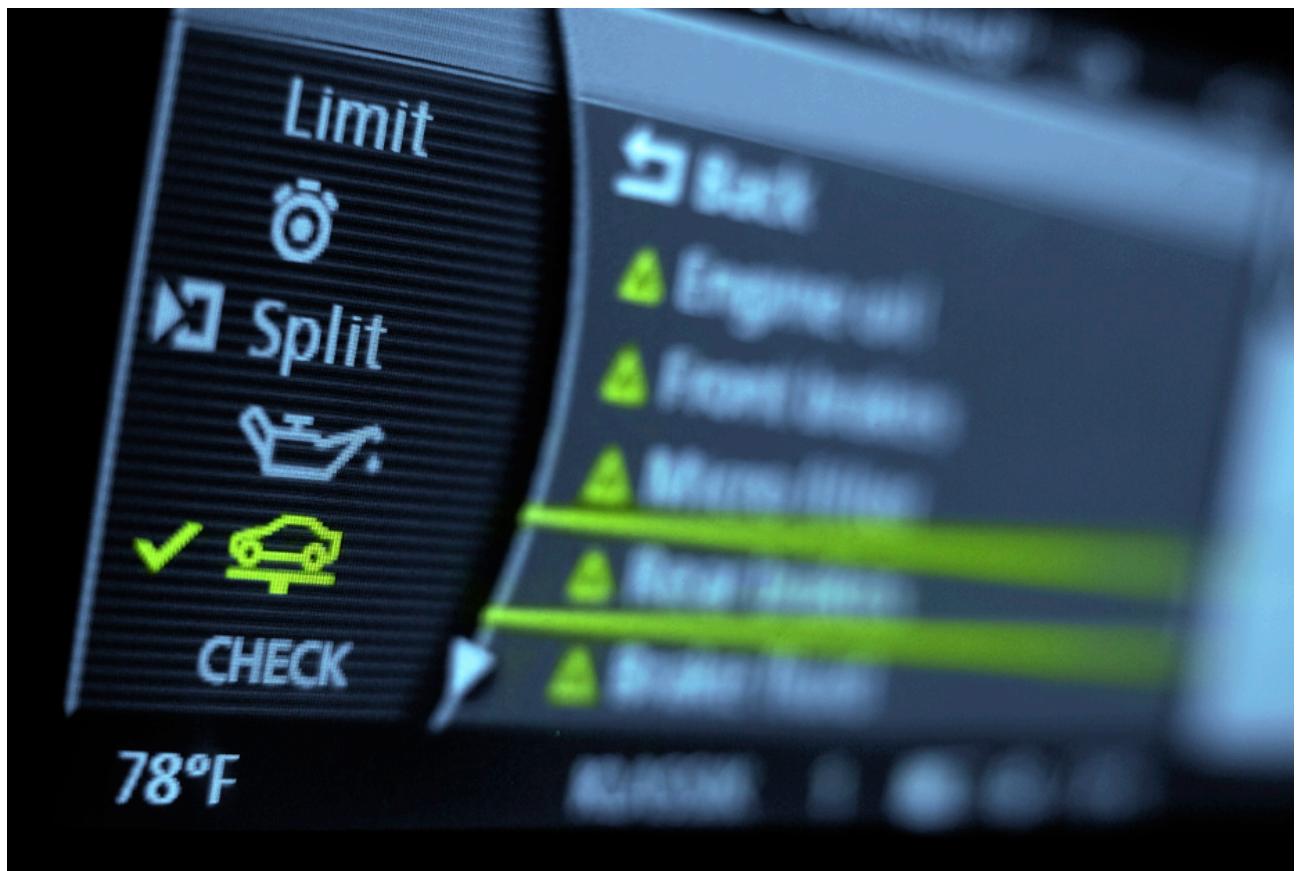




Elektrobit

EB tresos[®] AutoCore Generic 8 IP Stack documentation

product release 8.8.7





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 2022, Elektrobit Automotive GmbH.

Table of Contents

1. Overview of EB tresos AutoCore Generic 8 IP Stack documentation	33
2. Supported features	34
3. ACG8 IP Stack release notes	42
3.1. Overview	42
3.2. Scope of the release	42
3.2.1. Configuration tool	42
3.2.2. AUTOSAR modules	42
3.2.3. EB (Elektrobit) modules	43
3.2.4. MCAL modules and EB tresos AutoCore OS	43
3.3. Module release notes	43
3.3.1. DolP module release notes	43
3.3.1.1. Change log	44
3.3.1.2. New features	52
3.3.1.3. Elektrobit-specific enhancements	53
3.3.1.4. Deviations	57
3.3.1.5. Limitations	64
3.3.1.6. Open-source software	66
3.3.2. Ethlf module release notes	66
3.3.2.1. Change log	66
3.3.2.2. New features	73
3.3.2.3. Elektrobit-specific enhancements	74
3.3.2.4. Deviations	76
3.3.2.5. Limitations	82
3.3.2.6. Open-source software	84
3.3.3. EthSM module release notes	84
3.3.3.1. Change log	85
3.3.3.2. New features	90
3.3.3.3. Elektrobit-specific enhancements	90
3.3.3.4. Deviations	91
3.3.3.5. Limitations	93
3.3.3.6. Open-source software	93
3.3.4. Sd module release notes	93
3.3.4.1. Change log	93
3.3.4.2. New features	101
3.3.4.3. Elektrobit-specific enhancements	101
3.3.4.4. Deviations	102
3.3.4.5. Limitations	104
3.3.4.6. Open-source software	105
3.3.5. SoAd module release notes	106



3.3.5.1. Change log	106
3.3.5.2. New features	119
3.3.5.3. Elektrobit-specific enhancements	120
3.3.5.4. Deviations	121
3.3.5.5. Limitations	128
3.3.5.6. Open-source software	129
3.3.6. SomelpTp module release notes	129
3.3.6.1. Change log	130
3.3.6.2. New features	135
3.3.6.3. Elektrobit-specific enhancements	135
3.3.6.4. Deviations	136
3.3.6.5. Limitations	140
3.3.6.6. Open-source software	140
3.3.7. Tcplp module release notes	140
3.3.7.1. Change log	140
3.3.7.2. New features	153
3.3.7.3. Elektrobit-specific enhancements	154
3.3.7.4. Deviations	159
3.3.7.5. Limitations	203
3.3.7.6. Open-source software	208
3.3.8. UdpNm module release notes	208
3.3.8.1. Change log	209
3.3.8.2. New features	215
3.3.8.3. Elektrobit-specific enhancements	215
3.3.8.4. Deviations	216
3.3.8.5. Limitations	225
3.3.8.6. Open-source software	225
4. ACG8 IP Stack user guide	226
4.1. Overview	226
4.2. Background information	226
4.2.1. Modules of the AUTOSAR IP Stack	226
4.2.2. IP stack dependencies	228
4.2.3. Network management in AUTOSAR IP stack	229
4.3. Configuring the ACG8 IP Stack	229
4.3.1. Multiple provided service instances	229
4.3.1.1. Configuring the SoAd module	230
4.3.1.2. Configuring the Sd module	230
4.3.2. Debug support	231
4.3.2.1. GetAndResetMeasurementData service	231
4.4. DolP module user guide	231
4.4.1. Overview	231
4.4.2. Background Information	232



4.4.2.1. TCP closure APIs	233
4.4.2.2. TCP closure with FIN and RST	233
4.4.2.3. Delay of local IP address release	234
4.4.3. Configuring the DolP module	234
4.4.3.1. Configuring TCP closure APIs	234
4.4.3.2. Configuring TCP closure with FIN and RST	234
4.4.3.3. Configuring a local IP address release delay	234
4.5. EthIf module user guide	235
4.5.1. Overview	235
4.5.2. Background Information	235
4.5.2.1. Multiple device driver support	235
4.5.3. Configuring the EthIf module	236
4.5.3.1. Configuring the multiple device driver support	236
4.6. Sd module user guide	237
4.6.1. Overview	237
4.6.2. Background information	237
4.6.3. Configuring the Sd module	238
4.7. SoAd module user guide	239
4.7.1. Overview	239
4.7.2. Background information	239
4.7.2.1. N-PDU buffer pooling	239
4.7.2.1.1. Temporary unavailable buffer	239
4.7.2.1.2. Debug hooks	240
4.7.2.2. TLS and DTLS	240
4.7.3. Configuring the SoAd module	241
4.7.3.1. Configuring the N-PDU buffer pooling	241
4.7.3.2. Configuring TLS and DTLS	241
4.8. Tcplp module user guide	242
4.8.1. Overview	242
4.8.2. Background Information	242
4.8.2.1. Software hardening support	242
4.8.2.1.1. TCP SYN cookies	242
4.8.2.1.2. Unpredictable sequence numbers	243
4.8.2.2. Storage of the IP address to NVM RAM	243
4.8.2.3. IPv4 Duplicate Address Detection (ARP probing)	243
4.8.2.4. IPv6 Duplicate Address Detection	244
4.8.2.5. Debug support in Tcplp	244
4.8.3. Configuring the TcpIp module	245
4.8.3.1. Configuring the software hardening features	245
4.8.3.2. Configuring the permanent IP address storage to NVM RAM	249
4.8.3.3. Configuring the IPv4 Duplicate Address Detection (ARP probing)	251
4.8.3.4. Configuring the IPv6 Duplicate Address Detection	253



4.9. QoS Support user guide	254
4.9.1. Overview	254
4.9.2. Background information	255
4.9.3. Configuring QoS Support	257
4.9.3.1. Configuring the Ethernet Driver Module (<code>Eth</code>)	257
4.9.3.2. Configuring the Ethernet Interface Module (<code>EthIf</code>)	258
4.9.3.3. Configuring the TCP/IP Module (<code>TcpIp</code>)	258
4.9.3.4. Configuring the Socket Adaptor Module (<code>SoAd</code>)	259
4.9.3.5. Configuring the Operating System Module (<code>Os</code>)	259
4.9.3.6. Configuring the Run-Time Environment Module (<code>Rte</code>)	260
5. ACG8 IP Stack module references	261
5.1. Overview	261
5.1.1. Notation in EB module references	261
5.1.1.1. Default value of configuration parameters	261
5.1.1.2. Range information of configuration parameters	261
5.2. DolP	262
5.2.1. Configuration parameters	262
5.2.1.1. CommonPublishedInformation	263
5.2.1.2. DolPConfigSet	266
5.2.1.3. DolPChannel	267
5.2.1.4. DolPPduRRxPdu	268
5.2.1.5. DolPPduRTxPdu	269
5.2.1.6. DolPCustomChannel	270
5.2.1.7. DolPPduRRxPdu	270
5.2.1.8. DolPPduRTxPdu	271
5.2.1.9. DolPConnections	272
5.2.1.10. DolPTargetAddress	272
5.2.1.11. DolPTcpConnection	273
5.2.1.12. DolPSoAdRxPdu	274
5.2.1.13. DolPSoAdTxPdu	275
5.2.1.14. DolPUdpConnection	275
5.2.1.15. DolPSoAdRxPdu	276
5.2.1.16. DolPSoAdTxPdu	277
5.2.1.17. DolPUdpVehicleAnnouncement	278
5.2.1.18. DolPSoAdTxPdu	278
5.2.1.19. DolPRoutingActivation	279
5.2.1.20. DolPRoutingActivationAuthenticationCallback	281
5.2.1.21. DolPRoutingActivationConfirmationCallback	282
5.2.1.22. DolPTester	283
5.2.1.23. DolPGeneral	284
5.2.1.24. DolPGetGid	299
5.2.1.25. DolPPowerModeCallback	299



5.2.1.26. DoIPTriggerGidSynchronization	300
5.2.1.27. DoIPProtocolVersion	300
5.2.1.28. DoIPDynamicGIDMasterSelectionCallback	301
5.2.1.29. DoIPVIDRequestReceivedCallback	302
5.2.1.30. DoIPGetVIN	302
5.2.1.31. DoIPTcpSoConModeChgCallback	303
5.2.1.32. DoIPSecurityEventRefs	303
5.2.1.33. DoIPDefensiveProgramming	307
5.2.1.34. PublishedInformation	310
5.2.2. Application programming interface (API)	310
5.2.2.1. Macro constants	311
5.2.2.1.1. DOIP_E_INVALID_PARAMETER	311
5.2.2.1.2. DOIP_E_INVALID_PDU_SDU_ID	311
5.2.2.1.3. DOIP_E_PARAM_POINTER	311
5.2.2.1.4. DOIP_E_UNINIT	311
5.2.2.1.5. DOIP_INSTANCE_ID	311
5.2.2.1.6. DOIP_SID_ACTIVATIONLINESWITCH	312
5.2.2.1.7. DOIP_SID_CANCELRECEIVE	312
5.2.2.1.8. DOIP_SID_CANCELTRANSMIT	312
5.2.2.1.9. DOIP_SID_DISCONNECTTESTER	312
5.2.2.1.10. DOIP_SID_GETANDRESETMEASUREMENTDATA	312
5.2.2.1.11. DOIP_SID_GETSOCONIDFROMRXPDUID	312
5.2.2.1.12. DOIP_SID_GETSOCONIDFROMTXPDUID	313
5.2.2.1.13. DOIP_SID_GETVERSIONINFO	313
5.2.2.1.14. DOIP_SID_IFTRANSMIT	313
5.2.2.1.15. DOIP_SID_INIT	313
5.2.2.1.16. DOIP_SID_LOCALIPADDRASSIGNMENTCHG	313
5.2.2.1.17. DOIP_SID_MAINFUNCTION	313
5.2.2.1.18. DOIP_SID_MAINFUNCTION_TX	314
5.2.2.1.19. DOIP_SID_SOADIFRXINDICATION	314
5.2.2.1.20. DOIP_SID_SOADIFTXCONFIRMATION	314
5.2.2.1.21. DOIP_SID_SOADTPCOPYRXDATA	314
5.2.2.1.22. DOIP_SID_SOADTPCOPYTXDATA	314
5.2.2.1.23. DOIP_SID_SOADTPRXINDICATION	314
5.2.2.1.24. DOIP_SID_SOADTPSTARTOFRECEPTION	315
5.2.2.1.25. DOIP_SID_SOADTPTXCONFIRMATION	315
5.2.2.1.26. DOIP_SID_SOCONMODECHG	315
5.2.2.1.27. DOIP_SID_TPTRANSMIT	315
5.2.2.2. Functions	315
5.2.2.2.1. DoIP_ActivationLineSwitchActive	315
5.2.2.2.2. DoIP_ActivationLineSwitchInactive	316
5.2.2.2.3. DoIP_CancelReceive	316



5.2.2.2.4. DolP_CancelTransmit	316
5.2.2.2.5. DolP_DisconnectTester	317
5.2.2.2.6. DolP_GetAndResetMeasurementData	317
5.2.2.2.7. DolP_GetSoConIdFromRxPduld	318
5.2.2.2.8. DolP_GetSoConIdFromTxPduld	319
5.2.2.2.9. DolP_GetVersionInfo	319
5.2.2.2.10. DolP_Init	319
5.2.2.2.11. DolP_IsValidConfig	320
5.2.2.2.12. DolP_LocallpAddrAssignmentChg	320
5.2.2.2.13. DolP_MainFunction	321
5.2.2.2.14. DolP_MainFunctionTx	321
5.2.2.2.15. DolP_SoAdlfRxIndication	321
5.2.2.2.16. DolP_SoAdlfTxConfirmation	321
5.2.2.2.17. DolP_SoAdTpCopyRxData	322
5.2.2.2.18. DolP_SoAdTpCopyTxData	322
5.2.2.2.19. DolP_SoAdTpRxIndication	323
5.2.2.2.20. DolP_SoAdTpStartOfReception	323
5.2.2.2.21. DolP_SoAdTpTxConfirmation	324
5.2.2.2.22. DolP_SoConModeChg	324
5.2.2.2.23. DolP_TpTransmit	325
5.2.3. Integration notes	325
5.2.3.1. Exclusive areas	325
5.2.3.1.1. SCHM_DOIP_EXCLUSIVE_AREA_0	325
5.2.3.2. Production errors	326
5.2.3.3. Memory mapping	326
5.2.3.4. Integration requirements	326
5.2.3.4.1. DolP.EB_INTREQ_DolP_0001	326
5.2.3.4.2. DolP.EB_INTREQ_DolP_0002	327
5.2.3.4.3. DolP.EB_INTREQ_DolP_0003	327
5.2.3.4.4. DolP.EB_INTREQ_DolP_0004	327
5.2.3.4.5. DolP.EB_INTREQ_DolP_0005	327
5.2.3.4.6. DolP.EB_INTREQ_DolP_0006	328
5.2.3.4.7. DolP.EB_INTREQ_DolP_0007	328
5.3. Ethlf	328
5.3.1. Configuration parameters	328
5.3.1.1. CommonPublishedInformation	329
5.3.1.2. EthlfConfigSet	332
5.3.1.3. EthlfController	333
5.3.1.4. EthlfFrameOwnerConfig	336
5.3.1.5. EthlfPhysController	337
5.3.1.6. EthlfPhysCtrlRxMainFunctionPriorityProcessing	338
5.3.1.7. EthlfRxIndicationConfig	339



5.3.1.8. EthIfSwitch	340
5.3.1.9. EthIfSwitchMgmtInfoIndicationConfig	340
5.3.1.10. EthIfSwitchPortGroup	341
5.3.1.11. EthIfSwitchTimeStampIndicationConfig	342
5.3.1.12. EthIfTransceiver	342
5.3.1.13. EthIfTrcvLinkStateChgConfig	343
5.3.1.14. EthIfTxConfirmationConfig	344
5.3.1.15. EthIfEthControllerType	344
5.3.1.16. EthIfEthTrcvType	345
5.3.1.17. EthIfEthSwtType	346
5.3.1.18. EthIfGeneral	346
5.3.1.19. EthIfSecurityEventRefs	366
5.3.1.20. ReportToDem	367
5.3.1.21. EthIfDefensiveProgramming	368
5.3.1.22. PublishedInformation	371
5.3.2. Recommended configurations	371
5.3.2.1. EthIfRecConfiguration	372
5.3.2.1.1. EthIfConfigSet	372
5.3.2.1.2. EthIfController_0	372
5.3.2.1.3. EthIfFrameOwnerConfig_IPv4	373
5.3.2.1.4. EthIfFrameOwnerConfig_IPv6	373
5.3.2.1.5. EthIfFrameOwnerConfig_ARP	373
5.3.2.1.6. EthIfPhysController_0	373
5.3.2.1.7. EthIfRxIndicationConfig_IPv4	373
5.3.2.1.8. EthIfRxIndicationConfig_IPv6	374
5.3.2.1.9. EthIfRxIndicationConfig_ARP	374
5.3.2.1.10. EthIfTrcvLinkStateChgConfig_EthSM	374
5.3.2.1.11. EthIfTxConfirmationConfig_0	374
5.3.2.2. EthIfRecConfigurationEthTSyn	374
5.3.2.2.1. EthIfConfigSet	375
5.3.2.2.2. EthIfController_0	375
5.3.2.2.3. EthIfFrameOwnerConfig_PTP	375
5.3.2.2.4. EthIfFrameOwnerConfig_IPv4	376
5.3.2.2.5. EthIfFrameOwnerConfig_IPv6	376
5.3.2.2.6. EthIfFrameOwnerConfig_ARP	376
5.3.2.2.7. EthIfPhysController_0	376
5.3.2.2.8. EthIfRxIndicationConfig_PTP	376
5.3.2.2.9. EthIfRxIndicationConfig_IPv4	377
5.3.2.2.10. EthIfRxIndicationConfig_IPv6	377
5.3.2.2.11. EthIfRxIndicationConfig_ARP	377
5.3.2.2.12. EthIfTrcvLinkStateChgConfig_EthSM	377
5.3.2.2.13. EthIfTrcvLinkStateChgConfig_EthTSyn	377



5.3.2.2.14. EthIfTxConfirmationConfig_PTP	377
5.3.2.2.15. EthIfGeneral	378
5.3.3. Application programming interface (API)	378
5.3.3.1. Macro constants	378
5.3.3.1.1. ETHIF_CBK_RXINDICATION_SVCID	378
5.3.3.1.2. ETHIF_CBK_TXCONFIRMATION_SVCID	378
5.3.3.1.3. ETHIF_CHECKWAKEUP_SVCID	378
5.3.3.1.4. ETHIF_CLEARSWITCHPORTSIGNALQUALITY_SVCID	378
5.3.3.1.5. ETHIF_CLEARTRCVSIGNALQUALITY_SVCID	379
5.3.3.1.6. ETHIF_CTRLMODEINDICATION_SVCID	379
5.3.3.1.7. ETHIF_DISABLERELATEDETHIFCTRLS_SVCID	379
5.3.3.1.8. ETHIF_ENABLEEGRESSTIMESTAMP_SVCID	379
5.3.3.1.9. ETHIF_ENABLERELATEDETHIFCTRLS_SVCID	379
5.3.3.1.10. ETHIF_E_HW_NOT_INITIALIZED	379
5.3.3.1.11. ETHIF_E_INIT_FAILED	379
5.3.3.1.12. ETHIF_E_INV_CTRL_IDX	380
5.3.3.1.13. ETHIF_E_INV_PARAM	380
5.3.3.1.14. ETHIF_E_INV_PORT_GROUP_IDX	380
5.3.3.1.15. ETHIF_E_INV_SWT_IDX	380
5.3.3.1.16. ETHIF_E_INV_TRCV_IDX	380
5.3.3.1.17. ETHIF_E_NOT_INITIALIZED	380
5.3.3.1.18. ETHIF_E_PARAM_POINTER	380
5.3.3.1.19. ETHIF_GETANDRESETMEASUREMENTDATA_SVCID	381
5.3.3.1.20. ETHIF_GETARLTABLE_SVCID	381
5.3.3.1.21. ETHIF_GETBUFFERLEVEL_SVCID	381
5.3.3.1.22. ETHIF_GETCABLEDIAGNOSTICSRESULT_SVCID	381
5.3.3.1.23. ETHIF_GETCONTROLLERMODE_SVCID	381
5.3.3.1.24. ETHIF_GETCTRLIDXLIST_SVCID	381
5.3.3.1.25. ETHIF_GETCURRENTTIME_SVCID	381
5.3.3.1.26. ETHIF_GETEGRESSTIMESTAMP_SVCID	382
5.3.3.1.27. ETHIF_GETINGRESSTIMESTAMP_SVCID	382
5.3.3.1.28. ETHIF_GETPHYIDENTIFIER_SVCID	382
5.3.3.1.29. ETHIF_GETPHYSADDR_SVCID	382
5.3.3.1.30. ETHIF_GETPORTMACADDR_SVCID	382
5.3.3.1.31. ETHIF_GETSWITCHPORTSIGNALQUALITY_SVCID	382
5.3.3.1.32. ETHIF_GETTRANSCEIVERWAKEUPMODE_SVCID	382
5.3.3.1.33. ETHIF_GETTRCVSIGNALQUALITY_SVCID	383
5.3.3.1.34. ETHIF_GETVERSIONINFO_SVCID	383
5.3.3.1.35. ETHIF_GETVLANID_SVCID	383
5.3.3.1.36. ETHIF_INIT_SVCID	383
5.3.3.1.37. ETHIF_INSTANCE_ID	383
5.3.3.1.38. ETHIF_MAINFUNCTIONRX_SVCID	383



5.3.3.1.39. ETHIF_MAINFUNCTIONSTATE_SVCID	384
5.3.3.1.40. ETHIF_MAINFUNCTIONTX_SVCID	384
5.3.3.1.41. ETHIF_PROVIDETXBUFFER_SVCID	384
5.3.3.1.42. ETHIF_READMII_SVCID	384
5.3.3.1.43. ETHIF_RESETCONFIGURATION_SVCID	384
5.3.3.1.44. ETHIF_RETRANSMIT_SVCID	384
5.3.3.1.45. ETHIF_SETCONTROLLERMODE_SVCID	384
5.3.3.1.46. ETHIF_SETCORRECTIONTIME_SVCID	385
5.3.3.1.47. ETHIF_SETFORWARDINGMODE_SVCID	385
5.3.3.1.48. ETHIF_SETGLOBALTIME_SVCID	385
5.3.3.1.49. ETHIF_SETPHYLOOPBACKMODE_SVCID	385
5.3.3.1.50. ETHIF_SETPHYSADDR_SVCID	385
5.3.3.1.51. ETHIF_SETPHYTESTMODE_SVCID	385
5.3.3.1.52. ETHIF_SETPHYTXMODE_SVCID	385
5.3.3.1.53. ETHIF_SETTRANSCEIVERWAKEUPMODE_SVCID	386
5.3.3.1.54. ETHIF_STARTALLPORTS_SVCID	386
5.3.3.1.55. ETHIF_STORECONFIGURATION_SVCID	386
5.3.3.1.56. ETHIF_SWITCHMGMTINFOINDICATION_SVCID	386
5.3.3.1.57. ETHIF_SWITCHPORTGROUPREQUESTMODE_SVCID	386
5.3.3.1.58. ETHIF_SWTGETCOUNTERVALUES_SVCID	386
5.3.3.1.59. ETHIF_TRANSMIT_SVCID	386
5.3.3.1.60. ETHIF_TRCVMODEINDICATION_SVCID	387
5.3.3.1.61. ETHIF_UPDATEPHYSADDRFILTER_SVCID	387
5.3.3.1.62. ETHIF_VERIFYCONFIG_SVCID	387
5.3.3.1.63. ETHIF_WRITEMII_SVCID	387
5.3.3.1.64. TS_RELOCATABLE_CFG_ENABLE	387
5.3.3.2. Functions	387
5.3.3.2.1. EthIf_Cbk_RxIndication	387
5.3.3.2.2. EthIf_Cbk_TxConfirmation	388
5.3.3.2.3. EthIf_CheckWakeups	388
5.3.3.2.4. EthIf_ClearSwitchPortSignalQuality	389
5.3.3.2.5. EthIf_CtrlModeIndication	389
5.3.3.2.6. EthIf_DisableRelatedEthIfCtrls	390
5.3.3.2.7. EthIf_EnableEgressTimeStamp	390
5.3.3.2.8. EthIf_EnableRelatedEthIfCtrls	390
5.3.3.2.9. EthIf_GetAndResetMeasurementData	391
5.3.3.2.10. EthIf_GetArlTable	392
5.3.3.2.11. EthIf_GetBufferLevel	392
5.3.3.2.12. EthIf_GetControllerMode	393
5.3.3.2.13. EthIf_GetCtrlIdxList	393
5.3.3.2.14. EthIf_GetCurrentTime	394
5.3.3.2.15. EthIf_GetEgressTimeStamp	394



