unique identifier referencing to the PDU Routing Table and thereby specifying the socket to be used for transmission of the data.
	PduInfoPtr	A pointer to a structure with socket related data: data length and pointer to a data buffer.
<b>Return Value</b>	Standard Return Code	
	E_OK:	The request has been accepted
	E_NOT_OK:	The request has not been accepted, e.g. due to a still ongoing transmission in the corresponding socket or the to be transmitted message is too long.
<b>Description</b>	UdpNm_Transmit is implemented as an empty function and shall always return E_OK. The function UdpNm_Transmit is only available if the configuration switch UdpNmCommUserDataSupport is enabled.	



Preconditions:

- ▶ None

## 5.9.3. Integration notes

### 5.9.3.1. Exclusive areas

This section describes the exclusive areas used by the UdpNm module.

#### 5.9.3.1.1. SCHM\_UDP 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 Mapping exclusive areas in the basic software modules in the Integration notes section for details.

UdpNm uses exclusive areas for protecting the global data against concurrent read/write access:

- ▶ The status of UdpNm channels - the consistency of this global variable must be assured as it can be read/written by the UdpNm state machine and/or following user interfaces:
  - ▶ UdpNm\_NetworkRequest()
  - ▶ UdpNm\_NetworkRelease()
  - ▶ UdpNm\_EnableCommunication()
  - ▶ UdpNm\_DisableCommunication()
- ▶ The partial networking bits - the consistency of this global data must be assured as it can be read/written by the UdpNm state machine, RxIndication and/or following user interface:
  - ▶ UdpNm\_GetPduUserData()
- ▶ The NM PDU data - the consistency of the PDU data must be assured as it can be read/written by the RxIndication and/or following interfaces:
  - ▶ UdpNm\_GetUserData()
  - ▶ UdpNm\_GetPduData()

### 5.9.3.2. Production errors

Production errors are not reported by the `UdpNm` module.

### 5.9.3.3. 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
CONST_8
CONST_32
VAR_INIT_8
CONST_UNSPECIFIED
CODE
VAR_INIT_UNSPECIFIED
VAR_CLEARED_8
VAR_CLEARED_UNSPECIFIED
VAR_INIT_BOOLEAN
CONFIG_DATA_UNSPECIFIED

### 5.9.3.4. 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 `UdpNm` module.



## 6. Bibliography

### Bibliography

- [1] *802.1Q-2011 - IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks,*  
3 Park Avenue  
NY 10016-5997 New York 20, USA, Publish date: August 31st, 2011, IEEE Computer Society