5.3.3.2.16. EthIf_GetIngressTimeStamp	394
5.3.3.2.17. EthIf_GetPhysAddr	395
5.3.3.2.18. EthIf_GetPortMacAddr	395
5.3.3.2.19. EthIf_GetSwitchPortSignalQuality	396
5.3.3.2.20. EthIf_GetTransceiverWakeUpMode	396
5.3.3.2.21. EthIf_GetVersionInfo	397
5.3.3.2.22. EthIf_GetVlanId	397
5.3.3.2.23. EthIf_Init	398
5.3.3.2.24. EthIf_IsValidConfig	398
5.3.3.2.25. EthIf_MainFunctionRx	399
5.3.3.2.26. EthIf_MainFunctionRxPrio	399
5.3.3.2.27. EthIf_MainFunctionState	399
5.3.3.2.28. EthIf_MainFunctionTx	400
5.3.3.2.29. EthIf_ProvideTxBuffer	400
5.3.3.2.30. EthIf_ReadMii	401
5.3.3.2.31. EthIf_ResetConfiguration	401
5.3.3.2.32. EthIf_Retransmit	401
5.3.3.2.33. EthIf_RxIndication	402
5.3.3.2.34. EthIf_SetControllerMode	403
5.3.3.2.35. EthIf_SetCorrectionTime	403
5.3.3.2.36. EthIf_SetForwardingMode	404
5.3.3.2.37. EthIf_SetGlobalTime	404
5.3.3.2.38. EthIf_SetPhysAddr	405
5.3.3.2.39. EthIf_SetTransceiverWakeUpMode	405
5.3.3.2.40. EthIf_StartAllPorts	406
5.3.3.2.41. EthIf_StoreConfiguration	406
5.3.3.2.42. EthIf_SwitchMgmtInfoIndication	406
5.3.3.2.43. EthIf_SwitchPortGroupRequestMode	407
5.3.3.2.44. EthIf_SwtGetCounterValues	407
5.3.3.2.45. EthIf_Transmit	408
5.3.3.2.46. EthIf_TrcvModeIndication	408
5.3.3.2.47. EthIf_TxConfirmation	409
5.3.3.2.48. EthIf_UpdatePhysAddrFilter	409
5.3.3.2.49. EthIf_VerifyConfig	410
5.3.3.2.50. EthIf_WriteMii	410
5.3.4. Integration notes	411
5.3.4.1. Exclusive areas	411
5.3.4.1.1. SCHM_ETHIF_EXCLUSIVE_AREA	411
5.3.4.2. Production errors	411
5.3.4.3. Memory mapping	411
5.3.4.4. Integration requirements	412
5.3.4.4.1. lim.EthIf.EB_INTREQ_EthIf_0001	412



5.3.4.4.2. lim.EthIf.EB_INTREQ_EthIf_0002	412
5.3.4.4.3. lim.EthIf.EB_INTREQ_EthIf_0003	412
5.3.4.4.4. lim.EthIf.EB_INTREQ_EthIf_0004	413
5.3.4.4.5. lim.EthIf.EB_INTREQ_EthIf_0005	413
5.3.4.4.6. lim.EthIf.EB_INTREQ_EthIf_0006	413
5.3.4.4.7. lim.EthIf.EB_INTREQ_EthIf_0007	413
5.3.4.4.8. lim.EthIf.EB_INTREQ_EthIf_0008	413
5.3.4.4.9. lim.EthIf.EB_INTREQ_EthIf_0009	413
5.3.4.4.10. lim.EthIf.EB_INTREQ_EthIf_0010	414
5.3.4.4.11. lim.EthIf.EB_INTREQ_EthIf_0011	414
5.3.4.4.12. lim.EthIf.EB_INTREQ_EthIf_0012	414
5.3.4.4.13. lim.EthIf.EB_INTREQ_EthIf_0013	414
5.3.4.4.14. lim.EthIf.EB_INTREQ_EthIf_0014	414
5.3.4.4.15. lim.EthIf.EB_INTREQ_EthIf_0015	414
5.4. EthSM	415
5.4.1. Configuration parameters	415
5.4.1.1. CommonPublishedInformation	416
5.4.1.2. EthSMConfig	419
5.4.1.3. EthSMGeneral	419
5.4.1.4. ReportToDem	422
5.4.1.5. EthSMDefensiveProgramming	423
5.4.1.6. EthSMNetwork	426
5.4.1.7. EthSMDemEventParameterRefs	427
5.4.1.8. PublishedInformation	428
5.4.2. Application programming interface (API)	429
5.4.2.1. Macro constants	429
5.4.2.1.1. ETHSM_CTRLMODEINDICATION_SVCID	429
5.4.2.1.2. ETHSM_DEVAUTHMODEINDICATION_SVCID	429
5.4.2.1.3. ETHSM_E_BUSSMMODEIND	429
5.4.2.1.4. ETHSM_E_INVALID_NETWORK_HANDLE	429
5.4.2.1.5. ETHSM_E_INVALID_NETWORK_MODE	430
5.4.2.1.6. ETHSM_E_INVALID_TRCV_LINK_STATE	430
5.4.2.1.7. ETHSM_E_INVALID_TcpIpMode	430
5.4.2.1.8. ETHSM_E_PARAM_CONTROLLER	430
5.4.2.1.9. ETHSM_E_PARAM_POINTER	430
5.4.2.1.10. ETHSM_E_PARAM_TRANSCEIVER	430
5.4.2.1.11. ETHSM_E_UNINIT	431
5.4.2.1.12. ETHSM_GETCURRENTCOMMODE_SVCID	431
5.4.2.1.13. ETHSM_GETCURRENTINTERNALMODE_SVCID	431
5.4.2.1.14. ETHSM_GETVERSIONINFO_SVCID	431
5.4.2.1.15. ETHSM_INIT_SVCID	431
5.4.2.1.16. ETHSM_INSTANCE_ID	431



5.4.2.1.17. ETHSM_MAINFUNCTION_SVCID	432
5.4.2.1.18. ETHSM_REQUESTCOMMODE_SVCID	432
5.4.2.1.19. ETHSM_TCPIPMODEINDICATION_SVCID	432
5.4.2.1.20. ETHSM_TRCVLINKSTATECHG_SVCID	432
5.4.2.1.21. ETHSM_TRCVMODEINDICATION_SVCID	432
5.4.2.2. Functions	432
5.4.2.2.1. EthSM_GetCurrentComMode	432
5.4.2.2.2. EthSM_GetVersionInfo	433
5.4.2.2.3. EthSM_Init	433
5.4.2.2.4. EthSM_IsValidConfig	434
5.4.2.2.5. EthSM_MainFunction	434
5.4.2.2.6. EthSM_RequestComMode	435
5.4.2.2.7. EthSM_TcpIpModeIndication	435
5.4.3. Integration notes	435
5.4.3.1. Exclusive areas	435
5.4.3.1.1. SCHM_ETHSM_EXCLUSIVE_AREA_0	436
5.4.3.2. Production errors	436
5.4.3.3. Memory mapping	436
5.4.3.4. Integration requirements	437
5.4.3.4.1. lim.EthSm.EB_INTREQ_EthSm_0001	437
5.5. Sd	437
5.5.1. Configuration parameters	437
5.5.1.1. CommonPublishedInformation	438
5.5.1.2. PublishedInformation	441
5.5.1.3. SdConfig	442
5.5.1.4. SdInstance	442
5.5.1.5. SdClientService	444
5.5.1.6. SdClientCapabilityRecord	448
5.5.1.7. SdConsumedEventGroup	448
5.5.1.8. SdClientCapabilityRecord	451
5.5.1.9. SdBlacklistedVersions	451
5.5.1.10. SdConsumedMethods	452
5.5.1.11. SdClientTimer	452
5.5.1.12. SdInstanceDemEventParameterRefs	455
5.5.1.13. SdInstanceMulticastRxPdu	456
5.5.1.14. SdInstanceTxPdu	457
5.5.1.15. SdInstanceUnicastRxPdu	457
5.5.1.16. SdServerService	458
5.5.1.17. SdEventHandler	461
5.5.1.18. SdEventHandlerMulticast	463
5.5.1.19. SdEventHandlerTcp	463
5.5.1.20. SdEventHandlerUdp	464



5.5.1.21. SdServerCapabilityRecord	465
5.5.1.22. SdProvidedMethods	465
5.5.1.23. SdServerCapabilityRecord	466
5.5.1.24. SdServerTimer	466
5.5.1.25. SdGeneral	469
5.5.1.26. VendorSpecific	473
5.5.1.27. ProductionErrorHandling	475
5.5.1.28. Sd_SendDataBackServerNotAvailable	479
5.5.1.29. Sd_SendDataBackMalformedMsg	480
5.5.1.30. Sd_SendDataBackSubscribeNackReceived	480
5.5.1.31. SdDefensiveProgramming	481
5.5.2. Application programming interface (API)	484
5.5.2.1. Macro constants	484
5.5.2.1.1. SD_CHECKBUFFERFILLLEVEL_SVCID	484
5.5.2.1.2. SD_CLIENTSERVICECEVGEXITAVAILABLE_SVCID	484
5.5.2.1.3. SD_CLIENTSERVICEGOTOAVAILABLE_SVCID	484
5.5.2.1.4. SD_CLIENTSERVICEMAINFUNCTION_SVCID	484
5.5.2.1.5. SD_CLIENTSERVICEMAITIMERS_SVCID	485
5.5.2.1.6. SD_CLIENTSERVICEMAITTL_SVCID	485
5.5.2.1.7. SD_CLIENTSERVICESETSTATE_SVCID	485
5.5.2.1.8. SD_CLIENTSERVICESINITAFTERTHALT_SVCID	485
5.5.2.1.9. SD_CLIENTSERVICE_COMSMEVENT_OFFER_SVCID	485
5.5.2.1.10. SD_CLIENTSERVICE_COMSMEVENT_START_SVCID	485
5.5.2.1.11. SD_CLIENTSERVICE_COMSMEVENT_STOPOFFER_SVCID	485
5.5.2.1.12. SD_CLIENTSERVICE_COMSMEVENT_STOP_SVCID	486
5.5.2.1.13. SD_CLIENTSERVICE_COMSMEVENT_TIMEOUT_SVCID	486
5.5.2.1.14. SD_CLIENTSERVICE_COMSM_EVENT_STOP_SVCID	486
5.5.2.1.15. SD_CLIENTSERVICE_COMSM_EVENT_TIMEOUT	486
5.5.2.1.16. SD_CLIENTSERVICE_COMSM_SOCONMODECHECK_SVCID	486
5.5.2.1.17. SD_CLIENTSERVICE_CONSUMEDEVENTGROUPSETSTATE_SVCID	486
5.5.2.1.18. SD_CLIENTSERVICE_EVENT_OFFER_SERVICE_SVCID	487
5.5.2.1.19. SD_CLIENTSERVICE_PROCESSOFFERS_SVCID	487
5.5.2.1.20. SD_CLIENTSERVICE_TTL_RUN_OUT	487
5.5.2.1.21. SD_COMPAREADDRESS_SVCID	487
5.5.2.1.22. SD_CONNECTION_SETTING_FAILED	487
5.5.2.1.23. SD_CONSUMEDEVENTGROUPSETSTATE_SVCID	487
5.5.2.1.24. SD_CONTROL_SOCKET_IMPOSSIBLE	488
5.5.2.1.25. SD_CONTROL_SOCKET_NOT_RELEASED	488
5.5.2.1.26. SD_E_CLIENT_IMPOSSIBLE_STATE	488
5.5.2.1.27. SD_E_COUNT_OF_RETRY_SUBSCRIPTION_EXCEEDED	488
5.5.2.1.28. SD_E_DIVISOR_IS_NEGATIVE	488



5.5.2.1.29. SD_E_INVALID_ARG	488
5.5.2.1.30. SD_E_INV_ID	489
5.5.2.1.31. SD_E_INV_MODE	489
5.5.2.1.32. SD_E_IPADDR_ASSIGNMENT_NOT_FINISHED	489
5.5.2.1.33. SD_E_NOT_INITIALIZED	489
5.5.2.1.34. SD_E_OUT_OF_RES	489
5.5.2.1.35. SD_E_PARAM_POINTER	489
5.5.2.1.36. SD_E_RETRY_INFINITE_TTL_NOT_INFINITE	490
5.5.2.1.37. SD_E_RETRY_PROCESS_BIGGER_THAN_TTL	490
5.5.2.1.38. SD_E_SHOULD_NOT_HAPPEN	490
5.5.2.1.39. SD_GETRANDOM_SVCID	490
5.5.2.1.40. SD_GETVERSIONINFO_SVCID	490
5.5.2.1.41. SD_HANDLERESPONSETIMERS_SVCID	490
5.5.2.1.42. SD_INIT_SVCID	490
5.5.2.1.43. SD_INSTANCEMAINFUNCTION_SVCID	491
5.5.2.1.44. SD_INSTANCE_ID	491
5.5.2.1.45. SD_INTERNAL_API_ID	491
5.5.2.1.46. SD_INVARIANT_ASSERT	491
5.5.2.1.47. SD_INVARIANT_ASSERT_NO_EVAL	491
5.5.2.1.48. SD_LOCALIPADDRASSIGNMENTCHG_SVCID	491
5.5.2.1.49. SD_MAINFUNCTION_SVCID	492
5.5.2.1.50. SD_POSTCONDITION_ASSERT	492
5.5.2.1.51. SD_POSTCONDITION_ASSERT_NO_EVAL	492
5.5.2.1.52. SD_PRECONDITION_ASSERT	492
5.5.2.1.53. SD_PRECONDITION_ASSERT_NO_EVAL	492
5.5.2.1.54. SD_RECEIVEGETENTRYOPTIONS_SVCID	492
5.5.2.1.55. SD_RESETREMOTENODERXSTATE_SVCID	493
5.5.2.1.56. SD_RXINDICATION_SVCID	493
5.5.2.1.57. SD_SENDQUEUEOPTIONS_SVCID	493
5.5.2.1.58. SD_SEND_FLUSHBUFFER_SVCID	493
5.5.2.1.59. SD_SERVERSERVICEADDEVENTGROUPSUBS_SVCID	493
5.5.2.1.60. SD_SERVERSERVICEADDMULTEVENTGROUPSUBS_SVCID	493
5.5.2.1.61. SD_SERVERSERVICEEVENTSTOPSUBSCRIBE_SVCID	494
5.5.2.1.62. SD_SERVERSERVICEHANDLETTL_SVCID	494
5.5.2.1.63. SD_SERVERSERVICEMAINFUNCTION_SVCID	494
5.5.2.1.64. SD_SERVERSERVICEMAITIMERS_SVCID	494
5.5.2.1.65. SD_SERVERSERVICEREMOVEEVENTGROUPSUBS_SVCID	494
5.5.2.1.66. SD_SERVERSERVICERESTREMOTECOMMNECTION_SVCID	494
5.5.2.1.67. SD_SERVERSERVICESETSTATE_SVCID	495
5.5.2.1.68. SD_SERVERSERVICE_COMSMEVENT_START_SVCID	495
5.5.2.1.69. SD_SERVERSERVICE_COMSMEVENT_STOPHALT_SVCID	495
5.5.2.1.70. SD_SERVERSERVICE_COMSMEVENT_TIMEOUT_SVCID	495



5.5.2.1.71. SD_STATIC_ASSERT	495
5.5.2.1.72. SD_UNREACHABLE_CODE_ASSERT	495
5.5.2.2. Functions	496
5.5.2.2.1. Sd_ClientServiceSetState	496
5.5.2.2.2. Sd_ConsumedEventGroupSetState	496
5.5.2.2.3. Sd_GetVersionInfo	497
5.5.2.2.4. Sd_Init	497
5.5.2.2.5. Sd_IsValidConfig	497
5.5.2.2.6. Sd_LocallpAddrAssignmentChg	498
5.5.2.2.7. Sd_MainFunction	498
5.5.2.2.8. Sd_RxIndication	499
5.5.2.2.9. Sd_ServerServiceSetState	499
5.5.2.2.10. Sd_StartRandom	500
5.5.3. Integration notes	500
5.5.3.1. Exclusive areas	500
5.5.3.1.1. SCHM_SD_EXCLUSIVE_AREA_0	500
5.5.3.2. Production errors	500
5.5.3.3. Memory mapping	501
5.5.3.4. Integration requirements	501
5.5.3.4.1. Sd.EB.IntReq.EB_INTREQ_Sd_0001	501
5.5.3.4.2. Sd.EB.IntReq.EB_INTREQ_Sd_0002	502
5.5.3.4.3. Sd.EB.IntReq.EB_INTREQ_Sd_0003	502
5.6. SoAd	502
5.6.1. Configuration parameters	502
5.6.1.1. CommonPublishedInformation	503
5.6.1.2. PublishedInformation	506
5.6.1.3. SoAdDefensiveProgramming	507
5.6.1.4. SoAdGeneral	510
5.6.1.5. SoAdSecurityEventRefs	513
5.6.1.6. VendorSpecific	515
5.6.1.7. SoAdBswModules	520
5.6.1.8. SoAdConfig	523
5.6.1.9. SoAdPduRoute	524
5.6.1.10. SoAdPduRouteDest	526
5.6.1.11. SoAdSocketConnectionGroup	528
5.6.1.12. SoAdSocketConnection	532
5.6.1.13. SoAdSocketRemoteAddress	533
5.6.1.14. SoAdSocketProtocol	533
5.6.1.15. SoAdSocketUdp	534
5.6.1.16. SoAdSocketTcp	537
5.6.1.17. SoAdSocketRoute	540
5.6.1.18. SoAdSocketRouteDest	541



5.6.1.19. SoAdRoutingGroup	542
5.6.2. Recommended configurations	543
5.6.2.1. SoAdRecConfigurationDflt	543
5.6.2.1.1. PduR	544
5.6.2.1.2. Sd	544
5.6.2.1.3. DolP	544
5.6.3. Application programming interface (API)	545
5.6.3.1. Type definitions	545
5.6.3.1.1. SoAd_DtlsConnectionStateType	545
5.6.3.2. Macro constants	545
5.6.3.2.1. SOAD_CHANGEPARAMETER_SVCID	545
5.6.3.2.2. SOAD_CLOSESOCON_SVCID	545
5.6.3.2.3. SOAD_COPYTXDATA_SVCID	546
5.6.3.2.4. SOAD_DISABLEROUTING_SVCID	546
5.6.3.2.5. SOAD_DISABLESPECIFICROUTING_SVCID	546
5.6.3.2.6. SOAD_DTLSSTATECHG_SVCID	546
5.6.3.2.7. SOAD_DTLS_STATE_CLOSED	546
5.6.3.2.8. SOAD_DTLS_STATE_CONNECTED	546
5.6.3.2.9. SOAD_DTLS_STATE_CONNECTING	546
5.6.3.2.10. SOAD_ENABLEROUTING_SVCID	547
5.6.3.2.11. SOAD_ENABLESPECIFICROUTING_SVCID	547
5.6.3.2.12. SOAD_E_INV_ARG	547
5.6.3.2.13. SOAD_E_INV_LENGTH_IN_PDUHEADER	547
5.6.3.2.14. SOAD_E_INV_METADATA	547
5.6.3.2.15. SOAD_E_INV_PDUHEADER_ID	547
5.6.3.2.16. SOAD_E_INV_PDUUID	547
5.6.3.2.17. SOAD_E_INV_SOCKETID	548
5.6.3.2.18. SOAD_E_NOBUFS	548
5.6.3.2.19. SOAD_E_NOTINIT	548
5.6.3.2.20. SOAD_E_PARAM_POINTER	548
5.6.3.2.21. SOAD_E_TRIGGERTXBUF	548
5.6.3.2.22. SOAD_E_TWO_PDUS SAME_SOCON	548
5.6.3.2.23. SOAD_E_TWO_SOCON SAME_PDU	549
5.6.3.2.24. SOAD_GETANDRESETMEASUREMENTDATA_SVCID	549
5.6.3.2.25. SOAD_GETLOCALADDR_SVCID	549
5.6.3.2.26. SOAD_GETPHYSADDR_SVCID	549
5.6.3.2.27. SOAD_GETREMOTEADDR_SVCID	549
5.6.3.2.28. SOAD_GETSOCONID_SVCID	549
5.6.3.2.29. SOAD_GETSOCONMODE_SVCID	549
5.6.3.2.30. SOAD_GETVERSIONINFO_SVCID	550
5.6.3.2.31. SOAD_IFROUTINGGROUPTRANSMIT_SVCID	550
5.6.3.2.32. SOAD_IFSPECIFICROUTINGGROUPTRANSMIT_SVCID	550



5.6.3.2.33. SOAD_IFTRANSMIT_SVCID	550
5.6.3.2.34. SOAD_INIT_SVCID	550
5.6.3.2.35. SOAD_INSTANCE_ID	550
5.6.3.2.36. SOAD_INTERNAL_SVCID	551
5.6.3.2.37. SOAD_INVALID_ULFUNCID	551
5.6.3.2.38. SOAD_ISCONNECTIONREADY_SVCID	551
5.6.3.2.39. SOAD_LOCALIPADDRASSIGNMENTCHG_SVCID	551
5.6.3.2.40. SOAD_MAINFUNCTIONTX_SVCID	551
5.6.3.2.41. SOAD_MAINFUNCTION_SVCID	551
5.6.3.2.42. SOAD_OPENSOCN_SVCID	551
5.6.3.2.43. SOAD_READDHCPHOSTNAMEOPTION_SVCID	552
5.6.3.2.44. SOAD_RELEASEIPADDRASSIGNMENT_SVCID	552
5.6.3.2.45. SOAD_RELEASEREMOTEADDR_SVCID	552
5.6.3.2.46. SOAD_REQUESTIPADDRASSIGNMENT_SVCID	552
5.6.3.2.47. SOAD_RXINDICATION_SVCID	552
5.6.3.2.48. SOAD_SETREMOTEADDR_SVCID	552
5.6.3.2.49. SOAD_SETUNIQUEREMOTEADDR_SVCID	553
5.6.3.2.50. SOAD_TCpaccepted_SVCID	553
5.6.3.2.51. SOAD_TCPCONNECTED_SVCID	553
5.6.3.2.52. SOAD_TCPIPEVENT_SVCID	553
5.6.3.2.53. SOAD_TPCANCELRECEIVE_SVCID	553
5.6.3.2.54. SOAD_TPCANCELTRANSMIT_SVCID	553
5.6.3.2.55. SOAD_TPCHANGEPARAMETER_SVCID	553
5.6.3.2.56. SOAD_TPTRANSMIT_SVCID	554
5.6.3.2.57. SOAD_TXCONFIRMATION_SVCID	554
5.6.3.2.58. SOAD_Writedhcphostnameoption_SVCID	554
5.6.3.2.59. SoAdIf_Transmit	554
5.6.3.2.60. SoAdTp_CancelReceive	554
5.6.3.2.61. SoAdTp_CancelTransmit	554
5.6.3.2.62. SoAdTp_Transmit	554
5.6.3.3. Functions	555
5.6.3.3.1. SoAd_ChangeParameter	555
5.6.3.3.2. SoAd_CloseSoCon	555
5.6.3.3.3. SoAd_CopyTxData	556
5.6.3.3.4. SoAd_DisableRouting	556
5.6.3.3.5. SoAd_DisableSpecificRouting	557
5.6.3.3.6. SoAd_DtlsConnectionStateChg	557
5.6.3.3.7. SoAd_EnableRouting	558
5.6.3.3.8. SoAd_EnableSpecificRouting	558
5.6.3.3.9. SoAd_GetAndResetMeasurementData	559
5.6.3.3.10. SoAd_GetLocalAddr	560
5.6.3.3.11. SoAd_GetPhysAddr	561



5.6.3.3.12. SoAd_GetRemoteAddr	561
5.6.3.3.13. SoAd_GetSoConId	562
5.6.3.3.14. SoAd_GetSoConMode	562
5.6.3.3.15. SoAd_GetVersionInfo	563
5.6.3.3.16. SoAd_IfRoutingGroupTransmit	563
5.6.3.3.17. SoAd_IfSpecificRoutingGroupTransmit	564
5.6.3.3.18. SoAd_IfTransmit	564
5.6.3.3.19. SoAd_Init	565
5.6.3.3.20. SoAd_IsConnectionReady	565
5.6.3.3.21. SoAd_IsValidConfig	566
5.6.3.3.22. SoAd_LocallpAddrAssignmentChg	566
5.6.3.3.23. SoAd_MainFunction	567
5.6.3.3.24. SoAd_MainFunctionTx	567
5.6.3.3.25. SoAd_OpenSoCon	567
5.6.3.3.26. SoAd_ReadDhcpHostNameOption	568
5.6.3.3.27. SoAd_ReleaselpAddrAssignment	568
5.6.3.3.28. SoAd_ReleaseRemoteAddr	569
5.6.3.3.29. SoAd_RequestlpAddrAssignment	569
5.6.3.3.30. SoAd_RxIndication	570
5.6.3.3.31. SoAd_SetRemoteAddr	571
5.6.3.3.32. SoAd_SetUniqueRemoteAddr	571
5.6.3.3.33. SoAd_TcpAccepted	572
5.6.3.3.34. SoAd_TcpConnected	573
5.6.3.3.35. SoAd_TcplpEvent	573
5.6.3.3.36. SoAd_TpCancelReceive	573
5.6.3.3.37. SoAd_TpCancelTransmit	574
5.6.3.3.38. SoAd_TpChangeParameter	574
5.6.3.3.39. SoAd_TpTransmit	575
5.6.3.3.40. SoAd_TxConfirmation	576
5.6.3.3.41. SoAd_WriteDhcpHostNameOption	576
5.6.4. Integration notes	577
5.6.4.1. Exclusive areas	577
5.6.4.1.1. SOAD_ENTER_CRITSEC()	577
5.6.4.2. Production errors	577
5.6.4.3. Memory mapping	577
5.6.4.4. Integration requirements	578
5.6.4.4.1. lim.SoAd.EB_INTREQ_SoAd_0001	578
5.6.4.4.2. lim.SoAd.EB_INTREQ_SoAd_0003	579
5.6.4.4.3. lim.SoAd.EB_INTREQ_SoAd_0004	580
5.6.4.4.4. lim.SoAd.EB_INTREQ_SoAd_0006	581
5.6.4.4.5. lim.SoAd.EB_INTREQ_SoAd_0008	581
5.6.4.4.6. lim.SoAd.EB_INTREQ_SoAd_0009	581



5.6.4.4.7. lim.SoAd.EB_INTREQ_SoAd_0010	581
5.6.4.4.8. lim.SoAd.EB_INTREQ_SoAd_0011	581
5.6.4.4.9. lim.SoAd.EB_INTREQ_SoAd_0012	582
5.6.4.4.10. lim.SoAd.EB_INTREQ_SoAd_0013	582
5.7. SomelpTp	582
5.7.1. Configuration parameters	582
5.7.1.1. CommonPublishedInformation	583
5.7.1.2. PublishedInformation	586
5.7.1.3. SomelpTpChannel	587
5.7.1.4. SomelpTpRxNSdu	588
5.7.1.5. SomelpTpRxNPdu	589
5.7.1.6. SomelpTpTxNSdu	590
5.7.1.7. SomelpTpTxNPdu	591
5.7.1.8. SomelpTpUnSupportedChannel	591
5.7.1.9. SomelpTpUnSupportedRxNSdu	592
5.7.1.10. SomelpTpUnSupportedRxNPdu	592
5.7.1.11. SomelpTpUnSupportedTxNSdu	592
5.7.1.12. SomelpTpUnSupportedTxNPdu	592
5.7.1.13. SomelpTpGeneral	592
5.7.2. Application programming interface (API)	594
5.7.2.1. Macro constants	594
5.7.2.1.1. SOMEIPTP_DET_REPORT_ERROR	594
5.7.2.1.2. SOMEIPTP_DET_REPORT_RUNTIME_ERROR	594
5.7.2.1.3. SOMEIPTP_E_ASSEMBLY_INTERRUPT	594
5.7.2.1.4. SOMEIPTP_E_ASSEMBLY_INTERRUPT_SOR	595
5.7.2.1.5. SOMEIPTP_E_DISASSEMBLY_INTERRUPT	595
5.7.2.1.6. SOMEIPTP_E_INCONSISTENT_HEADER	595
5.7.2.1.7. SOMEIPTP_E_INCONSISTENT_SEQUENCE	595
5.7.2.1.8. SOMEIPTP_E_INVALID_CALL	595
5.7.2.1.9. SOMEIPTP_E_MESSAGE_TYPE	595
5.7.2.1.10. SOMEIPTP_E_MISSING_SOMEIPTP_HEADER	596
5.7.2.1.11. SOMEIPTP_E_NOTINIT	596
5.7.2.1.12. SOMEIPTP_E_PARAM	596
5.7.2.1.13. SOMEIPTP_E_PARAM_POINTER	596
5.7.2.1.14. SOMEIPTP_INSTANCE_ID	596
5.7.2.1.15. SOMEIPTP_SID_GETVERSIONINFO	596
5.7.2.1.16. SOMEIPTP_SID_INIT	597
5.7.2.1.17. SOMEIPTP_SID_MAINFUNCTIONRX	597
5.7.2.1.18. SOMEIPTP_SID_MAINFUNCTIONTX	597
5.7.2.1.19. SOMEIPTP_SID_PDURIFTRANSMIT	597
5.7.2.1.20. SOMEIPTP_SID_PDURTPCOPYRXDATA	597
5.7.2.1.21. SOMEIPTP_SID_PDURTPCOPYTXDATA	597



5.7.2.1.22. SOMEIPTP_SID_PDURTPRXINDICATION	598
5.7.2.1.23. SOMEIPTP_SID_PDURTPSTARTOFRECEPTION	598
5.7.2.1.24. SOMEIPTP_SID_PDURPTXCONFIRMATION	598
5.7.2.1.25. SOMEIPTP_SID_RXINDICATION	598
5.7.2.1.26. SOMEIPTP_SID_TRANSMIT	598
5.7.2.1.27. SOMEIPTP_SID_TRIGGERTRANSMIT	598
5.7.2.1.28. SOMEIPTP_SID_TXCONFIRMATION	598
5.7.2.2. Functions	599
5.7.2.2.1. SomelpTp_GetVersionInfo	599
5.7.2.2.2. SomelpTp_Init	599
5.7.2.2.3. SomelpTp_MainFunctionRx	599
5.7.2.2.4. SomelpTp_MainFunctionTx	599
5.7.2.2.5. SomelpTp_PduRIfTransmit	600
5.7.2.2.6. SomelpTp_PduRTpCopyRxData	600
5.7.2.2.7. SomelpTp_PduRTpCopyTxData	600
5.7.2.2.8. SomelpTp_PduRTpRxIndication	601
5.7.2.2.9. SomelpTp_PduRTpStartOfReception	601
5.7.2.2.10. SomelpTp_PduRTpTxConfirmation	601
5.7.2.2.11. SomelpTp_RxIndication	601
5.7.2.2.12. SomelpTp_Transmit	602
5.7.2.2.13. SomelpTp_TriggerTransmit	602
5.7.2.2.14. SomelpTp_TxConfirmation	603
5.7.3. Integration notes	603
5.7.3.1. Exclusive areas	603
5.7.3.1.1. SCHM_SOMEIPTP_EXCLUSIVE_AREA_0	603
5.7.3.1.2. SCHM_SOMEIPTP_EXCLUSIVE_AREA_1	604
5.7.3.2. Production errors	604
5.7.3.3. Memory mapping	604
5.7.3.4. Integration requirements	605
5.8. Tcplp	605
5.8.1. Configuration parameters	605
5.8.1.1. CommonPublishedInformation	606
5.8.1.2. PublishedInformation	609
5.8.1.3. TcplpConfig	609
5.8.1.4. TcplpCtrl	610
5.8.1.5. TcplpOffloadChecksum	612
5.8.1.6. TcplpIpVXCtrl	613
5.8.1.7. TcplpIpV4Ctrl	613
5.8.1.8. TcplpIpV4MtuConfig	615
5.8.1.9. TcplpIpV6Ctrl	616
5.8.1.10. TcplpIpV6MtuConfig	617
5.8.1.11. TcplpDhcpServerConfig	618



5.8.1.12. TcplpDhcpAddressAssignment	619
5.8.1.13. TcplpIpConfig	620
5.8.1.14. TcplpIpFragmentationConfig	620
5.8.1.15. TcplpIpV4Config	624
5.8.1.16. TcplpArpConfig	625
5.8.1.17. TcplpAutolpConfig	627
5.8.1.18. TcplpDhcpConfig	629
5.8.1.19. TcplpDhcpConfigurableOptions	634
5.8.1.20. TcplpIcmpConfig	635
5.8.1.21. TcplpIcmpMsgHandler	636
5.8.1.22. TcplpIpV4ArpPacketFilter	636
5.8.1.23. TcplpIpFragmentationConfig	637
5.8.1.24. TcplpIpV6Config	639
5.8.1.25. TcplpDhcpV6Config	639
5.8.1.26. TcplpDhcpV6ConfigurableOptions	646
5.8.1.27. TcplpIcmpV6Config	647
5.8.1.28. TcplpIcmpV6MsgHandler	649
5.8.1.29. TcplpIpV6FragmentationConfig	650
5.8.1.30. TcplpNdpConfig	652
5.8.1.31. TcplpNdpArNudConfig	652
5.8.1.32. TcplpNdpPrefixRouterDiscoveryConfig	657
5.8.1.33. TcplpNdpPrefixList	661
5.8.1.34. TcplpNdpPrefixListEntry	661
5.8.1.35. TcplpNdpSlaacConfig	662
5.8.1.36. TcplpIpV6ConfigExtHeaderFilter	664
5.8.1.37. TcplpLocalAddr	664
5.8.1.38. TcplpAddrAssignment	666
5.8.1.39. TcplpStaticIpAddressConfig	669
5.8.1.40. TcplpNvmBlock	670
5.8.1.41. TcplpPhysAddrConfig	671
5.8.1.42. TcplpPhysAddrChgHandler	671
5.8.1.43. TcplpSocketOwnerConfig	672
5.8.1.44. TcplpSocketOwner	672
5.8.1.45. TcplpTcpConfig	675
5.8.1.46. TcplpTcpConfigOptionFilter	685
5.8.1.47. TcplpTcpUnpredictableSeqNumbers	686
5.8.1.48. TcplpTcpSynCookies	687
5.8.1.49. TcplpUdpConfig	689
5.8.1.50. TcplpDuplicateAddressDetectionConfig	690
5.8.1.51. TcplpRxPolicyCheckIngressHandlerConfig	690
5.8.1.52. TcplpRxPolicyCheckRoutingHandlerConfig	691
5.8.1.53. TcplpRxPacketPostProcessHandlerConfig	692



5.8.1.54. TcplpRxPacketDropHandlerConfig	692
5.8.1.55. TcplpMemoryConfig	693
5.8.1.56. TcplpMemoryPool	693
5.8.1.57. TcplpIpSecConfig	694
5.8.1.58. TcplpIpSecGeneral	695
5.8.1.59. TcplpIkeGeneral	697
5.8.1.60. TcplpIkeSignatureCfg	702
5.8.1.61. TcplpIkeSignatureVerify	705
5.8.1.62. TcplpIkeFragmentationCfg	706
5.8.1.63. TcplpIpSecConnections	708
5.8.1.64. TcplpIpSecRemoteAddrConfig	710
5.8.1.65. TcplpIpSecRemoteSingleAddress	710
5.8.1.66. TcplpIpSecRemoteAddressRange	711
5.8.1.67. TcplpIpSecRemotePhysAddr	712
5.8.1.68. TcplpIpSecSecurityPolicies	712
5.8.1.69. TcplpIpSecSecurityRule	714
5.8.1.70. TcplpIkeConnectionCfg	715
5.8.1.71. TcplpIkeLocalAuthenticationMethod	716
5.8.1.72. TcplpIkeLocalPresharedKeyAuthenticationMethod	716
5.8.1.73. TcplpIkeLocalCertificateAuthenticationMethod	717
5.8.1.74. TcplpIkeRemoteAuthenticationMethod	718
5.8.1.75. TcplpIkeRemotePresharedKeyAuthenticationMethod	719
5.8.1.76. TcplpIkeRemoteCertificateAuthenticationMethod	719
5.8.1.77. TcplpIpSecSecurityAssociationCommonCfg	720
5.8.1.78. TcplpIpSecSecurityProposal	722
5.8.1.79. TcplpIpSecSecurityAssociationCfg	723
5.8.1.80. TcplpIkeSecurityAssociationCommonCfg	724
5.8.1.81. TcplpIkeSecurityProposal	726
5.8.1.82. TcplpIkeSecurityAssociationCfg	729
5.8.1.83. TcplpIkeCertificate	732
5.8.1.84. TcplpIkeIdentifiers	733
5.8.1.85. TcplpIkeIdentifierItems	734
5.8.1.86. TcplpIkeIdentifierItemId	734
5.8.1.87. TcplpIkePreconfiguredIdentifierItemId	734
5.8.1.88. TcplpIkeCustomIdentifierItemId	735
5.8.1.89. TcplpIkeIdentifierSrc	735
5.8.1.90. TcplpIkeConfiguredIdentifier	736
5.8.1.91. TcplpIkeExtractedIdentifier	736
5.8.1.92. TcplpIpSecReportErrorHandler	737
5.8.1.93. TcplpDefensiveProgramming	738
5.8.1.94. TcplpGeneral	741
5.8.1.95. TcplpSecurityEventRefs	752



5.8.1.96. TcpllpV4General	753
5.8.1.97. TcpllpV6General	757
5.8.2. Recommended configurations	761
5.8.2.1. TcplpRecConfiguration	761
5.8.2.1.1. TcplpConfig	761
5.8.2.1.2. TcplpSocketOwnerConfig	762
5.8.2.1.3. TcplpSocketOwner	762
5.8.2.1.4. TcpllpConfig	762
5.8.2.1.5. TcpllpV4Config	763
5.8.2.1.6. TcplpArpConfig	763
5.8.3. Application programming interface (API)	763
5.8.3.1. Type definitions	763
5.8.3.1.1. Tcplp_ArpCacheEntryType	763
5.8.3.1.2. Tcplp_EventType	764
5.8.3.1.3. Tcplp_IPsecStateType	764
5.8.3.1.4. Tcplp_IpAddrAssignmentType	764
5.8.3.1.5. Tcplp_IpAddrStateType	764
5.8.3.1.6. Tcplp_IpSecChildSaStatusType	764
5.8.3.1.7. Tcplp_IpSeclkeSaStatusType	765
5.8.3.1.8. Tcplp_Ip_HandleIdType	765
5.8.3.1.9. Tcplp_Ip_RxReturnType	766
5.8.3.1.10. Tcplp_NdpCacheEntryType	766
5.8.3.1.11. Tcplp_ParamIdType	766
5.8.3.1.12. Tcplp_ProtocolType	766
5.8.3.1.13. Tcplp_ReturnType	766
5.8.3.1.14. Tcplp_SockAddrPtrType	766
5.8.3.1.15. Tcplp_SockAddrType	767
5.8.3.1.16. Tcplp_SocketIdType	767
5.8.3.1.17. Tcplp_StateType	767
5.8.3.2. Macro constants	767
5.8.3.2.1. TCPIP_AF_INET	767
5.8.3.2.2. TCPIP_AF_INET6	768
5.8.3.2.3. TCPIP_AF_UNSPEC	768
5.8.3.2.4. TCPIP_DHCPV6_OPTION_FQDN	768
5.8.3.2.5. TCPIP_DHCP_OPTION_FQDN	768
5.8.3.2.6. TCPIP_DTLS_STATE_CLOSED	768
5.8.3.2.7. TCPIP_DTLS_STATE_CONNECTED	768
5.8.3.2.8. TCPIP_DTLS_STATE_CONNECTING	769
5.8.3.2.9. TCPIP_E_ADDRINUSE	769
5.8.3.2.10. TCPIP_E_ADDRNOTASSIGNED	769
5.8.3.2.11. TCPIP_E_ADDRNOTAVAIL	769
5.8.3.2.12. TCPIP_E_AFNOSUPPORT	769



5.8.3.2.13. TCPIP_E_DESTADDRREQ	769
5.8.3.2.14. TCPIP_E_DROP	770
5.8.3.2.15. TCPIP_E_INV_ARG	770
5.8.3.2.16. TCPIP_E_INV_SOCKADDR	770
5.8.3.2.17. TCPIP_E_ISCONN	770
5.8.3.2.18. TCPIP_E_MEMORY	770
5.8.3.2.19. TCPIP_E_MSGSIZE	770
5.8.3.2.20. TCPIP_E_NOBUFS	771
5.8.3.2.21. TCPIP_E_NOPROTOOPT	771
5.8.3.2.22. TCPIP_E_NOTCONN	771
5.8.3.2.23. TCPIP_E_NOTINIT	771
5.8.3.2.24. TCPIP_E_NOT_OK	771
5.8.3.2.25. TCPIP_E_NOT_PERMITTED	771
5.8.3.2.26. TCPIP_E_PARAM_POINTER	772
5.8.3.2.27. TCPIP_E_PENDING	772
5.8.3.2.28. TCPIP_E_PHYS_ADDR_MISS	772
5.8.3.2.29. TCPIP_E_PROTOTYPE	772
5.8.3.2.30. TCPIP_IPADDR_ASSIGNMENT_ALL	772
5.8.3.2.31. TCPIP_IPADDR_ASSIGNMENT_DHCP	772
5.8.3.2.32. TCPIP_IPADDR_ASSIGNMENT_IPV6_ROUTER	772
5.8.3.2.33. TCPIP_IPADDR_ASSIGNMENT_LINKLOCAL	773
5.8.3.2.34. TCPIP_IPADDR_ASSIGNMENT_LINKLOCAL_DOIP	773
5.8.3.2.35. TCPIP_IPADDR_ASSIGNMENT_STATIC	773
5.8.3.2.36. TCPIP_IPADDR_STATE_ASSIGNED	773
5.8.3.2.37. TCPIP_IPADDR_STATE_ONHOLD	773
5.8.3.2.38. TCPIP_IPADDR_STATE_UNASSIGNED	773
5.8.3.2.39. TCPIP IPPROTO_TCP	774
5.8.3.2.40. TCPIP IPPROTO_UDP	774
5.8.3.2.41. TCPIP_IPSEC_ACTIVE	774
5.8.3.2.42. TCPIP_IPSEC_INACTIVE	774
5.8.3.2.43. TCPIP_IPSEC_INVALID_ICV	774
5.8.3.2.44. TCPIP_IPSEC_SECURE	774
5.8.3.2.45. TCPIP_IP_RX_DEST_UNREACHABLE	774
5.8.3.2.46. TCPIP_IP_RX_EXT_HDR_OK	775
5.8.3.2.47. TCPIP_IP_RX_FRAGMENT_HEADER_SHORT	775
5.8.3.2.48. TCPIP_IP_RX_FRAGMENT_LENGTH	775
5.8.3.2.49. TCPIP_IP_RX_FRAGMENT_OFFSET	775
5.8.3.2.50. TCPIP_IP_RX_FRAGMENT_OUT_OF_BUFFERS	775
5.8.3.2.51. TCPIP_IP_RX_FRAGMENT_OVERLAP	775
5.8.3.2.52. TCPIP_IP_RX_FRAGMENT_RESERVED	776
5.8.3.2.53. TCPIP_IP_RX_FRAGMENT_TIMEOUT	776
5.8.3.2.54. TCPIP_IP_RX_INVALID_HEADER	776



5.8.3.2.55. TCPIP_IP_RX_NOT_OK	776
5.8.3.2.56. TCPIP_IP_RX_OK	776
5.8.3.2.57. TCPIP_IP_RX_PORT_UNREACHABLE	776
5.8.3.2.58. TCPIP_IP_RX_PROTOCOL_UNREACHABLE	776
5.8.3.2.59. TCPIP_IP_RX_PROTOCOL_UNREACHABLE_FRAG	777
5.8.3.2.60. TCPIP_IP_RX_RES_POLICYCHECK	777
5.8.3.2.61. TCPIP_IP_RX_SRCADDR_INVALID	777
5.8.3.2.62. TCPIP_IP_RX_UNRECOGNIZED_OPTION	777
5.8.3.2.63. TCPIP_IP_RX_UNRECOGNIZED_ROUTING_TYPE	777
5.8.3.2.64. TCPIP_LOCALADDRID_ANY	777
5.8.3.2.65. TCPIP_OK	778
5.8.3.2.66. TCPIP_PARAMID_DSCP	778
5.8.3.2.67. TCPIP_PARAMID_FLOWLABEL	778
5.8.3.2.68. TCPIP_PARAMID_FRAMEPRIO	778
5.8.3.2.69. TCPIP_PARAMID_PATHMTU_ENABLE	778
5.8.3.2.70. TCPIP_PARAMID_TCP_KEEPALIVE	778
5.8.3.2.71. TCPIP_PARAMID_TCP_KEEPALIVE_INTERVAL	779
5.8.3.2.72. TCPIP_PARAMID_TCP_KEEPALIVE_PROBES_MAX	779
5.8.3.2.73. TCPIP_PARAMID_TCP_KEEPALIVE_TIME	779
5.8.3.2.74. TCPIP_PARAMID_TCP_NAGLE	779
5.8.3.2.75. TCPIP_PARAMID_TCP_OPTIONFILTER	779
5.8.3.2.76. TCPIP_PARAMID_TCP_RXWND_MAX	779
5.8.3.2.77. TCPIP_PARAMID_TCP_TLS_RXWND_MAX	779
5.8.3.2.78. TCPIP_PARAMID_TLS_CONNECTION_ASSIGNMENT	780
5.8.3.2.79. TCPIP_PARAMID_TTL	780
5.8.3.2.80. TCPIP_PARAMID_UDP_CHECKSUM	780
5.8.3.2.81. TCPIP_PARAMID_UDP_UNSPECIFIED_IP	780
5.8.3.2.82. TCPIP_PHYS_ADDR_ENTRY_STATE_DELAY	780
5.8.3.2.83. TCPIP_PHYS_ADDR_ENTRY_STATE_FREE	780
5.8.3.2.84. TCPIP_PHYS_ADDR_ENTRY_STATE_INCOMPLETE	781
5.8.3.2.85. TCPIP_PHYS_ADDR_ENTRY_STATE_PROBE	781
5.8.3.2.86. TCPIP_PHYS_ADDR_ENTRY_STATE_REACHABLE	781
5.8.3.2.87. TCPIP_PHYS_ADDR_ENTRY_STATE_STALE	781
5.8.3.2.88. TCPIP_PHYS_ADDR_ENTRY_STATE_STATIC	781
5.8.3.2.89. TCPIP_PORT_ANY	781
5.8.3.2.90. TCPIP_SOCKETID_INVALID	782
5.8.3.2.91. TCPIP_STATE_OFFLINE	782
5.8.3.2.92. TCPIP_STATE_ONHOLD	782
5.8.3.2.93. TCPIP_STATE_ONLINE	782
5.8.3.2.94. TCPIP_STATE_SHUTDOWN	782
5.8.3.2.95. TCPIP_STATE_STARTUP	782
5.8.3.2.96. TCPIP_TCP_CLOSED	783



5.8.3.2.97. TCPIP_TCP_FIN_RECEIVED	783
5.8.3.2.98. TCPIP_TCP_RESET	783
5.8.3.2.99. TCPIP_TLS_ALERT	783
5.8.3.2.100. TCPIP_TLS_CLOSE_NOTIFY_RECEIVED	783
5.8.3.2.101. TCPIP_TLS_HANDSHAKE_SUCCEEDED	783
5.8.3.2.102. TCPIP_TLS_WARNING	784
5.8.3.2.103. TCPIP_UDP_CLOSED	784
5.8.3.3. Functions	784
5.8.3.3.1. Tcplp_Bind	784
5.8.3.3.2. Tcplp_ChangeParameter	785
5.8.3.3.3. Tcplp_Close	786
5.8.3.3.4. Tcplp_DhcpGetStatus	786
5.8.3.3.5. Tcplp_DhcpReadOption	787
5.8.3.3.6. Tcplp_DhcpV6ReadOption	788
5.8.3.3.7. Tcplp_DhcpV6WriteOption	788
5.8.3.3.8. Tcplp_DhcpWriteOption	789
5.8.3.3.9. Tcplp_DtlsClose	790
5.8.3.3.10. Tcplp_DtlsConnectionStateChg	790
5.8.3.3.11. Tcplp_GetAndResetMeasurementData	791
5.8.3.3.12. Tcplp_GetArpCacheEntries	792
5.8.3.3.13. Tcplp_GetCtrlIdx	792
5.8.3.3.14. Tcplp_GetIpvAddr	793
5.8.3.3.15. Tcplp_GetNdpCacheEntries	794
5.8.3.3.16. Tcplp_GetPhysAddr	795
5.8.3.3.17. Tcplp_GetRemotePhysAddr	795
5.8.3.3.18. Tcplp_GetVersionInfo	796
5.8.3.3.19. Tcplp_IcmpTransmit	797
5.8.3.3.20. Tcplp_IcmpTransmitError	797
5.8.3.3.21. Tcplp_Init	798
5.8.3.3.22. Tcplp_IpSecGetStatus	799
5.8.3.3.23. Tcplp_IsConnectionReady	800
5.8.3.3.24. Tcplp_IsValidConfig	801
5.8.3.3.25. Tcplp_MainFunction	801
5.8.3.3.26. Tcplp_MainFunctionTx	801
5.8.3.3.27. Tcplp_ReleaselpAddrAssignment	802
5.8.3.3.28. Tcplp_RequestComMode	802
5.8.3.3.29. Tcplp_RequestIpAddrAssignment	803
5.8.3.3.30. Tcplp_RequestIpSecMode	803
5.8.3.3.31. Tcplp_RxIndication	804
5.8.3.3.32. Tcplp_SecurityMainFunction	805
5.8.3.3.33. Tcplp_SetRemotePhysAddr	805
5.8.3.3.34. Tcplp_Socket_getSocket	806



5.8.3.3.35. Tcplp_TcpConnect	807
5.8.3.3.36. Tcplp_TcpListen	807
5.8.3.3.37. Tcplp_TcpReceived	808
5.8.3.3.38. Tcplp_TcpTransmit	809
5.8.3.3.39. Tcplp_TlsClose	810
5.8.3.3.40. Tcplp_TlsTcpReceived	810
5.8.3.3.41. Tcplp_TlsTcpTransmit	811
5.8.3.3.42. Tcplp_TlsUdpTransmit	812
5.8.3.3.43. Tcplp_UdpTransmit	813
5.8.3.3.44. Tcplp_UICopyTxData	814
5.8.3.3.45. Tcplp_UIRxIndication	815
5.8.3.3.46. Tcplp_UITcpIpEvent	815
5.8.3.3.47. Tcplp_UITxConfirmation	816
5.8.4. Integration notes	816
5.8.4.1. Exclusive areas	816
5.8.4.1.1. SCHM_TCPIP_EXCLUSIVE_AREA_0	816
5.8.4.1.2. SCHM_TCPIP_EXCLUSIVE_AREA_1	817
5.8.4.1.3. SCHM_TCPIP_EXCLUSIVE_AREA_2	817
5.8.4.1.4. SCHM_TCPIP_EXCLUSIVE_AREA_3	817
5.8.4.2. Production errors	818
5.8.4.3. Memory mapping	818
5.8.4.4. Integration requirements	819
5.8.4.4.1. Tcplp.EB_INTREQ_Tcplp_0001	819
5.8.4.4.2. Tcplp.EB_INTREQ_Tcplp_0002	819
5.8.4.4.3. Tcplp.EB_INTREQ_Tcplp_0003	819
5.8.4.4.4. Tcplp.EB_INTREQ_Tcplp_0004	819
5.8.4.4.5. Tcplp.EB_INTREQ_Tcplp_0005	820
5.8.4.4.6. Tcplp.EB_INTREQ_Tcplp_0006	820
5.8.4.4.7. Tcplp.EB_INTREQ_Tcplp_0007	820
5.8.4.4.8. Tcplp.EB_INTREQ_Tcplp_0008	820
5.8.4.4.9. Tcplp.EB_INTREQ.uisnSecretKeyLen	820
5.8.4.4.10. Tcplp.EB_INTREQ.secretHashOutLen	821
5.8.4.4.11. Tcplp.EB_INTREQ.storingIpAddresses	821
5.8.4.4.12. Tcplp.EB_INTREQ.ASILD_EthIf_ProvideTxBuffer	821
5.8.4.4.13. Tcplp.EB_INTREQ.ASILD_EthIf_GetPhysAddr	821
5.8.4.4.14. Tcplp.EB_INTREQ.ASILD_Tcplp_GetRemotePhysAddr	821
5.8.4.4.15. Tcplp.EB_INTREQ.localCertificates	821
5.8.4.4.16. Tcplp.EB_INTREQ.SecurityMainFunction	822
5.8.4.4.17. Tcplp.EB_INTREQ.IpSecGetStatus	822
5.8.4.4.18. Tcplp.EB_INTREQAppendingEsn	822
5.9. UdpNm	822
5.9.1. Configuration parameters	822



5.9.1.1. CommonPublishedInformation	823
5.9.1.2. PublishedInformation	826
5.9.1.3. UdpNmGeneral	827
5.9.1.4. UdpNmGlobalConfig	829
5.9.1.5. UdpNmChannelConfig	838
5.9.1.6. UdpNmRxPdu	851
5.9.1.7. UdpNmTxPdu	852
5.9.1.8. UdpNmUserDataTxPdu	852
5.9.1.9. UdpNmUserDataRxPdu	853
5.9.1.10. UdpNmPnInfo	853
5.9.1.11. UdpNmPnFilterMaskByte	854
5.9.1.12. UdpNmDefensiveProgramming	855
5.9.2. Application programming interface (API)	858
5.9.2.1. Macro constants	858
5.9.2.1.1. UDPNM_AR_RELEASE_MAJOR_VERSION	858
5.9.2.1.2. UDPNM_AR_RELEASE_MINOR_VERSION	858
5.9.2.1.3. UDPNM_AR_RELEASE_REVISION_VERSION	859
5.9.2.1.4. UDPNM_E_BUSSLEEPMODE	859
5.9.2.1.5. UDPNM_E_CARWAKEUPINDICATION	859
5.9.2.1.6. UDPNM_E_INIT_FAILED	859
5.9.2.1.7. UDPNM_E_INVALID_CHANNEL	859
5.9.2.1.8. UDPNM_E_INVALID_FUNCTION_ARG	859
5.9.2.1.9. UDPNM_E_INVALID_PDUID	860
5.9.2.1.10. UDPNM_E_INVALID_PN_SYNC_SHUTDOWN_REQUEST	860
5.9.2.1.11. UDPNM_E_NETWORKMODE	860
5.9.2.1.12. UDPNM_E_NETWORKSTARTINDICATION	860
5.9.2.1.13. UDPNM_E_NO_INIT	860
5.9.2.1.14. UDPNM_E_NULL_POINTER	860
5.9.2.1.15. UDPNM_E_PDURXINDICATION	860
5.9.2.1.16. UDPNM_E_PREPAREBUSSLEEPMODE	861
5.9.2.1.17. UDPNM_E_REMOTESENTRYCANCELLATION	861
5.9.2.1.18. UDPNM_E_REMOTESENTRYINDICATION	861
5.9.2.1.19. UDPNM_E_REPEATMESSAGEINDICATION	861
5.9.2.1.20. UDPNM_E_STATECHANGENOTIFICATION	861
5.9.2.1.21. UDPNM_E_TRANSMISSION_OF_PN_SHUTDOWN_MESSAGE_FAILED	861
5.9.2.1.22. UDPNM_INSTANCE_ID	862
5.9.2.1.23. UDPNM_INVALID_PDU_INSTANCE_ID	862
5.9.2.1.24. UDPNM_MODULE_ID	862
5.9.2.1.25. UDPNM_SERVID_CHECKREMOTESLEEPINDICATION	862
5.9.2.1.26. UDPNM_SERVID_DISABLECOMMUNICATION	862
5.9.2.1.27. UDPNM_SERVID_ENABLECOMMUNICATION	862



5.9.2.1.28. UDPNM_SERVID_GETLOCALNODEIDENTIFIER	863
5.9.2.1.29. UDPNM_SERVID_GETNODEIDENTIFIER	863
5.9.2.1.30. UDPNM_SERVID_GETPDUDATA	863
5.9.2.1.31. UDPNM_SERVID_GETSTATE	863
5.9.2.1.32. UDPNM_SERVID_GETUSERDATA	863
5.9.2.1.33. UDPNM_SERVID_GETVERSIONINFO	864
5.9.2.1.34. UDPNM_SERVID_INIT	864
5.9.2.1.35. UDPNM_SERVID_MAINFUNCTION_X	864
5.9.2.1.36. UDPNM_SERVID_NETWORKGWERAREQUEST	864
5.9.2.1.37. UDPNM_SERVID_NETWORKRELEASE	864
5.9.2.1.38. UDPNM_SERVID_NETWORKREQUEST	864
5.9.2.1.39. UDPNM_SERVID_PASSIVESTARTUP	865
5.9.2.1.40. UDPNM_SERVID_REPEATMESSAGEREQUEST	865
5.9.2.1.41. UDPNM_SERVID_REQUESTBUSSYNCHRONIZATION	865
5.9.2.1.42. UDPNM_SERVID_REQUESTSYNCHRONIZEDPNCSSHUTDOWN	865
5.9.2.1.43. UDPNM_SERVID_RXINDICATION	865
5.9.2.1.44. UDPNM_SERVID_SETUSERDATA	866
5.9.2.1.45. UDPNM_SERVID_TXCONFIRMATION	866
5.9.2.1.46. UDPNM_SW_MAJOR_VERSION	866
5.9.2.1.47. UDPNM_SW_MINOR_VERSION	866
5.9.2.1.48. UDPNM_SW_PATCH_VERSION	866
5.9.2.1.49. UDPNM_VENDOR_ID	866
5.9.2.2. Functions	867
5.9.2.2.1. UdpNm_CheckRemoteSleepIndication	867
5.9.2.2.2. UdpNm_DisableCommunication	867
5.9.2.2.3. UdpNm_EnableCommunication	868
5.9.2.2.4. UdpNm_GetLocalNodeIdentifier	868
5.9.2.2.5. UdpNm_GetNodeIdentifier	869
5.9.2.2.6. UdpNm_GetPduData	869
5.9.2.2.7. UdpNm_GetState	870
5.9.2.2.8. UdpNm_GetUserData	870
5.9.2.2.9. UdpNm_GetVersionInfo	871
5.9.2.2.10. UdpNm_Init	871
5.9.2.2.11. UdpNm_IsValidConfig	872
5.9.2.2.12. UdpNm_NetworkGwEraRequest	872
5.9.2.2.13. UdpNm_NetworkRelease	872
5.9.2.2.14. UdpNm_NetworkRequest	873
5.9.2.2.15. UdpNm_PassiveStartUp	873
5.9.2.2.16. UdpNm_RepeatMessageRequest	874
5.9.2.2.17. UdpNm_RequestBusSynchronization	874
5.9.2.2.18. UdpNm_RequestSynchronizedPncShutdown	875
5.9.2.2.19. UdpNm_SetUserData	875



5.9.2.2.20. UdpNm_SoAdIfRxIndication	876
5.9.2.2.21. UdpNm_SoAdIfTxConfirmation	876
5.9.2.2.22. UdpNm_Transmit	877
5.9.3. Integration notes	878
5.9.3.1. Exclusive areas	878
5.9.3.1.1. SCHM_UDP自称_EXCLUSIVE_AREA_0	878
5.9.3.2. Production errors	878
5.9.3.3. Memory mapping	879
5.9.3.4. Integration requirements	879
6. Bibliography	880



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 synchronized PNC shutdown:** Support for synchronized PNC shutdown across multiple ECUs.
- ▶ **Support for post-build:**
- ▶ Support for handling post-build loadable and selectable configuration in Sd, DoIP, EthIf, SoAd, TcpIP, and UdpNm.



- ▶ Support for post-build loadable configuration in EthSM.
- ▶ **DolP support:** Support for Diagnostic over IP (DolP) 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 the vehicle network and vice versa.
 - ▶ **Diagnostic message relaying:** Relaying of diagnostic messages between the external testing device 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 the 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.
 - ▶ **Diagnostic power mode information:** The DolP module provides information to the external testing device as to whether diagnostics can be performed on the connected vehicle or whether the vehicle must be put into a different diagnostic power mode first.
 - ▶ **Service component interfaces:** Provision of service component interfaces in addition to callback functions.
 - ▶ **Inactivity time-out for TCP connections:** Support for inactivity time-outs to free TCP connections of an inactive tester.
 - ▶ **Parallel testers:** Support for 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 an inactive tester connection.
 - ▶ **Support OEM-specific custom payload types:** DolP routes OEM-specific custom payload types to be processed by complex device drivers.
 - ▶ **Detect a 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 the tester already before the diagnostic communication is established to keep the ECU awake. Rationale: The ECU might switch to sleep mode faster than it takes to establish the diagnostic communication.



- ▶ **Reject insecure connection requests:** DolP rejects connection requests that are received via an insecure connection. Implementation according to AUTOSAR RfC AR-3349.
- ▶ **Closing TCP connections with FIN and RST:** DolP is able to close TCP connections as a soft closure (closure with FIN), and as an abrupt TCP connection reset (closure with RST flag).
- ▶ **External APIs for closing TCP connections:** DolP provides a set of APIs to allow an upper layer to disconnect testers by closing its TCP connection.
- ▶ **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:



DEM event	Condition
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: 29.2.0 b220916-0321

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
DoIP	4.1.3 []	4.1.3 [0000]	1.1.26	Elektrobit Automotive GmbH
EthIf	4.3.0 []	4.3.0 [0000]	1.9.21	Elektrobit Automotive GmbH
EthSM	4.3.0 []	4.3.0 [0000]	1.6.16	Elektrobit Automotive GmbH
Sd	4.2.1 []	4.2.1 [0000]	1.4.16	Elektrobit Automotive GmbH
SoAd	4.2.2 []	4.2.2 [0000]	1.8.22	Elektrobit Automotive GmbH
SomelpTp	4.3.0 []	4.3.0 [0000]	1.0.36	Elektrobit Automotive GmbH

Module name	AUTOSAR version and revision	SWS version and revision	Module version	Supplier
Tcplp	4.3.0 []	4.3.0 [0000]	3.5.17	Elektrobit Automotive GmbH
UdpNm	4.1.3 []	3.3.0 [3]	2.9.11	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`¹. 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
- ▶ AUTOSAR SWS document version: 4.1.3
- ▶ Module version: 1.1.26.B567464
- ▶ Supplier: Elektrobit Automotive GmbH

¹`$TRESOS_BASE` is the location at which you installed EB tresos Studio.



3.3.1.1. Change log

This chapter lists the changes between different versions.

Module version 1.1.26

2022-09-16

- ▶ Implemented support of protocol version 3.

Module version 1.1.25

2022-06-10

- ▶ Improved handling of DoIPGetVin user callout.
- ▶ Implemented support for reporting security events to IdsM.

Module version 1.1.24

2022-02-18

- ▶ ASCDOIP-734 Fixed known issue: Vehicle announcement messages are transmitted only after first activation line switch.

Module version 1.1.23

2021-12-10

- ▶ ASCDOIP-734 Fixed known issue: DoIP does not close TCP connections with FIN.
- ▶ Implemented APIs to close a DoIP TCP connection.
- ▶ Implemented Report TxConfirmation NOT_OK if SoAd_TpTransmit returns NOT_OK.

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 DolP_MainFunction to DolP_MainFunctionTx and added an option to make the latter externally callable.
- ▶ ASCDOIP-368 Fixed known issue: DolP fails to transmit diagnostic acknowledge message when DolP-NumByteDiagAckNack > 0.
- ▶ ASCDOIP-369 Fixed known issue: Alive check request resets Rx state machine.
- ▶ Changed to provide both header and payload when DolP_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: DolP_SoConModeChg() is not called due to incorrect configuration in DolP_AdjacentLayer.properties file.
- ▶ ASCDOIP-331 Fixed known issue: Generator sets RoutingActivationIdx to 0 for all routes regardless of configured DolPRoutingActivationNumber.
- ▶ ASCDOIP-332 Fixed known issue: DolP RX side blocked forever when response message is transmitted in the context of DolP_SoAdTpTxConfirmation() and SoAd_TpTransmit() returns E_NOT_OK.
- ▶ Added limit to DolPMaxRequestBytes based on PduLengthType.
- ▶ ASCDOIP-337 Fixed known issue: DolP 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

- ▶ Implemented support of protocol version 3.



3.3.1.3. Elektrobit-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.
- ▶ `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.

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, or `DoIP_ActivationLineSwitchInactive()` is called. "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, etc. When `DoIPEnableTcpClosureWithFIN` is disabled, Tcp connection reset is performed in all cases.



- ▶ DoIP is extended with a set of external APIs to allow an upper layer to disconnect testers by closing its TCP connections. Both closure methods, with FIN and RST flags, are supported.

List of available APIs are:

```
Std_ReturnType DoIP_DisconnectTester(SoAd_SoConIdType SoConId, boolean Abort); Std_ReturnType DoIP_GetSoConIdFromRxPduId(PduIdType RxPduId, SoAd_SoConIdType* SoConIdPtr); Std_ReturnType DoIP_GetSoConIdFromTxPduId(PduIdType TxPduId, SoAd_SoConIdType* SoConIdPtr);
```

SoConId parameter in `DoIP_DisconnectTester()` is provided via routing activation callback functions when configured, or it can be retrieved with `DoIP_GetSoConIdFromRxPduId()` or `DoIP_GetSoConIdFromTxPduId()` APIs.

Actual connection closure is performed in the next main function call.

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.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(). 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 AUTSOAR bugzillas: https://www.autosar.org/bugzilla/show_bug.cgi?id=66944. https://www.autosar.org/bugzilla/show_bug.cgi?id=74847.

Requirements:

SWS_DoIP_00201, SWS_DoIP_00202, SWS_DoIP_00285, SWS_DoIP_91000, SWS_DoIP_00302, SWS_DoIP_00204, SWS_DoIP_00303, SWS_DoIP_00234, SWS_DoIP_00306, 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`

- ▶ Callback function `User_DoIPGetVin` used instead of `Dcm_GetVin()`

Description:

The function `Dcm_GetVin()` is not called. Instead `User_DoIPGetVin` 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_TpCancelTransmit()` and `DoIP_TpCancelReceive()` 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 parameters `PduInfoPtr` and `Retry`.

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 for `DoIPRoutingActivationRef` and `DoIPTargetAddressRef`

Description:

Lower multiplicity changed from 1 to 0 for `../DoIPTester/DoIPRoutingActivationRef` and `../DoIPRoutingActivation/DoIPTargetAddressRef` due to introduction of automatic routing activation.

Requirements:

ECUC_DoIP_00034, ECUC_DoIP_00062

- ▶ 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. 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

- ▶ 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_00130, SWS_DoIP_00131, SWS_DoIP_00226, SWS_DoIP_00277, SWS_DoIP_00279

- ▶ Transmitting NACK 0x06 before routing activation configurable



Description:

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 DoIPEnableTcpClosureWithFIN 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,

- ▶ SoAd_TpTransmit() moved from the context of DoIP_TpTransmit() to DoIP_MainFunction()

Description:

When the function DoIP_TpTransmit is called and the data package is allowed to be sent according to the current DoIP protocol related information, message will only be queued here and E_OK will be returned. Sending of message will be initiated (SoAd_TpTransmit()) during DoIP_MainFunction() call, and DoIP header will be assembled during DoIP_SoAdTpCopyTxData() call.

Requirements:

SWS_DoIP_00230, SWS_DoIP_00284

- ▶ Parameters DoIPRoutingActivationSecurityRequired and DoIPRoutingActivationSecurityRequired are not optional

Description:

Lower multiplicity changed from 0 to 1 for the parameters: ..\DoIPRoutingActivation\DoIPRoutingActivationSecurityRequired and ..\DoIPTcpConnection\DoIPTcpConnectionSecurityRequired, i.e. they are no longer optional.



Rationale:

Making parameter `DoIPTcpConnectionSecurityRequired` optional has no impact.

Requirements:

ECUC_DoIP_00096, ECUC_DoIP_00097

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

DolP 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: 1.9.21.B567464
- ▶ Supplier: Elektrobit Automotive GmbH

3.3.2.1. Change log

This chapter lists the changes between different versions.

Module version 1.9.21

2022-06-10

- ▶ Implemented support for introduction of new EthIfController instances at PostBuild time.

Module version 1.9.20

2022-02-18

- ▶ Implemented Measurement data support update.
- ▶ Added config parameter EthIfEnableSignalQualityApi to enable/disable read and clear signal quality APIs.
- ▶ ASCETHIF-695 Fixed known issue: Multiple and single driver support using vendor ID and vendor API infix require matching BSWMD and config schema package short name.
- ▶ Implemented support for reporting security events to IdsM.

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_TrcvLinkStateChg() 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

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



3.3.2.3. Elektrobit-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.

- ▶ `EthIf` provides additional Measurement data counters

Description:

In addition to the AUTOSAR 4.3.0 specified `EthIf_MeasurementIdxTypes` `EthIf` also provides following counters: `ETHIF_MEAS_DROP_UNKNOWN_ETHERTYPE` - An ethernet datagram was dropped due the unknown EtherType `ETHIF_MEAS_DROP_DOUBLE_VLANTAG` - An ethernet datagram was dropped due to double VLAN tag i.e. two 802.1Q headers with TPID = 0x8100.

- ▶ Configuration parameter `EthIfController/EthIfCtrlIdx` is modified to be post-build changeable.

Description:

Although according to SWS, `EthIfCtrlIdx` has pre-compile configuration class, according to TPS_ECUC_-08002, it shall be possible to add new container elements of `EthIfController`.

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.

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.

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_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`, `SWS_EthIf_91054`



- ▶ `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

- ▶ 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, ECUC_EthIf_00058

- ▶ Value configuration class changed for EthIfCtrlIdx configuration parameter.

Description:

The following configuration parameter is modified to be post-build changeable: - EthIfController/EthIfCtrlIdx

Rationale:

Rationale: Although according to SWS, EthIfCtrlIdx has pre-compile configuration class, according to TPS_ECUC_08002, it shall be possible to add new container elements of EthIfController.

Requirements:

EthIf.ECUC_EthIf_00026

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_GetArlTable - 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.

- ▶ Signal Quality functionality described in AUTOSAR Release 4.3.1 is not fully implemented.

Description:

For backward compatibility reason and required Switch functionality EthIf supports: 1) EthIf_GetTrcvSignalQuality which calls EthTrcv_GetPhySignalQuality according to ASR4.3.0 2) EthIf_GetSwitchPortSignalQuality which calls EthSwt_GetPortSignalQuality according to ASR4.3.1 3) EthIf_ClearTrcvSignalQuality according to ASR4.3.1 4) EthIf_ClearSwitchPortSignalQuality according to ASR4.3.1 EthIf does not support: 1) EthIf_GetTrcvSignalQuality does not call EthTrcv_GetTrcvSignalQuality as specified in ASR4.-3.1 2) EthIf_ClearTrcvSignalQuality does not call EthTrcv_ClearSignalQuality as specified in ASR4.3.1 3) EthIf_ClearSwitchPortSignalQuality does not call EthSwt_ClearSignalQuality as specified in ASR4.3.1 4) EthIfSignalQualityCheckPeriod as specified in ASR4.3.1 is not supported

Rationale:

Signal Quality functionality described in AUTOSAR Release 4.3.1 is contradictory.

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: 1.6.16.B567464
- ▶ Supplier: Elektrobit Automotive GmbH



3.3.3.1. Change log

This chapter lists the changes between different versions.

Module version 1.6.16

2022-06-10

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

Module version 1.6.15

2022-02-18

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

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. Elektrobit-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. With the change of EthIf_SetControllerMode() from synchronous to asynchronous EthSM has to wait for EthSM_CtrlModeIndication() which is not considered in the specified state machine. Therefore, additional sub states and transitions are required.

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

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: 1.4.16.B567464
- ▶ Supplier: Elektrobit Automotive GmbH

3.3.4.1. Change log

This chapter lists the changes between different versions.

Module version 1.4.16

2022-09-16



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

Module version 1.4.15

2022-09-13

- ▶ ASCSD-862 Fixed known issue: Sd module might get stuck in infinite loop when receiving a particularly crafted message.
- ▶ Improved optimization of RAM consumption of Sd for large values of SdMaximumRemoteNodes.

Module version 1.4.14

2022-06-10

- ▶ ASCSD-860 Fixed known issue: Wrong error messages are shown if a PDU/socket route references a connection bundle.

Module version 1.4.13

2022-04-01

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

Module version 1.4.12

2022-02-18

- ▶ ASCSD-819 Fixed known issue: If functionality Server Service Not Available Call-back is enabled, TCP endpoint IP Address is not sent back correctly.
- ▶ Added optimization of RAM consumption of Sd for large values of SdMaximumRemoteNodes.

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. Elektrobit-specific enhancements

This chapter lists the enhancements provided by the module.

- ▶ This module provides no Elektrobit-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: 1.8.22.B567464
- ▶ Supplier: Elektrobit Automotive GmbH

3.3.5.1. Change log

This chapter lists the changes between different versions.

Module version 1.8.22

2022-09-16

- ▶ Improved TCP-TLS client connection establishment for SD SomeIP communication.

Module version 1.8.21

2022-06-10

- ▶ Implemented support for reporting security events to IdsM.
- ▶ Implemented handling FIN received with TIs.

Module version 1.8.20

2022-05-06

- ▶ Improved TLS API handling.

Module version 1.8.19

2022-02-18

- ▶ Added support for DTLS.

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 be 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 `TcpIp-Bind()`, `TcpIp_TcpListen()`, and `TcpIp_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 `TcpIp_DhcpRead/WriteOption()` pass wrong parameter formats and wrong option values.
- ▶ ASCSOAD-334 Fixed known issue: `SoAd_ChangeParameter()` calls `TcpIp_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 `TcpIpAddrId` 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

- ▶ Handling FIN received with TIs.



Implemented support for reporting security events to IdsM.

3.3.5.3. Elektrobit-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 `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 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.

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 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 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`

- ▶ `SoAd_TcplpEvent()` does not report a Det for event unknown to `SoAd`.



Description:

If SoAd_TcpIpEvent() is called with an event that is currently not known to SoAd, the module no longer reports a Det and ignores the unknown event.

Rationale:

This allows lower layers to report more events than currently required by SoAd level and ensures easy future usage.

Requirements:

SWS_SoAd_00278

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: 1.0.36.B567464
- ▶ Supplier: Elektrobit Automotive GmbH



3.3.6.1. Change log

This chapter lists the changes between different versions.

Module version 1.0.36

2022-09-16

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

Module version 1.0.35

2022-07-08

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

Module version 1.0.34

2022-06-10

- ▶ ASCSOMEIPTP-126 Fixed known issue: SomelpTp calls the PduR upper TP functions with a wrong PDU ID

Module version 1.0.33

2022-05-06

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

Module version 1.0.32

2022-03-18

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

Module version 1.0.31

2022-02-18

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

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: SomeTp 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_SomeTpCopyTxData().
- ▶ Support for TxConfirmation Timeout support.
- ▶ SomeTp shall provide PduR Tp and IF APIs compliant to ASR 403.
- ▶ ASCSOMEIPTP-47 Fixed known issue: Invalid check for the SduLength passed through SomeTp_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. Elektrobit-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

SomeIpTp 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: 3.5.17.B567464
- ▶ Supplier: Elektrobit Automotive GmbH

3.3.7.1. Change log

This chapter lists the changes between different versions.

Module version 3.5.17

2022-09-16

- ▶ ASCTCPIP-3050 Fixed known issue: Possible reuse of GCM initialization vectors when 2^32 INIT requests are forced by an adversary within IKE SA lifetime
- ▶ ASCTCPIP-3115 Fixed known issue: Tcplp continues processing the message discarded by the policy check
- ▶ Added support for detailed policy violations logging

Module version 3.5.16

2022-06-10

- ▶ ASCTCPIP-3035 Fixed known issue: Tcplp does not correctly access the security database
- ▶ ASCTCPIP-3044 Fixed known issue: Using trusted end-point certificate with sect571r1 public key allows bypassing authentication check



- ▶ ASCTCPIP-3047 Fixed known issue: IKE negotiation is not possible if the remote end-point certificate uses ECDSA with P-521 elliptic curve
- ▶ ASCTCPIP-3045 Fixed known issue: Possible out-of-bounds read when parsing the ID payload
- ▶ ASCTCPIP-3049 Fixed known issue: Possible read out-of-bounds on 16-bit architectures when parsing remote ID payload
- ▶ ASCTCPIP-3048 Fixed known issue: Possible read out-of-bounds when parsing the subject identifier in remote identification payload
- ▶ ASCTCPIP-3076 Fixed known issue: SA used before IKE AUTH Response is sent out
- ▶ Added support for introduction of new TcplpCtrl instances at PostBuild time

Module version 3.5.15

2022-02-18

- ▶ ASCTCPIP-2934 Fixed known issue: TCP data transmission without window check
- ▶ Added the support for IKE
- ▶ ASCTCPIP-2935 Fixed known issue: Undefined behavior on reception of IKEv2 INIT message with ID payload
- ▶ Added IPv4 address format checks when calling Tcplp_RequestIpAddrAssignment to replace an existing static address with a new one
- ▶ ASCTCPIP-2944 Fixed known issue: Tcplp_IsConnectionReady() obtains and uses an incorrect controller index
- ▶ ASCTCPIP-2966 Fixed known issue: Certificate trust anchor of a received chain is not properly recognized in IKEv2
- ▶ ASCTCPIP-2992 Fixed known issue: Tcp handles the FIN transmission, data transmission, and FIN acknowledgement during controller SHUTDOWN procedure incorrectly
- ▶ Improve measurement data counter handling
- ▶ Added the support for reporting security events to IdsM

Module version 3.5.14

2021-10-08

- ▶ Added support to handle different return values from Tcplp_CopyTxDataAPI
- ▶ Added Measurement data support update
- ▶ Added IPv6 address format checks when calling Tcplp_RequestIpAddrAssignment to replace an existing static address with a new one



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 `_TxConfirmation` with incorrect Length
- ▶ ASCTCPIP-432 Fixed known issue: `_TxConfirmation` might be called after `_TcpEvent(TCPIP_TCP_CLOSED)`



- ▶ ASCTCPIP-430 Fixed known issue: If an out of order FIN is received TCP calls 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 _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

- ▶ Detailed policy violations logging



3.3.7.3. Elektrobit-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 to 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

- ▶ 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

- ▶ 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 and dynamic security associations in a Security Association Database (SAD). Dynamic security associations are negotiated between the Tcplp and the remote host during the start up of the Tcplp through the Internet Key Exchange Protocol Version 2. Features supported by the IKEv2 implementation are listed below. 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.

Tcplp provides the API Tcplp_RequestIpSecMode() to dis/enable security between the Tcplp and the selected remote host. If enabled all SECURED policies configured for the remote host are activated and secured traffic is exchanged between the Tcplp and the remote host. If disabled the SECURED policies are treated as BYPASSED.

- ▶ 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.

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

The Tcplp supports the AES-CMAC-96 integrity algorithm for the Authentication Header.

- ▶ 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
- ▶ The HMAC Algorithm according to IETF RFC 4868

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



- ▶ HMAC-SHA-256-128

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`.

- ▶ The Anti-Replay Algorithm according to IETF RFC 6479

The Tcplp implements Anti-Replay Algorithm for the Authentication Header without the need for bit shifting and it reduces the number of times an anti-replay window is adjusted.

- ▶ Extended Sequence Numbers for AH according to IETF RFC 4302

The Tcplp uses Extended Sequence Numbers for detecting replay attacks and for dynamic security associations negotiated through IKEv2 per default. Note: 32 bit sequence numbers are not supported.

- ▶ Recommended algorithms for Authentication Header (AH) according to RFC 8221

The Tcplp supports the following AH Authentication Algorithms :

- ▶ AES_128_GMAC
- ▶ AES_256_GMAC
- ▶ HMAC_SHA2_256_128

- ▶ IPsec features deviations to the RFC are listed in [Section 3.3.7.4, “Deviations”](#).

- ▶ Internet Key Exchange Protocol Version 2 (IKEv2)

- ▶ Internet Key Exchange Protocol Version 2 (IKEv2) according to IETF RFC 7296

- ▶ Tcplp supports the establishment of IKE SAs between the Tcplp and a configurable amount of remote hosts through IKE_SA_INIT/IKE_AUTH exchanges.
- ▶ Tcplp supports the establishment of a single IPsec Security Association per IKE security association through the IKE_AUTH exchange.
- ▶ Tcplp supports the re-authentication of the IKE SA (creating a new IKE SA from scratch by using IKE_SA_INIT/IKE_AUTH exchanges) by the initiator or responder after a configurable amount of time. The new IKE SA is created in parallel to the existing IKE SA and then the old IKE SA is deleted (Make-Before-Break principle)
- ▶ Tcplp supports configurable life time for the IKE security association and deletion after life time expires.
- ▶ Tcplp provides the API `Tcplp_RequestIpSecMode()` to dis/enable IKE between the Tcplp and the selected remote host.

- ▶ Window size according to IETF RFC 7296

The Tcplp supports an IKE window size of one.



- ▶ ID types to identify a remote host according to IETF RFC 7296

The Tcplp supports the following ID types in the configuration of the connection table and the Identification Payload:

- ▶ ID_IPV4_ADDR
- ▶ ID_IPV6_ADDR
- ▶ ID_DER_ASN1_DN
- ▶ Certificate Encoding of supported according to IETF RFC 7296

The Tcplp supports the following Certificate Encoding of Certificates in the CERT payload:

- ▶ X.509 Certificate - Signature
- ▶ Elliptic Curve Cryptography (ECC) groups according to IETF RFC 5903

The Tcplp supports the following Diffie-Hellman Group Transforms for use in the Internet Key Exchange version 2 (IKEv2) protocols:

- ▶ 256-Bit Random ECP Group
- ▶ 384-Bit Random ECP Group
- ▶ Signature Authentication in the Internet Key Exchange Version 2 according to RFC 7427

The Tcplp supports the following signature algorithms to generate the signature :

- ▶ ECDSA-with-SHA256
- ▶ ECDSA-with-SHA384
- ▶ ECDSA-with-SHA512
- ▶ Authentication methods for the authentication payload according to RFC 7427 and RFC 7296

The Tcplp supports the following authentication methods:

- ▶ Shared Key Message Integrity Code according to RFC 7296
- ▶ Digital Signature according to RFC 7427
- ▶ AES-CBC Cipher Algorithm according to RFC 7296 and RFC 3602

The Tcplp supports the following AES-CBC algorithm and key sizes for the Encryption Algorithm Transform in IKEv2:

- ▶ ENCR_AES_CBC with key size 128bit
- ▶ ENCR_AES_CBC with key size 256bit
- ▶ The HMAC Algorithm as a Pseudorandom Function in IKEv2 according to IETF RFC 4868

The Tcplp supports the following HMAC algorithm for the Pseudorandom Function Transform in IKEv2:

- ▶ PRF_HMAC_SHA2_256



- ▶ PRF_HMAC_SHA2_384
- ▶ The HMAC Algorithm for authentication and integrity verification in IKEv2 according to IETF RFC 4868

The Tcplp supports the following HMAC algorithm for the Integrity Algorithm Transform in IKEv2:

- ▶ AUTH_HMAC_SHA2_256_128
- ▶ AUTH_HMAC_SHA2_384_192
- ▶ The Traffic Selector Type in IKEv2 according to IETF RFC 7296

The Tcplp supports the following Traffic Selector Types in IKEv2:

- ▶ TS_IPV4_ADDR_RANGE
- ▶ TS_IPV6_ADDR_RANGE
- ▶ Message Fragmentation for IKEv2 according to IETF RFC 7383

The Tcplp supports to enable the IKEv2 message fragmentation and reassembly with the Encrypted and Authenticated Fragment Payload if the message exceeds the MTU

- ▶ GCM Algorithm according to RFC 4106

The Tcplp supports the following GCM Encryption Transforms for use in the Internet Key Exchange version 2 (IKEv2) protocols:

- ▶ AES-GCM with 16-octet ICV and 256 bit key
- ▶ AES-GCM with 12-octet ICV and 256 bit key
- ▶ Recommended algorithms in the Internet Key Exchange Version 2 according to RFC 8247

The Tcplp supports the following encryption algorithm :

- ▶ ENCR_AES_CBC with 128 bit key
- ▶ ENCR_AES_CBC with 256 bit key
- ▶ ENCR_AES_GCM_16 with 256 bit key

The Tcplp supports the following pseudorandom functions :

- ▶ PRF_HMAC_SHA2_256

The Tcplp supports the following integrity algorithm :

- ▶ AUTH_HMAC_SHA2_256_128

The Tcplp supports the following Diffie Hellman groups :

- ▶ 256-bit random ECP

The Tcplp supports the following authentication methods :

- ▶ Shared Key Message Integrity Code



- ▶ Digital Signature

The Tcplp supports the following hash functions for IKEv2 digital signature :

- ▶ SHA2-256
- ▶ SHA2-384
- ▶ SHA2-512

The Tcplp supports the following authentication for IKEv2 digital signature :

- ▶ ecdsa-with-sha256
- ▶ IKEv2 features deviations to the RFC are listed in [Section 3.3.7.4, “Deviations”](#).

- ▶ 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 (SrcIp32,DestIp32,Id16,Protocol8) as unique reassembling ID, whereas IPv6 omits the protocol, i.e.: only uses (SrcIp128,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 Tcplp 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_TcpIp_00077`, `ECUC_TcpIp_00078`, `ECUC_TcpIp_00079`, `ECUC_TcpIp_00080`, `ECUC_TcpIp_-00099`, `ECUC_TcpIp_00103`, `ECUC_TcpIp_00114`, `ECUC_TcpIp_00157`, `ECUC_TcpIp_00158`, `ECUC_TcpIp_00159`, `ECUC_TcpIp_00160`, `ECUC_TcpIp_00161`, `ECUC_TcpIp_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>_TcpConnect-ed() 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`
- `TcplpAutoIpConfig`
- `TcplpDhcpConfig`
- `TcplpIpV6DhcpConfig`
- `TcplpIpV6NdpConfig`

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 resetting 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 TcplpV4ArpPacketFilter 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:



- TcpIpDhcpV6CnfDelayMax

- TcpIpDhcpV6CnfDelayMin

- TcpIpDhcpV6InfDelayMax

- TcpIpDhcpV6InfDelayMin

- TcpIpDhcpV6SolDelayMax

- TcpIpDhcpV6SolDelayMin

Requirements:

ECUC_TcpIp_00116, ECUC_TcpIp_00117, ECUC_TcpIp_00118, ECUC_TcpIp_00119, ECUC_TcpIp_-
00120, ECUC_TcpIp_00121

- [IPV6] Dynamic reconfiguration of MTU via Router Advertisements is not supported

Description:

The TcpIp 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 TcpIp does not update the hop limit according to the value received in a router advertisement, the configured hop limit (TcpIpUdpTtl, TcpIpTcpTtl, TcpIpIcmpTtl, TcpIpIcmpV6HopLimit) is used instead.

Requirements:

SWS_TCPIP_00164, SWS_TCPIP_00157

- [IPV6] Dynamic reconfiguration of reachable time via Router Advertisements is not supported

Description:

The TcpIp does not update the reachable time according to the value received in a router advertisement, the configured reachable time (TcpIpNdpDefaultReachableTime) 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:

- TcplpNdpDynamicHopLimitEnabled
- TcplpNdpDynamicMtuEnabled
- TcplpNdpDynamicReachableTimeEnabled
- TcplpNdpDynamicRetransTimeEnabled
- TcplpNdpAddressResolutionUnreachabilityDetectionEnabled
- TcplpNdpMinRandomFactor
- TcplpNdpMaxRandomFactor
- TcplpNdpDefaultReachableTime
- TcplpNdpNeighborUnreachabilityDetectionEnabled
- TcplpNdpRandomReachableTimeEnabled

Requirements:

ECUC_Tcplp_00146, ECUC_Tcplp_00147, ECUC_Tcplp_00148, ECUC_Tcplp_00145, ECUC_Tcplp_-00091, ECUC_Tcplp_00134, ECUC_Tcplp_00135, ECUC_Tcplp_00130, ECUC_Tcplp_00136, ECUC_Tcplp_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 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

- ▶ [IPv6] 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 gener-



ate 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

- ▶ [IPv6] 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] 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



- ▶ [IPv6] 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

- ▶ [IPv6] 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

- ▶ [IPsec] 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, 4302, 4543, 5282, 4106.

- ▶ [IPsec] 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.

- ▶ [IPsec] 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.

- ▶ [IPsec] 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.

- ▶ [IPsec] 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.

- ▶ [IPsec] 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.

- ▶ [IPsec] 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.

- ▶ [IPsec] 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.

- ▶ [IPsec] Certain HMAC algorithms for AH defined in IETF RFC 4868 are not supported



Description:

Tcplp does not support the following HMAC integrity algorithm for AH

- ▶ HMAC-SHA-384-192
- ▶ HMAC-SHA-512-256

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4868.

- ▶ [IPsec] Certain GMAC algorithms defined in IETF RFC 4543 are not supported

Description:

Tcplp does not support the following GMAC integrity algorithm for AH

- ▶ AUTH_AES_192_GMAC

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4543.

- ▶ [IPsec] GMAC Initialization Vector can repeat

Description:

Tcplp allows use of GMAC with statically configured keys, without re-keying IV can repeat

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4543.

- ▶ [IPsec] 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.

- ▶ [IPsec] 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.

- ▶ [IPsec] 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.

- ▶ [IPsec] Tcplp does not support certain identifier types

Description:

Tcplp 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

Tcplp identifies a remote host by an IP address only.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ [IPsec] Tcplp does not support the Sequence Counter Overflow in SAD

Description:

Tcplp 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.

- ▶ [IPsec] Tcplp does not send ICMP Error Message when received frame is discarded

Description:

Tcplp 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.

- ▶ [IPsec] 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.

- ▶ [IPsec] 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.

- ▶ [IKEv2] Certain ECP Groups defined in IETF RFC 5903 are not supported

Description:

Tcplp does not support the following Diffie-Hellman Group Transforms for IKEv2

- ▶ 521-bit Random ECP Group

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 5903.

- ▶ [IKEv2] Certain algorithms defined in IETF RFC 8247 are not supported

Description:

Tcplp does not support the following encryption algorithms for IKEv2

- ▶ ENCR_CHACHA20_POLY1305
- ▶ ENCR_AES_GCM_16 with 128 bit key

Tcplp does not support the following pseudorandom function for IKEv2

- ▶ PRF_HMAC_SHA2_512
- ▶ PRF_HMAC_SHA1

Tcplp does not support the following integrity algorithm for IKEv2

- ▶ AUTH_HMAC_SHA2_512_256
- ▶ AUTH_HMAC_SHA1_96

Tcplp does not support the following Diffie-Hellman group for IKEv2

- ▶ 2048-bit MODP Group

Tcplp does not support the following authentication method for IKEv2

- ▶ RSA Digital Signature

Tcplp does not support the following digital signature authentication method for IKEv2

- ▶ RSASSA-PSS with SHA-256

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 8247.

- ▶ [IKEv2] Certain algorithms defined in IETF RFC 8221 are not supported

Description:



Tcplp does not support the following algorithms for AH authentication

- ▶ HMAC_SHA2_512_256
- ▶ HMAC_SHA1_96
- ▶ AES_XCBC_96
- ▶ AUTH_NONE

Tcplp does not support any encryption or authentication algorithms for ESP

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 8221.

- ▶ [IKEv2] Certain algorithms defined in IETF RFC 7427 are not supported

Description:

Tcplp does not support the following algorithms for signature authentication

- ▶ ECDSA-with-SHA1
- ▶ RSA-with-SHA1
- ▶ RSA-with-SHA256
- ▶ RSA-with-SHA384
- ▶ RSA-with-SHA512
- ▶ DSA-with-SHA1
- ▶ DSA-with-SHA256
- ▶ RSASSA-with-EmptyParameters
- ▶ RSASSA-with-DefaultParameters
- ▶ RSASSA-with-SHA256

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7427.

- ▶ [IKEv2] Repeated Authentication in IKEv2 Protocol defined in RFC 4478 is not supported

Description:

AUTH_LIFETIME notification is not transmitted when Tcplp is responder and is ignored when Tcplp is initiator

Rationale:



Tcplp does not use EAP and Configuration payloads meaning that creation of a new IKE SA can be initiated by either party (initiator or responder in the original IKE SA)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4478.

- ▶ [IKEv2] Additional transforms of known type are ignored

Description:

Received proposals are accepted as long as the locally configured transforms constitute a subset of the received transforms and all additional transforms are either ENCR, PRF, INTEG, D-H or ESN. E.g. if a proposal is received with combined cipher mode and integrity transform different from "NONE" the proposal is accepted as long as the locally configured proposal is a subset of the received proposal.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Certain HMAC Algorithm for Pseudorandom Function in IKEv2 according to IETF RFC 4868 are not supported

Description:

Tcplp does not support the following HMAC algorithm for the Pseudorandom Function Transform in IKEv2:

- ▶ PRF_HMAC_SHA2_512

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4868.

- ▶ [IKEv2] Certain HMAC Algorithm for authentication and integrity verification in IKEv2 according to IETF RFC 4868 are not supported

Description:

Tcplp does not support the following HMAC algorithm for the Integrity Algorithm Transform in IKEv2:

- ▶ AUTH_HMAC_SHA2_512_256

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4868.

- ▶ [IKEv2] Re-keying of IPsec Security Association not supported

Description:



Tcplp does not allow to re-key an IPsec security association through the CREATE_CHILD_SA exchange after a configured amount of time, specified counter value such as number of received bytes or when the maximal sequence number is reached. The life time of an IPSec security association is equal to the life time of the IKE security association.

If a CREATE_CHILD_SA request is received which requests re-keying of the IPSec SA the Tcplp will response with a NO_ADDITIONAL_SAS notification.

Rationale:

An IPsec security association can be refreshed through the re-authentication of the IKE security association. (creating a new IKE SA from scratch by using IKE_SA_INIT/IKE_AUTH exchanges)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301, 7296.

- ▶ [IKEv2] Re-keying of IKE Security Association not supported

Description:

Tcplp does not allow to re-key an IKE security association through the CREATE_CHILD_SA exchange after a configured amount of time.

If a CREATE_CHILD_SA request is received which requests re-keying of the IKE SA the Tcplp will response with a NO_ADDITIONAL_SAS notification.

Rationale:

An IKE security association can be refreshed through the re-authentication of the IKE security association. (creating a new IKE SA from scratch by using IKE_SA_INIT/IKE_AUTH exchanges)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Certain Id types to identify a remote host in the connection table are not supported

Description:

Tcplp does not support the Id types to identify a remote host in IKEv2:

- ▶ DNS name (specific or partial)
- ▶ RFC 822 email address (complete or partially qualified)
- ▶ Key ID (exact match only)

Requirements:



No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ [IKEv2] Only Key Exchange version 2 (IKEv2) is supported

Description:

Tcplp does not support any version of the Internet Key Exchange other than 2. In particular version 1 (IKEv1) defined in IETF RFC 4109 is not supported.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301, 7296.

- ▶ [IPsec] Anti-replay service not configurable for Security Association

Description:

- ▶ For dynamically configured security association (through IKEv2) the Anti-replay service is always turned on per default and cannot be turned off.
- ▶ For manually configured security association the Anti-replay service is always turned off per default and cannot be turned on.

Rationale:

Manually configured security association shall not use anti-reply and Extended sequence number because the synchronization of the security association cannot be guaranteed (e.g. one of the two hosts might restart and reset the sequence number counter)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ [IPsec] Extended Sequence Number (ESN) not configurable for Security Association

Description:

- ▶ Tcplp supports 64 bit sequence number only, 32 bit seq numbers are not supported.
- ▶ For dynamically configured security association (through IKEv2) the ESN is always turned on per default and cannot be turned off.
- ▶ For manually configured security association the ESN is always turned off per default and cannot be turned on.

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.

- ▶ [IKEv2] IKE notification INVALID_SELECTORS not supported

Description:

The Tcplp does not send the IKE notification INVALID_SELECTORS to the sender (IPsec peer), which indicates that the received packet was discarded because of failure to pass selector checks. If the Tcplp receives such a IKE notification, e.g. in an INFORMATIONAL exchange the notification is ignored.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- ▶ [IPsec] Synchronization due to Significant Packet Loss not supported

Description:

If there is an undetected packet loss of 2^{32} or more consecutive packets on a single SA, then the transmitter and receiver will lose synchronization of the high-order bits. The Tcplp will not try re-synchronize the sequence number.

Rationale:

The possibility of data loss in a vehicle is very low. IKE will detect that a remote host is not reachable anymore through Dead Peer Detection before 2^{32} packets are lost and will try to re-initiate the creation of a new IKE SA.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4302 and RFC 6479.

- ▶ [IKEv2] Mixed preshared key and certificate authentication

Description:

When host 1 uses preshared key authentication and host 2 uses certificate authentication method "digital signature" then only one SIGNATURE_HASH_ALGORITHMS notify is sent from the host 1 to host 2 containing the supported signature hash algorithms. No SIGNATURE_HASH_ALGORITHMS notify is sent from host 2 to host 1.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7427.

- ▶ [IKEv2] Certain algorithms defined in IETF RFC 5282 and RFC 4106 are not supported

Description:



Tcplp does not support the following GCM encryption algorithms for IKEv2

- ▶ AES-GCM with 16-octet ICV and 128 bit key
- ▶ AES-GCM with 16-octet ICV and 192 bit key
- ▶ AES-GCM with 12-octet ICV and 128 bit key
- ▶ AES-GCM with 12-octet ICV and 192 bit key
- ▶ AES-GCM with 8-octet ICV and 128 bit key
- ▶ AES-GCM with 8-octet ICV and 192 bit key
- ▶ AES-GCM with 8-octet ICV and 256 bit key

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 5282, IETF RFC 4106.

- ▶ [IKEv2] Extensible Authentication Protocol not supported

Description:

Tcplp does not support authentication using Extensible Authentication Protocol (EAP)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] IP Compression not supported

Description:

Tcplp does not support IP Compression (IPComp)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Configuration Payload not supported

Description:

TX: Tcplp does not transmit Configuration Payloads. RX: When receiving an INFORMATIONAL request with Configuration Payloads (CP) an empty INFORMATIONAL response is sent back.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Creating additional IPsec Security Association not supported

Description:



Tcplp does not allow creating additional IPsec security association through the CREATE_CHILD_SA exchange. Only one IPsec security association is created in the AUTH exchange.

If a CREATE_CHILD_SA request is received Tcplp will response with a NO_ADDITIONAL_SAS notification.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] IKE cookies not supported

Description:

Tcplp does not support the use of cookies in the IKE exchange and is not protected against flooding attacks

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] IKE address allocation not supported

Description:

Tcplp does not support address allocation to an IPsec Remote Access Client (IRAC) trying to tunnel into a network protected by an IPsec Remote Access Server (IRAS)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] IKE ignores messages outside of an IKE SA

Description:

To avoid flooding and reduce number of messages Tcplp discards messages outside of an IKE SA

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] IKE notification SET_WINDOW_SIZE not supported

Description:

The Tcplp does not send the IKE notification SET_WINDOW_SIZE indicating the sending endpoint is capable of keeping state for multiple outstanding exchanges

Tcplp as Initiator will never indicate it supports window size greater than 1 and will never send multiple requests before receiving response to the first.



If Tcplp as responder receives SET_WINDOW_SIZE indicating the remote host supports windows size greater than 1 it will never send multiple requests before receiving response to the first.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] No IKE SA is created without an associated child SA

Description:

If no child SA is created during the AUTH exchange due to e.g. NO_PROPOSAL_CHOSEN the underlying IKE SA will be deleted since only one child SA is supported per IKE SA and there is no way to create additional child SA via CREATE_CHILD_SA exchange.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Only a complete match of the traffic selectors is supported

Description:

When negotiating a Child SA during the IKE_AUTH exchange the traffic selectors contained in the IKE_AUTH response must match the traffic selectors in the IKE_AUTH request completely. No narrowing shall be done by the responder. In case a complete match is not found an TS_UNACCEPTABLE notify is sent in the IKE_AUTH response.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] IP Compression is not supported

Description:

IP Compression is not supported by Tcplp. IPCOMP_SUPPORTED notifications are never send and ignored on receipt.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] INVALID_SPI notification is not supported

Description:

TX: Tcplp does not transmit INVALID_SPI notification when AH packet with unrecognized SPI is received. The packet with unrecognized SPI in AH header is silently dropped. RX: INVALID_SPI notification is considered an error condition and the IKE SA is closed



Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] INVALID_IKE_SPI notification is not supported

Description:

TX: Tcplp does not transmit INVALID_IKE_SPI notification if IKE message packet arrives on port 500 with an unrecognized IKE SPI. The message with unrecognized SPI in IKE header is silently dropped. RX: INVALID_IKE_SPI notification is considered an error condition and the IKE SA is closed

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] INVALID_MAJOR_VERSION notification is not supported

Description:

TX: Tcplp does not transmit INVALID_MAJOR_VERSION notification when receiving an IKE message with major version number different from 2. The message with invalid major version in IKE header is silently dropped. RX: INVALID_MAJOR_VERSION notification is considered an error condition and the IKE SA is closed

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] INVALID_MESSAGE_ID notification is not supported

Description:

TX: Tcplp does not transmit INVALID_MESSAGE_ID notification when an IKE Message ID outside the supported window is received. The received message with invalid message ID is silently dropped. RX: INVALID_MESSAGE_ID notification is considered an error condition and the IKE SA is closed

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] UNSUPPORTED_CRITICAL_PAYLOAD notification is not supported

Description:

Critical unsupported payloads are not expected from the remote host and their presence is considered an error condition and IKE SA is closed. Due to security concerns there is no UNSUPPORTED_CRITICAL_PAYLOAD notification send in the response in order to mitigate the possibility of flooding attacks. TX: Tcplp does not transmit UNSUPPORTED_CRITICAL_PAYLOAD notification when an IKE Message is re-



ceived with unrecognized payload type whose critical flag is set. RX: UNSUPPORTED_CRITICAL_PAYLOAD notification is considered an error condition and the IKE SA is closed

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] TEMPORARY_FAILURE notification is not supported

Description:

TX: Tcplp does not transmit TEMPORARY_FAILURE notification when an IKE Message is received that cannot be completed due to a temporary condition RX: TEMPORARY_FAILURE notification is considered an error condition and the IKE SA is closed

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] SINGLE_PAIR_REQUIRED notification is not supported

Description:

TX: Tcplp does not transmit SINGLE_PAIR_REQUIRED notification when an IKE Message is received with TS payload that contains more than a single pair. RX: SINGLE_PAIR_REQUIRED notification is considered an error condition and the IKE SA is closed

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] CHILD_SA_NOT_FOUND notification is not supported

Description:

TX: Tcplp does not transmit CHILD_SA_NOT_FOUND notification when an CREATE_CHILD_SA request is received for rekeying an non-existant child SA. Instead all CREATE_CHILD_SA requests are responded to with NO_ADDITIONAL_SAS. RX: CHILD_SA_NOT_FOUND notification is considered an error condition and when received in an response the IKE SA is closed

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Only port 500 is open for IKE exchange

Description:

Ike exchange can only occur over port 500. Messages received at port 4500 are silently dropped



No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] No replicated entities are allowed

Description:

One-to-one correspondence between identities and hosts.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Identification of IKE SAs

Description:

A given IKE SA is not identified only by the local SPI as recommended by the Rfc but is identified by the initiator SPI + remote IP address for init exchanges and initiator SPI + responder SPI + remote IP address for other exchanges.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Identification of Child SAs

Description:

A given Child SA is not identified only by the pair of SAs as recommended by the Rfc but is identified by the single SA that have remote and local SPI.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Transmission of Delete AH request

Description:

Tcplp does not transmit Delete AH INFORMATIONAL Request because CHILD SA will be removed with an IKE SA

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Transforms with restricted key space

Description:

Tcplp does not support transform algorithms for which not all values are valid keys.



Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Transport Mode NAT Traversal is not supported

Description:

The current implementation does not support NAT Traversal

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Transform Attribute

Description:

Only the "key length" attribute is currently supported.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Diffie-Hellman transform with ID "NONE"

Description:

A Diffie-Hellman transform with ID "NONE" (numerical value 0) is never offered nor accepted in an IKE SA proposal.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Supported Identification IDs

Description:

The following ID types are the only ones supported in Identification payloads: ID_IPV4_ADDR, ID_IPV6_ADDR, ID_DER ASN1_DN

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Supported Certificate Encoding

Description:

The Tcplp does only support the following Certificate Encoding of Certificates in the CERT payload:



- ▶ X.509 Certificate - Signature (value 4)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Supported Authentication Method

Description:

The Tcplp does only support the following Authentication Method in the AUTH payload:

- ▶ Shared Key Message Integrity Code (value 2)
- ▶ Digital Signature (14)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] No logging of unrecognized notification types

Description:

Unrecognized notify types of notify payloads in any received IKE request or response are not logged.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Unsupported notification types

Description:

The following notification types are handled as unrecognized by the current implementation:

- ▶ INTERNAL_ADDRESS_FAILURE
- ▶ FAILED_CP_REQUIRED (see deviation "Configuration Payload not supported")
- ▶ ADDITIONAL_TS_POSSIBLE (see deviation "Only a complete match of the traffic selectors is supported")
- ▶ IPCOMP_SUPPORTED (see deviation "IP Compression not supported")
- ▶ NAT_DETECTION_SOURCE_IP (see deviation "Transport Mode NAT Traversal is not supported")
- ▶ NAT_DETECTION_DESTINATION_IP (see deviation "Transport Mode NAT Traversal is not supported")
- ▶ COOKIE (see deviation "IKE cookies not supported")
- ▶ HTTP_CERT_LOOKUP_SUPPORTED (see deviation "Supported Certificate Encoding")
- ▶ REKEY_SA (see deviation "Re-keying of IPsec Security Association not supported")



- ▶ **ESP_TFC_PADDING_NOT_SUPPORTED** (see deviation "Tcplp does not support ESP")
- ▶ **NON_FIRST_FRAGMENTS ALSO** (see deviation "Creating additional IPsec Security Association not supported")

TX: Tcplp does not transmit these notification types RX: error notifications are considered fatal (IKE SA is closed), status notifications are ignored

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] The vendor ID payload is not supported

Description:

The vendor ID payload is never sent and ignored on receipt.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] MODP Diffie-Hellman groups for IKEv2 are not supported

Description:

Tcplp does not support MODP Diffie-Hellman groups

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Error notifications in request messages

Description:

All error notifications in request messages are considered to be fatal error conditions causing the IKE SA to be closed.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Address narrowing and port narrowing is not supported

Description:

Tcplp Does not support any type of narrowing.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.



- ▶ [IKEv2] Certificate validity check is not supported

Description:

Tcplp does not support certificate validity check, i.e. if the certificate has expired.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Immediate reconnection

Description:

If the INIT request is responded by an error notify this leads to an immediate reconnection attempt in case there is currently no other IKE SA active for the given connection. In case there is an active connection the IKE SA is closed without further reconnection attempts. The idea of this "never give up" approach is that the implementation should always try to establish a connection.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] No error handling for INFORMATIONAL messages

Description:

If an error is detected during processing of an IKE INFORMATIONAL message or an error notification other than AUTHENTICATION_FAILED is present in the received INFORMATIONAL message the message is dropped without further error handling.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Transmitted delete request messages

Description:

INFORMATIONAL IKE SA Delete request are only send 1. after reauthentication was succesfull (and the old IKE SA is deleted) 2. after receiving a CHILD SA delete message 3. after stopping an established connection (by calling Tcplp_IkeV2_stop)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Multiple identities per host not supported

Description:



The IDr payload in AUTH requests are ignored since ECUs are assumed to have only one identity.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] IP address ranges in traffic selectors not supported

Description:

TcpIp module does not support configuration of Ip address ranges for traffic selectors. Instead, IpSec connection remote IP address is set for both Starting Address and Ending Address fields for traffic selector.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] Ports of traffic selectors using ANY protocol can be configured.

Description:

TcpIp supports configuration of port range in Traffic Selectors using protocol ANY.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] IKE Responder forgetting the response not supported

Description:

TcpIp Ike does not support forgetting the response after some time. Currently the response is forgotten if new message is transmitted or the connection is restarted.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] IKE SAs are not closed with invalid Message ID

Description:

TcpIp Ike does not support closing or rekeying IKE SA when Message ID overflows.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- ▶ [IKEv2] IKE can't receive payloads before Encrypted

Description:



Tcplp Ike does not support transmitting and receiving payloads before Encrypted and Authenticated or Encrypted Fragmented payload.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296, IETF RFC 7383.

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 `TcpIpLocalAddr`.

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 `TcpIpNdpSlaacOptimisticDadEnabled` is enabled, it can only be reassigned by reinitialization of the `TcpIp`.

▶ Parsing of IKE and IpSec proposals

Description:

The received list of proposals in `IKE_SA_INIT` requests and `IKE_AUTH` requests is parsed sequentially until a match with the locally configured proposal of highest priority is found. Proposals that come after the match are not parsed and thereby any malformed proposal that might come after the match is ignored.

▶ Supported PRF transform

Description:

The PRF transforms that can be negotiated are `PRF_HMAC_SHA2_256` and `PRF_HMAC_SHA2_384`

▶ Multiple 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.

▶ Handling of IKE Reauthentication



Description:

EAP and Configuration payloads not supported by the current implementation, because of this there is no difference between reauthentication and a new IKE exchange. Reauthentication can be started by either initiator or responder.

- ▶ 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.

- ▶ Deleting Established IKE SA without transmitting a DELETE request

Description:

If address using IKE is abandoned due to being a duplicate, Ike SA will be deleted without sending a DELETE request.

If address using IKE is released, Ike SA will be deleted without sending a DELETE request.

If controller of the address using IKE is requested OFFLINE, Ike SA will be deleted without sending a DELETE request.

- ▶ TLS secured sockets close after TLS handshake

Description:

If a socket using TLS requested to close via Tcplp_Close without aborting, Tcplp will first notify TLS and wait for its approval.

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: 2.9.11.B567464
- ▶ Supplier: Elektrobit Automotive GmbH



3.3.8.1. Change log

This chapter lists the changes between different versions.

Module version 2.9.11

2022-10-12

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

Module version 2.9.10

2022-07-04

- ▶ Increase support for 504 PNCs
- ▶ ASCUDPNM-436 Fixed known issue: UdpNm might not compile if memory mapping is used with specific linker script options
- ▶ Implemented Support for Synchronized PNC shutdown.

Module version 2.9.9

2022-03-09

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

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_SoAdIfRxIndication 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 depedency 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. Elektrobit-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_GetUserData()` 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 number of PNCs from 56 to 504

Description:

Starting from AUTOSAR R20-11 requirements ECUC_UdpNm_00068 and ECUC_UdpNm_00069, UdpNmPnInfoLength range has changed from 1..7 to 1..63 and UdpNmPnInfoOffset range has changed from 1..31 to 1..63.

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.

- ▶ `UdpNmNodeID` shall be available only in case `UdpNmNodeIDEnabled` is set

Description:

Contrary to specifications `UdpNmNodeID` shall be available only in case `UdpNmNodeIDEnabled` 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_SoAdlfTxConfirmation 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

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

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

- ▶ Changes regarding to UdpNmPnFilterMaskByteIndex

Description:

Parameter `UdpNmPnFilterMaskByteIndex` can only be configured with a value between 0 and 62 since it represents the position within the mask byte arrays it shall always be smaller than `UdpNmPnInfoLength` which has the range 1-63.

Requirements:

UdpNm.R20-11.ECUC_UdpNm_00071

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, “EthIf module user guide”](#)
- ▶ [Section 4.6, “Sd module user guide”](#)
- ▶ [Section 4.7, “SoAd module user guide”](#)
- ▶ [Section 4.8, “Tcplp module user guide”](#)
- ▶ [Section 4.9, “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, cryptographically-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 4.8.2.5, “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 4.4.2, “Background Information”](#) provides an overview of the basic functionality of the DoIP.
- ▶ [Section 4.4.3, “Configuring the DoIP module”](#) 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 of [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”](#).

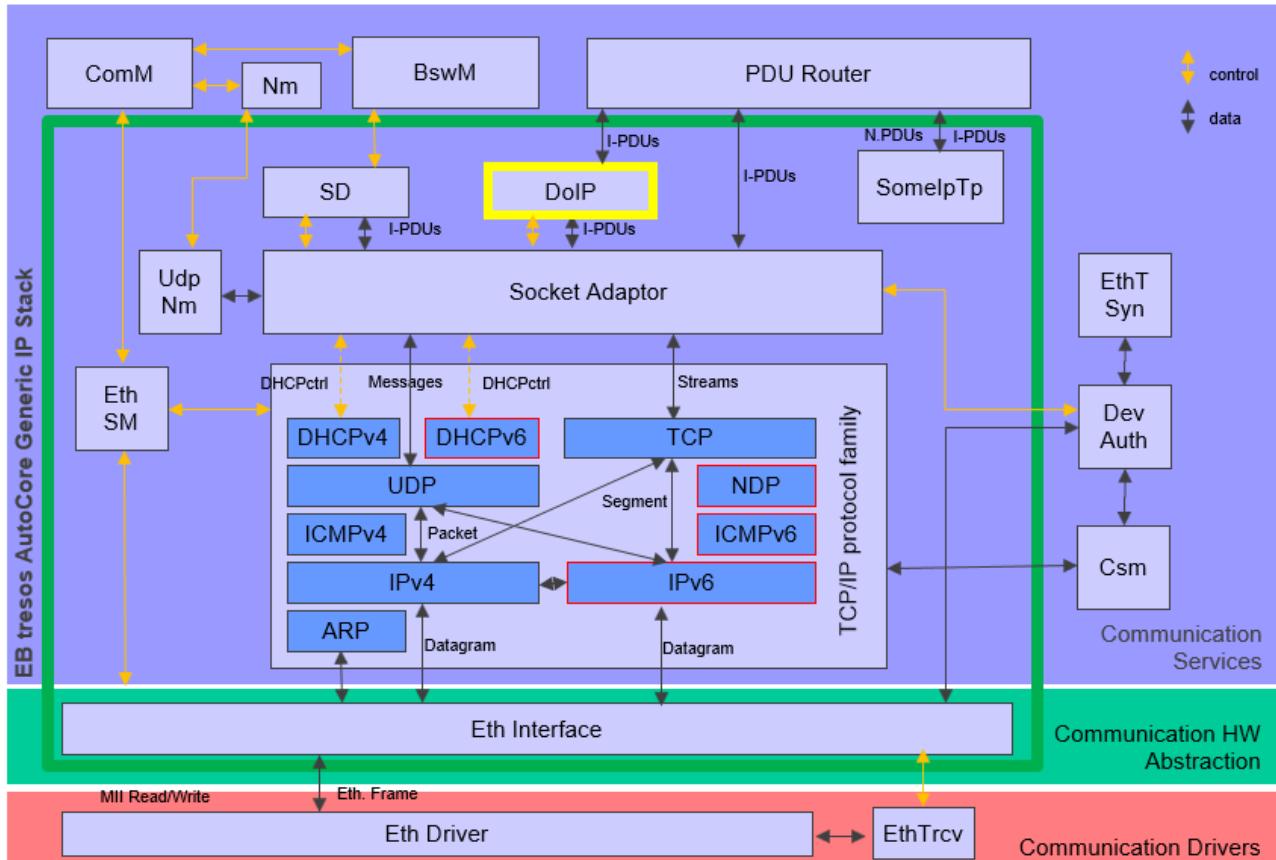


Figure 4.1. The AUTOSAR DoIP module



4.4.2.1. TCP closure APIs

DoIP is extended with a set of external APIs to allow an upper layer to disconnect testers by closing its TCP connections. Both closure methods, with FIN and RST flags, are supported.

These APIs enable the upper layer to interrupt low priority diagnostic communication when high priority diagnostic communication is required.

The following APIs are available:

- ▶ `Std_ReturnType DoIP_DisconnectTester(SoAd_SoConIdType SoConId, boolean Abort)`
- ▶ `Std_ReturnType DoIP_GetSoConIdFromRxPduId(PduIdType RxPduId, SoAd_SoConIdType* SoConIdPtr)`
- ▶ `Std_ReturnType DoIP_GetSoConIdFromTxPduId(PduIdType TxPduId, SoAd_SoConIdType* SoConIdPtr)`

The `SoConId` parameter in `DoIP_DisconnectTester()` is provided via routing activation callback functions when configured. Alternatively, it can be retrieved with `DoIP_GetSoConIdFromRxPduId()` or `DoIP_GetSoConIdFromTxPduId()` APIs.

The `Abort` parameter is propagated to `SoAd_CloseSoCon()` and has the following meaning:

- ▶ TRUE - Closes (resets) the TCP connection with RST flag.
- ▶ FALSE - Closes the TCP connection with FIN flag.

The actual connection closure is performed in the next main function call.

4.4.2.2. TCP closure with FIN and RST

The DoIP AUTOSAR specification R21-11 introduces closing of TCP connections with FIN. Additionally, for a proper closing of TLS-secured connections with CloseNotify, it is required that DoIP closes the connection with FIN.

In order to fulfill these requirements but also keep backward compatibility, DoIP is extended with a possibility to specify the way the connection shall be closed:

- ▶ 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, or when `DoIP_ActivationLineSwitchInactive()` is called.
- ▶ 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 except those described under soft closure, e.g. alive check or inactivity time-out.



4.4.2.3. Delay of local IP address release

When `DoIP_ActivationLineSwitchInactive()` is called, Tls requires a time delay between closing a connection and the release of its local IP address in order to allow Tls to send an encrypted CloseNotify.

4.4.3. Configuring the DoIP module

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.

4.4.3.1. Configuring TCP closure APIs

Enable this feature with configuration parameter `DoIPConnectionControl`.

4.4.3.2. Configuring TCP closure with FIN and RST

Enable the closure with FIN with the `DoIPEnableTcpClosureWithFIN` configuration parameter. When this parameter is disabled, then in all cases TCP connections are closed with RST flag.

4.4.3.3. Configuring a local IP address release delay



Configuring the local IP address release delay

Step 1

To enable the `DoIPLocalIPAddressReleaseDelay` configuration parameter, first enable the `DoIPEnableTcpClosureWithFIN` configuration parameter.



Step 2

In `DoIPLocalIPAddressReleaseDelay`, specify the time delay between closing a connection and the release of its local IP address. The value should cover the maximum possible sync or async encryption time during Tls closure, e.g. 100 ms.

If Tls is not used, you can set `DoIPLocalIPAddressReleaseDelay` to 0. This would release the local IP address in the next `DoIP_MainFunction()` call.

If Tls is used, the proposal for a proper ECU shutdown is as follows:

1. In DoIP configuration, enable `DoIPEnableTcpClosureWithFIN` and set the value of `DoIPLocalI-
PAddressReleaseDelay`.
2. Call `DoIP_ActivationLineSwitchInactive()`.
3. Wait for `BswM_EthSM_CurrentState(ETHSM_STATE_WAIT_ONLINE)`.
4. Call `ComM_RequestComMode(EthCh, COMM_NO_COMMUNICATION)`.
5. Wait for `BswM_EthSM_CurrentState(ETHSM_STATE_OFFLINE)`.
6. Reset ECU.

4.5. EthIf module user guide

4.5.1. Overview

This user guide provides you with EthIf-specific information:

- ▶ [Section 4.5.2, “Background Information”](#) explains the concepts of the EthIf module.
- ▶ [Section 4.5.3, “Configuring the EthIf module”](#) provides instructions on how to configure the EthIf mod-
ule.

For details on individual EthIf configuration parameters and APIs, see the descriptions provided in the EthIf module reference [Chapter 5, “ACG8 IP Stack module references”](#).

4.5.2. Background Information

4.5.2.1. Multiple device driver support

The EthIf module provides multiple device driver support for Eth and EthTrcv. For EthSwt, this feature is currently not implemented. Multiple device driver support is not available for AUTOSAR versions below 4.3.0.



Every `Eth` or `EthTrcv` driver that needs this support must have a valid `BswModuleDescription` file. This file shall be referenced by `EthIf` to provide `EthIf` with Vendor ID and VendorApiInfix. Each driver must have a unique combination of Vendor ID and/or VendorApiInfix.

For information on how to configure the feature, see [Section 4.5.3.1, “Configuring the multiple device driver support”](#).

4.5.3. Configuring the `EthIf` module

4.5.3.1. Configuring the multiple device driver support

Multiple device driver support is provided for `Eth` and `EthTrcv` drivers. It is currently not implemented for `EthSwt`. But you can modify the single `EthSwt` and its APIs as shown below.

Configure the mentioned containers only in the following two cases:

- ▶ if the support of multiple drivers is required, or
- ▶ if you use a driver that contains a Vendor ID and VendorApiInfix.

Each driver must have a unique combination of Vendor ID and/or VendorApiInfix.



Configuring the `EthIf` module

Step 1

For an `Eth` or `EthTrcv` driver that needs multiple drivers support, add an entry to the corresponding container:

- ▶ `Eth: EthIfEthControllerType`
- ▶ `EthTrcv: EthIfEthTrcvType`

Step 2

In `EthIfEthControllerBswmdImplementationRefs` and/or `EthIfEthTrcvBswmdImplementationRefs` respectively, reference the corresponding `Driver_Bswmd.arxml` file.

Step 3

In `EthIfEthCtrlTypeRef` and/or `EthIfEthTrcvTypeRef` respectively, reference the used `Eth` or `EthTrcv` driver.

Step 4

If you use an `EthSwt` driver with a Vendor ID/VendorApiInfix, add an entry to the `EthIfEthSwtType` container.



Step 5

In `EthIfEthSwtBswmdImplementationRefs`, reference the corresponding `Driver_Bswmd.arxml` file.

Step 6

In `EthIfEthSwtTypeRef`, reference the used `EthSwt` driver.

4.6. Sd module user guide

4.6.1. Overview

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

- ▶ [Section 4.6.2, “Background information”](#) explains the concepts of the `Sd` module.
- ▶ [Section 4.6.3, “Configuring the `Sd` module”](#) provides instructions on how to configure the `Sd` module.

4.6.2. 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”](#).

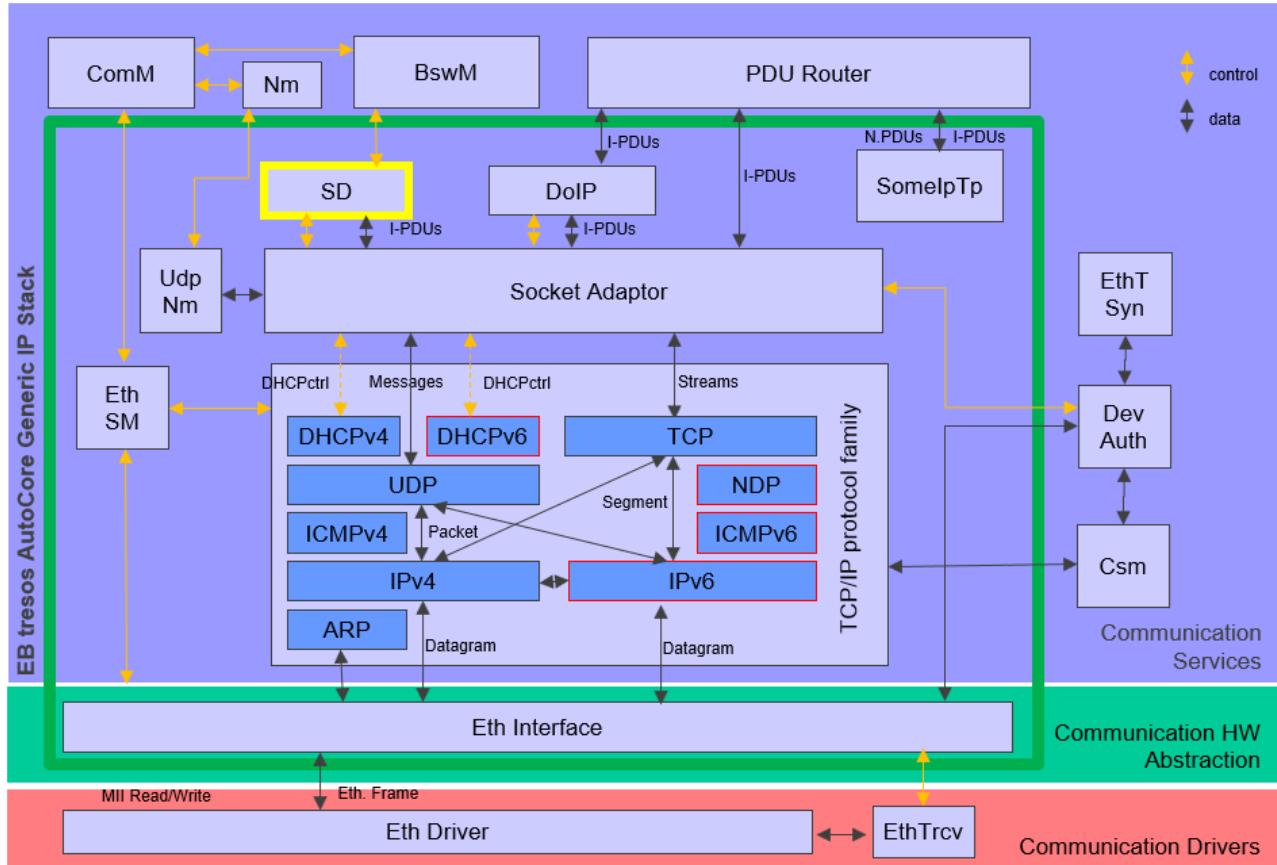


Figure 4.2. The AUTOSAR Sd module

4.6.3. 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.



4.7. SoAd module user guide

4.7.1. Overview

This chapter provides you with SoAd-specific information:

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

4.7.2. Background information

4.7.2.1. 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 4.7.3.1, “Configuring the N-PDU buffer pooling”](#).

4.7.2.1.1. 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
<ul style="list-style-type: none"> ▶ With the UDP socket in use AND ▶ Same SocketConnectionGroup AND ▶ Different SocketConnection 	The PDU is dropped.
<ul style="list-style-type: none"> ▶ 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

4.7.2.1.2. 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.

4.7.2.2. TLS and DTLS

The Socket Adaptor supports the protection of socket connection via Transport Layer Security (TLS) for TCP and Datagram Transport Layer Security (DTLS) for UDP. For TLS protection, each socket connection uses a separate TCP socket. Therefore, each TCP socket requires only one TLS connection. For DTLS, multiple socket connection share the same UDP socket. Therefore, each UDP socket requires multiple DTLS connections. The assignment of socket connection to DTLS connection is done at run-time.

The main use case of TLS is to protect DoIP and SOME/IP data communication. The main use case of DTLS is to protect SOME/IP data communication.

For information on how to configure TLS and DTLS protection, see [Section 4.7.3.2, “Configuring TLS and DTLS”](#).



4.7.3. Configuring the SoAd module

4.7.3.1. 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 `SoAdSocketnPduUdpTxBufferSizeMin` × `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/SoAdSocketnPduUdpTxBufferSizeMin` 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.

4.7.3.2. Configuring TLS and DTLS



Configuring the SoAd

The mapping of socket connections to `TlsConnection` can be configured via the references `SoAdTlsConnectionRef` and `SoAdDatagramTlsConnectionRef`. The mapping of UDP/TCP sockets to `TlsConnection` is done at run-time during socket connection establishment with `TcpIp_ChangeParameter()` and the handle IDs behind these references.



Step 1

For a TCP SoAdSocketConnectionGroup that requires TLS protection, enable SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection/SoAdTlsConnectionRef and reference a TlsConnection.

Step 2

For a UDP SoAdSocketConnectionGroup that requires DTLS protection, enable SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp/SoAdDataGramTlsConnectionRef and reference a TlsConnection suitable for DTLS.

The configuration of TlsConnection is not in the scope of this module.

4.8. TcpIp module user guide

4.8.1. Overview

This user guide provides you with TcpIp-specific information:

- ▶ [Section 4.8.2, “Background Information”](#) explains the concepts of the TcpIp module.
- ▶ [Section 4.8.3, “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”](#).

4.8.2. Background Information

4.8.2.1. Software hardening support

The TcpIp stack provides the following software hardening features:

- ▶ [TCP SYN cookies](#)
- ▶ [Unpredictable sequence numbers](#)

4.8.2.1.1. 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”](#).

4.8.2.1.2. 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”](#).

4.8.2.2. 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 4.8.3.2, “Configuring the permanent IP address storage to NVM RAM”](#).

4.8.2.3. 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 4.8.3.3, “Configuring the IPv4 Duplicate Address Detection \(ARP probing\)”](#).

4.8.2.4. 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 4.8.3.4, “Configuring the IPv6 Duplicate Address Detection”](#).

4.8.2.5. 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
	TCPIP_IKEV2_E_NOT_OK	201U
	TCPIP_IKEV2_E_NO_PROPOSAL_CHOSEN	203U



Protocol	Error code	Value
	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.

4.8.3. Configuring the `TcpIp` module

4.8.3.1. 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.



4.8.3.2. 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.



4.8.3.3. 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: Value 5
- TcpIpDhcpInitDelay: Value 1
- TcpIpIpv4DhcpAddrDefenseMechanism: Value TCPIP_DEFEND_ADDR
- TcpIpDhcpFQDNOptionEnabled: Enabled (checked)
- TcpIpDhcpIpv4DomainNameMaxSize: Value 11
- TcpIpDhcpConfigurableOptionsEnabled: Enabled (checked)
- TcpIpDhcpConfigurableOptionsEntriesMax: Value 7
- TcpIpDhcpConfigurableOptionsDataSizeMax: Value 23
- TcpIpDhcpArpProbingEnabled: Enabled (checked)
- TcpIpDhcpArpProbingType: Value 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.



4.8.3.4. 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 (checked)
 - TcpIpNdpSlaacOptimisticDadEnabled: Enabled (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 (checked)
 - TcpIpDhcpV6ConfigurableOptionsEntriesMax (1 -> 255): Value 3
 - TcpIpDhcpV6ConfigurableOptionsDataSizeMax (1 -> 65535): Value 10
 - TcpIpDhcpV6SlaacDadEnabled: Enabled (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`.

4.9. QoS Support user guide

4.9.1. 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 4.9.2, “Background information”](#) describes the basic concept of QoS guarantees with regard to latency for time-sensitive Ethernet communication.
- ▶ [Section 4.9.3, “Configuring QoS Support”](#) provides instructions on how to configure the QoS Support features in the ACG8 IP Stack modules.



4.9.2. 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”](#).

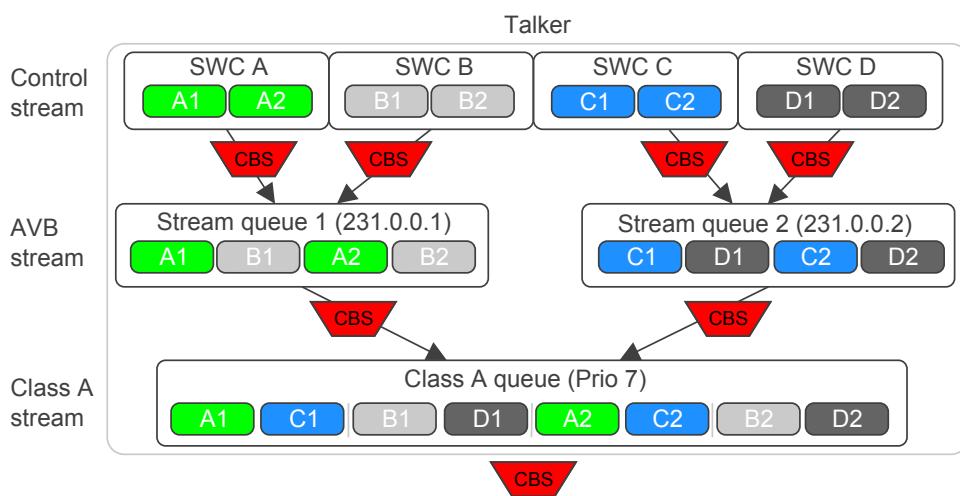


Figure 4.3. Sharing of an AVB stream

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.

4.9.3. 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.

4.9.3.1. 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.

4.9.3.2. 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 4.9.3.5, “Configuring the Operating System Module \(`Os`\)”](#) and [Section 4.9.3.6, “Configuring the Run-Time Environment Module \(`Rte`\)”](#)).

4.9.3.3. 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 4.9.3.1, “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 4.9.3.4, “Configuring the Socket Adaptor Module \(SoAd\)”](#)).

4.9.3.4. 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.

4.9.3.5. Configuring the Operating System Module (Os)

The `OsTask` provides the execution context (see [Section 4.9.3.6, “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.

4.9.3.6. 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
CommonPublishedInformation	1..1	Label: Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
DoIPConfigSet	1..1	This container contains the configuration parameters and sub containers of the AUTOSAR DoIP module.
DoIPGeneral	1..1	This container specifies the general configuration parameters of the DoIP module.
DoIPDefensiveProgramming	1..1	Label: Defensive Programming Options Parameters for defensive programming
PublishedInformation	1..1	Label: EB Published Information Additional published parameters not covered by CommonPublishedInformation container.

Parameters included	
Parameter name	Multiplicity
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT
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
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1
SwPatchVersion	1..1
ModuleId	1..1
VendorId	1..1
Release	1..1

Parameter Name	ArMajorVersion
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	ArMinorVersion
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
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ArPatchVersion
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	3
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMajorVersion
Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMinorVersion
Label	Software Minor Version
Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwPatchVersion
-----------------------	-----------------------



Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	26
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ModuleId
Label	Numeric Module ID
Description	Module ID of this module from Module List
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	173
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	VendorId
Label	Vendor ID
Description	Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	Release
Label	Release Information
Multiplicity	1..1
Type	STRING_LABEL
Default value	
Configuration class	PublishedInformation:

Origin	Elektrobit Automotive GmbH
---------------	----------------------------

5.2.1.2. DolPConfigSet

Containers included		
Container name	Multiplicity	Description
DolPChannel	1..65536	Configuration of one DolPChannel.
DolPCustomChannel	1..254	Configuration of DolPCustomChannels. This container can be enabled with DolPCustomPayload-TypeEnabled.
DolPConnections	1..1	Container contains all lower layer connection specific information, i.e. the single PDU References and Handle IDs to the SoAd.
DolPRoutingActivation	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.
DolPTester	1..255	This container describes the properties of the possible connectable Tester for the DolP entity.

Parameters included	
Parameter name	Multiplicity
DolPEid	0..1
DolPGid	0..1
DolPLogicalAddress	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:	VariantPostBuild
Origin	AUTOSAR_ECUC	



Parameter Name	DolPGid	
Description	Configured GID (Group ID of) for vehicle identification/vehicle announcement.	
Multiplicity	0..1	
Type	INTEGER	
Range	<=281474976710655 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPLogicalAddress	
Description	Describes the logical address of the DolP entity, i.e. the LA that will route diagnostic requests to the Dcm of the DolP entity.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65535 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.2.1.3. DolPChannel

Containers included		
Container name	Multiplicity	Description
DolPPduRRxPdu	1..1	This container contains the Rx PDUs to connect with the Rx PDUs of the PduR.
DolPPduRTxPdu	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.

Parameters included	
Parameter name	Multiplicity
DolPChannelSARef	1..1
DolPChannelTARef	1..1
DolPRoutingActivationType	1..1



Parameter Name	DolPChannelISARef	
Description	Reference to the DolPTester.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPChannelTARef	
Description	Reference to the target address.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPRoutingActivationType	
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. DolPPduRRxPdu

Parameters included		
Parameter name	Multiplicity	
DolPPduRRxPdul	1..1	
DolPPduRRxPduRef	1..1	



Parameter Name	DolPPduRRxPduld	
Description	The DolPPduRRxPduld is required by the API call DolP_CancelReceive.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65535 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPPduRTxPduRef	
Description	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.2.1.5. DolPPduRTxPdu

Parameters included		
Parameter name	Multiplicity	
DolPPduRTxPduld	1..1	
DolPPduRTxPduRef	1..1	

Parameter Name	DolPPduRTxPduld	
Description	The DolPPduRTxPduld is required by DolP_TpTransmit and DolP_CancelTransmit.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPPduRTxPduRef	
-----------------------	-------------------------	--



Description	Reference to the "global" PDU structure to allow harmonization of handle IDs in the COM-Stack.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.2.1.6. DolPCustomChannel

Containers included		
Container name	Multiplicity	Description
DolPPduRRxPdu	1..1	This container contains the Rx PDUs to connect with the Rx PDUs of the PduR.
DolPPduRTxPdu	1..1	This container contains the Tx PDUs to connect with the Tx PDUs of the PduR.

Parameters included	
Parameter name	Multiplicity
DolPTcpConnectionRef	1..1

Parameter Name	DolPTcpConnectionRef
Description	Reference to Tcp connection used for transmission of custom diagnostic messages.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild:
Origin	Elektrobit Automotive GmbH

5.2.1.7. DolPPduRRxPdu

Parameters included	
Parameter name	Multiplicity
DolPPduRRxPduld	1..1
DolPPduRRxPduRef	1..1



Parameter Name	DolPPduRRxPduld	
Description	Values should be consecutive after DolPPduRRxPdulds from DolPChannel	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65535 $>=\text{num:i}(\text{count(as:modconf('DolP')[1]/DolPConfigSet/*[1]/DolPChannel/*/DolP-PduRRxPdu/DolPPduRRxPduld))}$	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DolPPduRTxPduRef	
Description	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.2.1.8. DolPPduRTxPdu

Parameters included		
Parameter name	Multiplicity	
DolPPduRTxPduld	1..1	
DolPPduRTxPduRef	1..1	

Parameter Name	DolPPduRTxPduld	
Description	Values should be consecutive after DolPPduRTxPdulds from DolPChannel	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65535 $>=\text{num:i}(\text{count(as:modconf('DolP')[1]/DolPConfigSet/*[1]/DolPChannel/*/DolP-PduRTxPdu/DolPPduRTxPduld))}$	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	Elektrobit Automotive GmbH
---------------	----------------------------

Parameter Name	DolPPduRTxPduRef	
Description	Reference to the "global" PDU structure to allow harmonization of handle IDs in the COM-Stack.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.2.1.9. DolPConnections

Containers included		
Container name	Multiplicity	Description
DolPTargetAddress	1..65535	This container describes a possible TargetAddress that is supported by DolP.
DolPTcpConnection	2..255	This container describes a Tcp connection to the lower layer SoAd module.
DolPUdpConnection	1..255	This Container describes a UDP connection to the lower layer SoAd module.
DolPUdpVehicleAnnounce- ment	1..255	This container provides PDUs for UDP multicast vehicle announcements.

5.2.1.10. DolPTargetAddress

Parameters included	
Parameter name	Multiplicity
DolPTargetAddressValue	1..1

Parameter Name	DolPTargetAddressValue
Description	Valid Target Address of a DolP target address.
Multiplicity	1..1
Type	INTEGER
Range	<=65535



	>=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.2.1.11. DolPTcpConnection

Containers included		
Container name	Multiplicity	Description
DolPSoAdRxPdu	1..1	This container contains the Rx PDUs received by DolP.
DolPSoAdTxPdu	1..1	This container describes the Tx PDU sent via the SoAd.

Parameters included	
Parameter name	Multiplicity
DolPTcpConnectionSecurityRequired	1..1
DolPRequestAddressAssignment	1..1
DolPUpdateDhcpHostNameOption	1..1

Parameter Name	DolPTcpConnectionSecurityRequired
Description	Indicates if the associated TCP socket uses a secure connection (e.g. TLS).
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPRequestAddressAssignment
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	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	

5.2.1.12. DolPSoAdRxPdu

Parameters included		
Parameter name		Multiplicity
DolPSoAdRxPduld		1..1
DolPSoAdRxPduRef		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.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	



5.2.1.13. DolPSoAdTxPdu

Parameters included	
Parameter name	Multiplicity
DolPSoAdTxPduld	1..1
DolPSoAdTxPduRef	1..1

Parameter Name	DolPSoAdTxPduld	
Description	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.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPSoAdTxPduRef	
Description	Reference to the "global" PDU structure to allow harmonization of handle IDs in the COM-Stack.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.2.1.14. DolPUdpConnection

Containers included		
Container name	Multiplicity	Description
DolPSoAdRxPdu	1..1	This container contains the Rx PDUs received by DolP.
DolPSoAdTxPdu	1..1	This container describes the Tx PDU sent via the SoAd.

Parameters included	
Parameter name	Multiplicity
DolPRequestAddressAssignment	1..1
DolPUpdateDhcpHostNameOption	1..1



Parameter Name	DolPRequestAddressAssignment	
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	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	
DolPSoAdRxPduld	1..1	
DolPSoAdRxPduRef	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.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.2.1.16. DolPSoAdTxPdu

Parameters included		
Parameter name		Multiplicity
DolPSoAdTxPduld		1..1
DolPSoAdTxPduRef		1..1

Parameter Name	DolPSoAdTxPduld	
Description	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.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPSoAdTxPduRef	
Description	Reference to the "global" PDU structure to allow harmonization of handle IDs in the COM-Stack.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	



5.2.1.17. DolPUpdVehicleAnnouncement

Containers included		
Container name	Multiplicity	Description
DolPSoAdTxPdu	1..1	This container describes the Tx PDU sent via the SoAd.

Parameters included	
Parameter name	Multiplicity
DolPRequestAddressAssignment	1..1
DolPUpdateDhcpHostNameOption	1..1

Parameter Name	DolPRequestAddressAssignment	
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	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DolPUpdateDhcpHostNameOption	
Description	When set DolP module will update DHCP host name option related to this connection (starting with 'DolP-').	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.2.1.18. DolPSoAdTxPdu

Parameters included	
Parameter name	Multiplicity
DolPSoAdTxPduld	1..1



Parameters included

DoIPSoAdTxPduRef	1..1
----------------------------------	------

Parameter Name	DoIPSoAdTxPduld
Description	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.
Multiplicity	1..1
Type	INTEGER
Range	<=65535 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	DoIPSoAdTxPduRef
Description	Reference to the "global" PDU structure to allow harmonization of handle IDs in the COM-Stack.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.2.1.19. DoIPRoutingActivation

Containers included

Container name	Multiplicity	Description
DoIPRoutingActivationAuthenticationCallback	0..1	Container describes the Callback function to call on a Routing Activation Request for Authentication. If this container is configured but the DoIPRoutingActivationAuthenticationFunc parameter is not present, the DoIP module will use an RPort of ServiceInterface <RoutingActivation>_RoutingActivation with the name "CB<RoutingActivation>RoutingActivation". <RoutingActivation> is the ShortName of the DoIPRoutingActivation container.
DoIPRoutingActivationConfirmationCallback	0..1	Container describes the Callback function to call on a Routing Activation Request for Confirmation. If this container is



Containers included

		configured but the DoIPRoutingActivationConfirmationFunc parameter is not present the DoIP module will use an RPort of ServiceInterface <RoutingActivation>_RoutingActivation with the name "CB<RoutingActivation>RoutingActivation". <RoutingActivation> is the ShortName of the DoIPRoutingActivation container.
--	--	--

Parameters included

Parameter name	Multiplicity
DoIPRoutingActivationNumber	1..1
DoIPRoutingActivationSecurityRequired	1..1
DoIPTargetAddressRef	0..65535

Parameter Name	DoIPRoutingActivationNumber	
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	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DoIPRoutingActivationSecurityRequired	
Description	Indicates if a routing activation requires a secure TCP connection.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DoIPTargetAddressRef	
Description	Reference to all DoIPTargetAddress which are activated on this routing activation.	



Multiplicity	0..65535
Type	REFERENCE
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.2.1.20. DolPRoutingActivationAuthenticationCallback

Parameters included	
Parameter name	Multiplicity
DolPRoutingActivationAuthenticationFunc	0..1
DolPRoutingActivationAuthenticationReqLength	1..1
DolPRoutingActivationAuthenticationResLength	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 DolIP 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 AuthenticationReqData 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 AuthenticationResData will not be fetched via the API.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=4 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.2.1.21. DolPRoutingActivationConfirmationCallback

Parameters included	
Parameter name	Multiplicity
DolPRoutingActivationConfirmationFunc	0..1
DolPRoutingActivationConfirmationReqLength	1..1
DolPRoutingActivationConfirmationResLength	1..1

Parameter Name	DolPRoutingActivationConfirmationFunc
Description	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.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

Parameter Name	DolPRoutingActivationConfirmationReqLength
Description	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.



Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<=4 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPRoutingActivationConfirmationResLength
Description	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.
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<=4 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.2.1.22. DolPTester

Parameters included	
Parameter name	Multiplicity
DolPNumByteDiagAckNack	1..1
DolPTesterSA	1..1
DolPRoutingActivationRef	0..255

Parameter Name	DolPNumByteDiagAckNack
Description	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.
Multiplicity	1..1



Type	INTEGER
Range	<=4294967295 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPTesterSA
Description	Source Address of the Tester sent via routing activation or diagnostic message.
Multiplicity	1..1
Type	INTEGER
Range	<=65535 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPRoutingActivationRef
Description	Reference to a DolPRoutingActivation describing the possible routing activations of the DolPTester.
Multiplicity	0..255
Type	REFERENCE
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.2.1.23. DolPGeneral

Containers included		
Container name	Multiplicity	Description
DolPGetGid	0..1	This container describes the Callback function to get the GID.
DolPPowerModeCallback	1..1	This container describes the Callback function to get the Power Mode. This container shall always be present.
DolPTriggerGidSynchronization	0..1	This container describes the Callback function to trigger the GID synchronisation.

**Containers included**

Note: Feature not supported.		
DoIPProtocolVersion	1..1	This container defines the supported protocol versions of the DoIP entity.
DoIPDynamicGIDMasterSelectionCallback	0..1	<p>This container describes the Callback function to obtain DoIP entity GID master/slave status.</p> <p>If this container is not configured no Callback function will be used and Dynamic GID Synchronization feature will be disabled.</p>
DoIPVIDRequestReceived-Callback	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>
DoIPGetVIN	0..1	<p>This container describes the Callback function to get the VIN.</p> <p>If this function is not configured DoIP module uses the configured DoIPVinInvalidityPattern.</p>
DoPTcpSoConModeChgCallback	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>
DoPSecurityEventRefs	1..1	Container for the references to IdsMEvent elements representing the security events that the DoIP module shall report to the IdsM in case the corresponding security related event occurs (and if DoPEnableSecurityEventReporting is set to "true").

Parameters included

Parameter name	Multiplicity
DoIPAliveCheckResponseTimeout	1..1
DoIPDevelopmentErrorDetect	1..1
DoIPDhcpOptionVinUse	1..1
DoIPEntityStatusMaxByteFieldUse	1..1
DoIPGIDInvalidityPattern	1..1
DoIPGeneralInactivityTime	1..1
DoIPHeaderFileInclusion	0..n
DoIPHostNameSizeMax	1..1
DoIPInitialInactivityTime	1..1



Parameters included

DoIPInitialVehicleAnnouncementTime	1..1
DoIPMainFunctionPeriod	1..1
DoIPMaxRequestBytes	1..1
DoIPMaxTesterConnections	1..1
DoIPMaxUDPSocketConnections	1..1
DoIPMaxUDPRequestPerMessage	1..1
DoIPNodeType	1..1
DoIPUseEIDasGID	1..1
DoIPUseMacAddressForIdentification	1..1
DoIPUseVehicleIdentificationSyncStatus	1..1
DoIPVehicleAnnouncementInterval	1..1
DoIPVehicleAnnouncementRepetition	1..1
DoIPVersionInfoApi	1..1
DoIPVInGidMaster	1..1
DoIPVInInvalidityPattern	1..1
DoIPEnableMainFunctionTx	1..1
DoIPMaxChannels	1..1
DoIPMaxNumByteDiagAckNack	1..1
DoIPCustomPayloadTypeEnabled	1..1
DoIPGetAndResetMeasurementDataApi	1..1
DoIPConnectionControl	1..1
DoIPDhcpHostNamePrefix	1..1
DoIPEnableTcpClosureWithFIN	1..1
DoIPLocalIPAddressReleaseDelay	1..1
DoIPResponseBeforeRoutingActivation	1..1
DoIPMaxVehicleAnnouncementCon	1..1
DoIPRteUsage	1..1
DoIPRelocatablePbcfgEnable	1..1
DoIPEnableSecurityEventReporting	1..1
DoIPRoutingActivationCallbackList	0..255

Parameter Name	DoIPAliveCheckResponseTimeout
-----------------------	--------------------------------------



Description	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.
Multiplicity	1..1
Type	FLOAT
Default value	0.5
Range	<=Infinity >=0.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPDevelopmentErrorDetect
Description	Pre-processor switch for enabling development error detection support.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPDhcpOptionVinUse
Description	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.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPEntityStatusMaxByteFieldUse
Description	This parameter is used to distinguish the optional support of the Max data size element of a diagnostic entity status response.
Multiplicity	1..1
Type	BOOLEAN
Default value	false



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPGIDInvalidityPattern	
Description	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.	
Multiplicity	1..1	
Type	INTEGER	
Range	0 255	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPGeneralInactivityTime	
Description	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.	
Multiplicity	1..1	
Type	FLOAT	
Default value	1.0	
Range	<=Infinity >=0.0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPHeaderFileInclusion	
Description	Name of the header file(s) to be included by the DolP module containing the used C-callback declarations.	
Multiplicity	0..n	
Type	STRING	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPHostNameSizeMax	



Description	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 "DoIP-" on any configuration.
Multiplicity	1..1
Type	INTEGER
Default value	5
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPInitialInactivityTime
Description	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.
Multiplicity	1..1
Type	FLOAT
Default value	1.0
Range	<=Infinity >=0.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPInitialVehicleAnnouncementTime
Description	Time to wait in [s] for sending first vehicle announcement message after IP address assignment. Represents parameter A_DoIP_Announce_Wait of ISO 13400-2:2012.
Multiplicity	1..1
Type	FLOAT
Default value	0.005
Range	<=Infinity >=0.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPMainFunctionPeriod
-----------------------	-------------------------------



Description	Determines the frequency at which the DolP_MainFunction() is called in [s].	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.001	
Range	<=Infinity >=0.0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPMaxRequestBytes	
Description	Specifies the maximum allowed bytes of a DolP message request without the DolP header.	
Multiplicity	1..1	
Type	INTEGER	
Default value	10	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPMaxTesterConnections	
Description	Maximum amount of tester connections that shall be maintained at one time before alive check is performed.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=255 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPMaxUDPSocketConnections	
Description	Maximum number of UDP socket connections.	
Multiplicity	1..1	
Type	INTEGER	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DolPMaxUDPRRequestPerMessage	
Description	This parameter captures the maximum amount of UDP Requests necessary to handle parallel within a single UDP connection. Note: This configuration parameter is not used.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=255 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPNodeType	
Description	Describes the Type of the DolP node.	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	DOIP_NODE	
Range	DOIP_GATEWAY DOIP_NODE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPUseEIDasGID	
Description	Specifies if the DolP entity shall use its EID if it is the Master for vehicle identification gid on the vehicle identification/vehicle announcement.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	



Parameter Name	DolPUseMacAddressForIdentification	
Description	Provides the information if a configured EID at vehicle identification response/vehicle announcement is used or the MAC address.	
Multiplicity	1..1	
Type	BOOLEAN	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPUseVehicleIdentificationSyncStatus	
Description	Defines if the optional VIN/GID synchronization status is used additionally in the vehicle identification/announcement.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPVehicleAnnouncementInterval	
Description	Time to wait in [s] for sending subsequent vehicle announcement messages. Represents parameter A_DoIP_Announce_Interval of ISO 13400-2:2012.	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.005	
Range	<=Infinity >=0.0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	DolPVehicleAnnouncementRepetition	
Description	Amount of repetitions of the vehicle announcement message on IP address assignment. Represents parameter A_DoIP_Announce_Num of ISO 13400-2:2012.	
Multiplicity	1..1	
Type	INTEGER	
Default value	3	



Range	<=255
	>=1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPVersionInfoApi
Description	Activates the DolP_GetVersionInfo() API.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	DolPVinGidMaster
Description	Specifies if the DolP entity is the Vehicle Identification Master for the GID (Group ID). Note: Feature not supported.
Multiplicity	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 DolPChannel and DolPCustomChannel if enabled).	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65535 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DolPMaxNumByteDiagAckNack	
Description	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. This results in reduced RAM and ROM consumption.	
Multiplicity	1..1	
Type	INTEGER	
Default value	128	
Range	<=65535 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DolPCustomPayloadTypeEnabled	
-----------------------	-------------------------------------	--



Description	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.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	DoIPGetAndResetMeasurementDataApi
Description	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.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	DoIPConnectionControl
Description	This parameter enables a set of APIs to allow an upper layer to disconnect testers by closing its TCP connection. List of available APIs: ▶ DoIP_DisconnectTester() ▶ DoIP_GetSoConIdFromRxPduId() ▶ DoIP_GetSoConIdFromTxPduId()
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH



Parameter Name	DolPDhcpHostNamePrefix	
Description	If parameter DolPDhcpHostNamePrefix is not an empty string it will be added to the DHCP host name after DoIP- 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.	
Multiplicity	1..1	
Type	STRING	
Default value		
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DolPEnableTcpClosureWithFIN	
Description	Enables Tcp connection closure with FIN flag set when performed together with NACK transmission. In this case <code>SoAd_CloseSoCon()</code> will be called with Abort parameter set to <code>FALSE</code> . In other cases when DoIP needs to close Tcp connection, e.g. due to Inactivity or Alive check timeout, Abort parameter will be set to <code>TRUE</code> . This will perform a closure with RST flag set. When this parameter is disabled, <code>SoAd_CloseSoCon()</code> will always be called with Abort set to <code>TRUE</code> .	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DolPLocalIPAddressReleaseDelay	
Description	Time to wait in [s] after <code>DoIP_ActivationLineSwitchInactive()</code> is called before the local IP address assignment is released with <code>SoAd_ReleaseIpAddrAssignment</code> . For TLS connection a proper delay is required to allow TLS to send an encrypted CloseNotify before local address gets released. If TLS is not used this parameter can be set to 0. Parameter is enabled if DolPEnableTcpClosureWithFIN is set to TRUE.	
Multiplicity	1..1	
Type	FLOAT	



Default value	0.1
Range	<=Infinity >=0.0
Configuration class	PreCompile: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	DolPResponseBeforeRoutingActivation
Description	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.
Multiplicity	1..1
Type	ENUMERATION
Default value	DOIP_ENABLE_ALL_DIAG_NACK
Range	DOIP_ENABLE_ALL_DIAG_NACK DOIP_DISABLE_DIAG_NACK_0x06
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	DolPMaxVehicleAnnouncementCon
Description	Specifies maximum number of vehicle announcement connections that can be configured.
Multiplicity	1..1
Type	INTEGER
Default value	1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	DolPRteUsage
Description	Enables RTE Usage. If enabled, the DolP will generate an SWCD and supply the specified software component interfaces.
Multiplicity	1..1



Type	BOOLEAN
Default value	false
Configuration class	PreCompile:
Origin	Elektrobit Automotive GmbH

Parameter Name	DolPRelocatablePbcfgEnable	
Description	Enables/disable support for relocatable postbuild configuration. ▶ True: Postbuild configuration relocatable in memory. ▶ False: Postbuild configuration not relocatable in memory.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DolPEnableSecurityEventReporting	
Description	Enables (TRUE) or disables (FALSE) the reporting of security events to the IdsM.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DolPRoutingActivationCallbackList	
Description	<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, con- st 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>	



	<p>SoConId - SoConId of a Tcp connection that routing activation message was received on.</p> <p>SourceAddr - Source address field from routing activation message.</p> <p>ActivationType - Activation type field from routing activation message.</p> <p>OemSpecificPtr - Pointer to OEM specific field from routing activation message (4 bytes). NULL_PTR if OEM specific field was omitted.</p> <p>Note: User defined header files shall be added to the configuration container DolPHeaderFileInclusion.</p>
Multiplicity	0..255
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.2.1.24. DolPGetGid

Parameters included	
Parameter name	Multiplicity
DolPGetGID	0..1
Parameter Name	DolPGetGID
Description	Direct C Callback function to get the GID of the DoIP entity.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.2.1.25. DolPPowerModeCallback

Parameters included	
Parameter name	Multiplicity
DolPPowerMode	0..1
Parameter Name	DolPPowerMode



Description	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.	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.2.1.26. DoIPTriggerGidSynchronization

Parameters included		
Parameter name	Multiplicity	
DoIPTriggerGidSynchronization	0..1	
Parameter Name		
Description	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.	
	Note: Feature not supported.	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.2.1.27. DoIPProtocolVersion

Parameters included		
Parameter name	Multiplicity	
DoIP_ISO13400_2_2012	1..1	
DoIP_ISO13400_2_2019	1..1	
DoIPSecuredCommunicationForAllProtocolVersions	1..1	
Parameter Name		
DoIP_ISO13400_2_2012		



Description	If this parameter is enabled, the DoIP entity supports communication with ISO13400-2:2012 complaint tester using protocol version 0x02.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DolP_ISO13400_2_2019	
Description	If this parameter is enabled, the DoIP entity supports communication with ISO13400-2:2019 complaint tester using protocol version 0x03.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DolPSecuredCommunicationForAllProtocolVersions	
Description	If this parameter is enabled, the DoIP entity supports secured communication for all supported protocol version (including 0x02). ISO13400-2:2012 complaint tester are able to use TLS secured connections and will get the routing activation response code 0x07 as defined by ISO13400-2:2019. This parameter is for backward compatibility.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.2.1.28. DolPDynamicGIDMasterSelectionCallback

Parameters included	
Parameter name	Multiplicity



Parameters included

DoIPDynamicGIDMasterSelection	0..1
---	------

Parameter Name	DoIPDynamicGIDMasterSelection	
Description	Direct C Callback function to get the DoIP entity GID master/slave status.	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	Link:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.2.1.29. DoPVIDRequestReceivedCallback

Parameters included

Parameter name	Multiplicity
DoPVIDRequestReceived	1..1

Parameter Name	DoPVIDRequestReceived	
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. Note: User defined header files can be added to configuration container DoIP-HeaderFileInclusion.	
Multiplicity	1..1	
Type	FUNCTION-NAME	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.2.1.30. DoPGetVIN

Parameters included

Parameter name	Multiplicity
DoPGetVin	0..1



Parameter Name	DolPGetVin	
Description	The function should store VIN into specified location.	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	Link:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.2.1.31. DolPTcpSoConModeChgCallback

Parameters included	
Parameter name	Multiplicity
DolPTcpSoConModeChg	1..1

Parameter Name	DolPTcpSoConModeChg	
Description	<p>User defined callback function to be called to provide information about the TCP connection status.</p> <p>Syntax: (void) UserDefinedFunction(SoAd_SoConIdType SoConId, SoAd_SoConModeType Status)</p> <p>with UserDefinedFunction as placeholder for a user defined function name provided with this configuration field.</p> <p>Note: User defined header files can be added to configuration container DolP-HeaderFileInclusion.</p>	
Multiplicity	1..1	
Type	FUNCTION-NAME	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.2.1.32. DolPSecurityEventRefs

Parameters included	
Parameter name	Multiplicity
DOIP_SEV_DROP_UDP_INV_MSG	0..1



Parameters included

DOIP_SEV_VEHICLE_ID_RESP_TX	0..1
DOIP_SEV_ENTITY_STATUS_RESP_TX	0..1
DOIP_SEV_POWER_MODE_RESP_TX	0..1
DOIP_SEV_DROP_TCP_GEN_HDR_ERR	0..1
DOIP_SEV_ROUTING_ACT_RESP_TX	0..1
DOIP_SEV_ALIVE_RESP_WRONG_SA_RESET	0..1
DOIP_SEV_ALIVE_RESP	0..1
DOIP_SEV_DROP_DIAG_MSG_DIAG_HDR	0..1
DOIP_SEV_DROP_DIAG_MSG_UL	0..1
DOIP_SEC_TCP_CONN_RESET_INACTIVITY_TIMER	0..1
DOIP_SEC_TCP_CONN_RESET_ALIVE_CHECK_TIMER	0..1

Parameter Name

DOIP_SEV_DROP_UDP_INV_MSG

Description	Reference to the IdsMEvent DOIP_SEV_DROP_UDP_INV_MSG: DoIP message received on UDP connection discarded.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name

DOIP_SEV_VEHICLE_ID_RESP_TX

Description	Reference to the IdsMEvent DOIP_SEV_VEHICLE_ID_RESP_TX: Vehicle identification response message transmitted.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name

DOIP_SEV_ENTITY_STATUS_RESP_TX

Description	Reference to the IdsMEvent DOIP_SEV_ENTITY_STATUS_RESP_TX: DoIP entity status response message transmitted.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DOIP_SEV_POWER_MODE_RESP_TX	
Description	Reference to the IdsMEvent DOIP_SEV_POWER_MODE_RESP_TX: DoIP power mode response message transmitted.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DOIP_SEV_DROP_TCP_GEN_HDR_ERR	
Description	Reference to the IdsMEvent DOIP_SEV_DROP_TCP_GEN_HDR_ERR: DoIP message received on a TCP connection discarded due to generic header error.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DOIP_SEV_ROUTING_ACT_RESP_TX	
Description	Reference to the IdsMEvent DOIP_SEV_ROUTING_ACT_RESP_TX: Routing activation positive acknowledge transmitted.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DOIP_SEV_ALIVE_RESP_WRONG_SA_RESET	
Description	Reference to the IdsMEvent DOIP_SEV_ALIVE_RESP_WRONG_SA_RESET: Tcp connection got reset because Alive check response message is received on a registered Tcp connection with a wrong source address (received source address does not match the registered one).	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	Elektrobit Automotive GmbH
---------------	----------------------------

Parameter Name	DOIP_SEV_ALIVE_RESP
Description	Reference to the IdsMEvent DOIP_SEV_ALIVE_RESP: Alive check response message received on a registered Tcp connection with a registered source address, or received on an unregistered Tcp connection (ignored).
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	DOIP_SEV_DROP_DIAG_MSG_DIAG_HDR
Description	Reference to the IdsMEvent DOIP_SEV_DROP_DIAG_MSG_DIAG_HDR: Diagnostic message discarded due to the following reasons: <ul style="list-style-type: none">▶ Invalid Source address (NACK code 0x02)▶ Invalid Target address (NACK code 0x03)▶ Message too long (NACK code 0x04)▶ Inactive route (NACK code 0x06)▶ Silent discard
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	DOIP_SEV_DROP_DIAG_MSG_UL
Description	Reference to the IdsMEvent DOIP_SEV_DROP_DIAG_MSG_UL: Diagnostic message discarded because upper layer was not able to receive it (diagnostic negative acknowledge message generated with NACK code 0x05 or 0x08).
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	DOIP_SEC_TCP_CONN_RESET_INACTIVITY_TIMER
-----------------------	---



Description	Reference to the IdsMEvent DOIP_SEC_TCP_CONN_RESET_INACTIVITY_TIMER: Tcp connection got reset because Inactivity timer expired.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DOIP_SEC_TCP_CONN_RESET_ALIVE_CHECK_TIMER	
Description	Reference to the IdsMEvent DOIP_SEC_TCP_CONN_RESET_ALIVE_CHECK_TIMER: Tcp connection got reset because Alive check timer expired.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.2.1.33. DoIPDefensiveProgramming

Parameters included		
Parameter name		Multiplicity
DoIPDefProgEnabled		1..1
DoIPPrecondAssertEnabled		1..1
DoIPPostcondAssertEnabled		1..1
DoIPStaticAssertEnabled		1..1
DoIPUnreachAssertEnabled		1..1
DoPIInvariantAssertEnabled		1..1

Parameter Name	DoIPDefProgEnabled
Label	Enable Defensive Programming
Description	<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"> 1. Enable development error detection 2. Enable defensive programming



	3. Enable assertions as required
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	DoIPPrecondAssertEnabled
Label	Enable Precondition Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>DoIPDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>DoIPDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	DoIPPostcondAssertEnabled
Label	Enable Postcondition Assertions
Description	<p>Enables handling of postcondition assertion checks reported from the module DoIP.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Enable Development Error Detection (<code>DoIPDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>DoIPDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DolPStaticAssertEnabled	
Label	Enable Static Assertions	
Description	<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"> ▶ Enable Development Error Detection (<code>DoIPDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>DoIPDefProgEnabled</code>): must be enabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DolPUreachAssertEnabled	
Label	Enable Unreachable Code Assertions	
Description	<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"> ▶ Enable Development Error Detection (<code>DoIPDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>DoIPDefProgEnabled</code>): must be enabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	DolPIvariantAssertEnabled	
-----------------------	----------------------------------	--



Label	Enable Invariant Assertions
Description	<p>Enables handling of invariant assertion checks reported from functions of the module DoIP.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Enable Development Error Detection (<code>DoIPDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>DoIPDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PbcfgMSupport: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.2.1.34. PublishedInformation

Parameters included	
Parameter name	Multiplicity
PbcfgMSupport	1..1
Parameter Name	
Parameter Name	PbcfgMSupport
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	PublishedInformation:
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	
Value	0x03U
Description	DET error code: Invalid ID

5.2.2.1.3. DOIP_E_PARAM_POINTER

Purpose	
Value	0x02U
Description	DET error code: Invalid pointer (NULL_PTR)

5.2.2.1.4. DOIP_E_UNINIT

Purpose	
Value	0x01U
Description	DET error code: Module not initialized

5.2.2.1.5. DOIP_INSTANCE_ID

Purpose	Module instance ID.
Value	0U
Description	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

Purpose	
Value	0x0FU
Description	SID for DoIP_ActivationLineSwitch()

5.2.2.1.7. DOIP_SID_CANCELRECEIVE

Purpose	
Value	0x05U
Description	SID for DoIP_CancelReceive()

5.2.2.1.8. DOIP_SID_CANCELTRANSMIT

Purpose	
Value	0x04U
Description	SID for DoIP_CancelTransmit()

5.2.2.1.9. DOIP_SID_DISCONNECTTESTER

Purpose	
Value	0xF2U
Description	SID for DoIP_DisconnectTester()

5.2.2.1.10. DOIP_SID_GETANDRESETMEASUREMENTDATA

Purpose	Defines API id of function DoIP_GetAndResetMeasurementData() .
Value	0x50U

5.2.2.1.11. DOIP_SID_GETSOCONIDFROMRXPDUID

Purpose	
Value	0xF3U
Description	SID for DoIP_GetSoConIdFromRxPduId()



5.2.2.1.12. DOIP_SID_GETSOCONIDFROMTXPDUID

Purpose	
Value	0xF4U
Description	SID for DoIP_GetSoConIdFromTxPduld()

5.2.2.1.13. DOIP_SID_GETVERSIONINFO

Purpose	
Value	0x00U
Description	SID for DoIP_GetVersionInfo()

5.2.2.1.14. DOIP_SID_IFTRANSMIT

Purpose	
Value	0x49U
Description	SID for DoIP_IfTransmit()

5.2.2.1.15. DOIP_SID_INIT

Purpose	
Value	0x01U
Description	SID for DoIP_Init()

5.2.2.1.16. DOIP_SID_LOCALIPADDRASSIGNMENTCHG

Purpose	
Value	0x0CU
Description	SID for DoIP_LocalIpAddrAssignmentChg()

5.2.2.1.17. DOIP_SID_MAINFUNCTION

Purpose	
Value	0x02U
Description	SID for DoIP_MainFunction()



5.2.2.1.18. DOIP_SID_MAINFUNCTION_TX

Purpose	
Value	0xF1U
Description	SID for DoIP_MainFunctionTx()

5.2.2.1.19. DOIP_SID_SOADIFRXINDICATION

Purpose	
Value	0x42U
Description	SID for DoIP_SoAdIfRxIndication()

5.2.2.1.20. DOIP_SID_SOADIFTXCONFIRMATION

Purpose	
Value	0x40U
Description	SID for DoIP_SoAdIfTxConfirmation()

5.2.2.1.21. DOIP_SID_SOADTPCOPYRXDATA

Purpose	
Value	0x08U
Description	SID for DoIP_SoAdTpCopyRxData()

5.2.2.1.22. DOIP_SID_SOADTPCOPYTXDATA

Purpose	
Value	0x06U
Description	SID for DoIP_SoAdTpCopyTxData()

5.2.2.1.23. DOIP_SID_SOADTPRXINDICATION

Purpose	
----------------	--



Value	0x0AU
Description	SID for DoIP_SoAdTpRxIndication()

5.2.2.1.24. DOIP_SID_SOADTPSTARTOFRECEPTION

Purpose	
Value	0x09U
Description	SID for DoIP_SoAdTpStartOfReception()

5.2.2.1.25. DOIP_SID_SOADTPTXCONFIRMATION

Purpose	
Value	0x07U
Description	SID for DoIP_SoAdTpTxConfirmation()

5.2.2.1.26. DOIP_SID_SOCONMODECHG

Purpose	
Value	0x0BU
Description	SID for DoIP_SoConModeChg()

5.2.2.1.27. DOIP_SID_TPTRANSMIT

Purpose	
Value	0x49U
Description	SID for DoIP_TpTransmit()

5.2.2.2. Functions

5.2.2.2.1. DoIP_ActivationLineSwitchActive

Purpose	API to enable connections.
----------------	----------------------------



Synopsis	<code>void DoIP_ActivationLineSwitchActive (void);</code>
Service ID	0x0F
Sync/Async	Synchronous
Reentrancy	Non reentrant
Description	This function opens all configured socket connections.

5.2.2.2. DoIP_ActivationLineSwitchInactive

Purpose	API to disable connections.
Synopsis	<code>void DoIP_ActivationLineSwitchInactive (void);</code>
Service ID	0x0F
Sync/Async	Synchronous
Reentrancy	Non reentrant
Description	This function closes all configured socket connections.

5.2.2.2.3. DoIP_CancelReceive

Purpose	API to cancel a reception from SoAd to PduR.	
Synopsis	<code>Std_ReturnType DoIP_CancelReceive (PduIdType DoIPPduRRxId);</code>	
Service ID	0x05	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Pdulds. Non reentrant for the same Pduld	
Parameters (in)	DoIPPduRRxId	- DoIP handle ID to be used for DoIP API calls from PduR.
Return Value	Std_ReturnType E_OK E_NOT_OK	
	E_OK	- Request is accepted.
	E_NOT_OK	- Request is rejected.

5.2.2.2.4. DoIP_CancelTransmit

Purpose	API to cancel a transmission from PduR to SoAd.
----------------	---



Synopsis	<code>Std_ReturnType DoIP_CancelTransmit (PduIdType DoIPPduRTxId);</code>	
Service ID	0x04	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Pdulds. Non reentrant for the same Pduld	
Parameters (in)	<code>DoIPPduRTxId</code>	- DoIP handle ID to be used for DoIP API calls from PduR.
Return Value	Std_ReturnType E_OK - Request is accepted. E_NOT_OK - Request is rejected.	

5.2.2.2.5. DoIP_DisconnectTester

Purpose	API to close specific DoIP Tcp connection.	
Synopsis	<code>Std_ReturnType DoIP_DisconnectTester (SoAd_SoConIdType SoConId, boolean Abort);</code>	
Parameters (in)	SoConId Abort	- Socket connection index specifying which socket connection shall be closed. - TRUE: Socket connection shall be closed with RST. ▶ FALSE: Socket connection shall be closed with FIN.
Return Value	Std_ReturnType E_OK - Connection with specified SoConId found and scheduled for closing. E_NOT_OK - Connection with specified SoConId not found.	
Description	SoConId is provided via routing activation callback functions when configured, or it can be retrieved with DoIP_GetSoConIdFromRxPduld() and DoIP_GetSoConIdFromTxPduld() APIs. Actual connection closure is performed in the next main function call.	

5.2.2.2.6. DoIP_GetAndResetMeasurementData

Purpose	API to read and reset measurement data.
----------------	---



Synopsis	<code>Std_ReturnType DoIP_GetAndResetMeasurementData (DoIP_MeasurementIdxType MeasurementIdx , boolean MeasurementResetNeeded , uint32 * MeasurementDataPtr);</code>					
Parameters (in)	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.				
Parameters (out)	MeasurementDataPtr	Pointer to data buffer, where to copy measurement data.				
Return Value	<p><code>Std_ReturnType</code></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><code>E_OK</code></td> <td style="padding: 2px;">The function has been successfully executed.</td> </tr> <tr> <td style="padding: 2px;"><code>E_NOT_OK</code></td> <td style="padding: 2px;">The function could not be successfully executed.</td> </tr> </table>		<code>E_OK</code>	The function has been successfully executed.	<code>E_NOT_OK</code>	The function could not be successfully executed.
<code>E_OK</code>	The function has been successfully executed.					
<code>E_NOT_OK</code>	The function could not be successfully executed.					
Description	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.					

5.2.2.2.7. DoIP_GetSoConIdFromRxPduId

Purpose	API to retrieve Socket connection index of a Tcp connection.					
Synopsis	<code>Std_ReturnType DoIP_GetSoConIdFromRxPduId (PduIdType RxPduId , SoAd_SoConIdType * SoConIdPtr);</code>					
Parameters (in)	RxPduId	- Identification of the PDU for which SoConId shall be retrieved.				
Parameters (out)	SoConIdPtr	- Socket connection index of a Tcp connection to which PDU is assigned.				
Return Value	<p><code>Std_ReturnType</code></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><code>E_OK</code></td> <td style="padding: 2px;">- SoConId successfully retrieved.</td> </tr> <tr> <td style="padding: 2px;"><code>E_NOT_OK</code></td> <td style="padding: 2px;">- Retrieving SoConId failed.</td> </tr> </table>		<code>E_OK</code>	- SoConId successfully retrieved.	<code>E_NOT_OK</code>	- Retrieving SoConId failed.
<code>E_OK</code>	- SoConId successfully retrieved.					
<code>E_NOT_OK</code>	- Retrieving SoConId failed.					



Description	An API to retrieve Socket connection index of a Tcp connection to which specified Rx-Pduld is assigned.
--------------------	---

5.2.2.2.8. DoIP_GetSoConIdFromTxPduld

Purpose	API to retrieve Socket connection index of a Tcp connection.	
Synopsis	<pre>Std_ReturnType DoIP_GetSoConIdFromTxPduId (PduIdType TxPduId , SoAd_SoConIdType * SoConIdPtr);</pre>	
Parameters (in)	TxPduId	- Identification of the PDU for which SoConId shall be retrieved.
Parameters (out)	SoConIdPtr	- Socket connection index of a Tcp connection to which PDU is assigned.
Return Value	Std_ReturnType	
	E_OK	- SoConId successfully retrieved.
	E_NOT_OK	- Retrieving SoConId failed.
Description	An API to retrieve Socket connection index of a Tcp connection to which specified TxPduld is assigned.	

5.2.2.2.9. DoIP_GetVersionInfo

Purpose	API to get the module version information.
Synopsis	<code>void DoIP_GetVersionInfo (Std_VersionInfoType * VersionInfo);</code>
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (out)	VersionInfo - Pointer to return the module version information.
Description	This service returns the version information of this module.

5.2.2.2.10. DoIP_Init

Purpose	API to initialize the module.
----------------	-------------------------------



Synopsis	<code>void DoIP_Init (const DoIP_ConfigType * DoIP_ConfigPtr);</code>	
Service ID	0x01	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	DoIP_ConfigPtr	- Pointer to the module post build configuration.
Description	This service initializes the modules shared variables.	

5.2.2.11. DoIP_IsValidConfig

Purpose	Checks compatibility of the post-build-time configuration.	
Synopsis	<code>Std_ReturnType DoIP_IsValidConfig (const void * voidConfigPtr);</code>	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	voidConfigPtr	Pointer to the configuration data of the DoIP module.
Return Value	E_OK	Provided configuration is compatible.
E_NOT_OK	Provided configuration is not compatible.	
Description	This service checks the compatibility of the post-build-time configuration	

5.2.2.12. DoIP_LocallpAddrAssignmentChg

Purpose	API to indicate local IP address assignment changes.	
Synopsis	<code>void DoIP_LocallpAddrAssignmentChg (SoAd_SoConIdType SoConId , TcpIp_IpAddrStateType State);</code>	
Service ID	0x0C	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SoConIds. Non reentrant for the same SoConId	
Parameters (in)	SoConId	- Socket connection index.
	State	- Returns information if IP address is assigned or not.



5.2.2.2.13. DoIP_MainFunction

Purpose	Main function API.
Synopsis	<code>void DoIP_MainFunction (void);</code>
Service ID	0x02
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Description	This function executes module tasks periodically. fixed

5.2.2.2.14. DoIP_MainFunctionTx

Purpose	Transmit part of the main function if enabled to be external.
Synopsis	<code>void DoIP_MainFunctionTx (void);</code>
Service ID	0xF1

5.2.2.2.15. DoIP_SoAdIfRxIndication

Purpose	API to indicate a UDP reception.	
Synopsis	<code>void DoIP_SoAdIfRxIndication (PduIdType RxPduId , PduInfoType * PduInfoPtr);</code>	
Service ID	0x42	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Pdulds. Non reentrant for the same Pduld	
Parameters (in)	RxPduId	- DoIP handle ID to be used for DoIP APIs to be called from SoAd.
	PduInfoPtr	- Pointer to received data.

5.2.2.2.16. DoIP_SoAdIfTxConfirmation

Purpose	API to confirm a UDP transmission.	
Synopsis	<code>void DoIP_SoAdIfTxConfirmation (PduIdType TxPduId);</code>	
Service ID	0x40	
Sync/Async	Synchronous	



Reentrancy	Non reentrant	
Parameters (in)	TxPduId	- DoIP handle ID to be used for DoIP APIs to be called from SoAd.

5.2.2.2.17. DoIP_SoAdTpCopyRxData

Purpose	API to provide received data.	
Synopsis	<pre>BufReq_ReturnType DoIP_SoAdTpCopyRxData (PduIdType RxPduId , const PduInfoType * PduInfoPtr , PduLengthType * BufferSizePtr);</pre>	
Service ID	0x08	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Pduds. Non reentrant for the same Pdud	
Parameters (in)	RxPduId	- DoIP handle ID to be used for DoIP APIs to be called from SoAd.
	PduInfoPtr	- Pointer providing received data and data length.
Parameters (out)	BufferSizePtr	- Amount of data which shall be provided next call.
Return Value	BufReq_ReturnType	
	BUFREQ_OK	- Data is copied.
	BUFREQ_E_BUSY	- Request postponed. No data is copied.
	BUFREQ_E_NOT_OK	- Request failed.

5.2.2.2.18. DoIP_SoAdTpCopyTxData

Purpose	API to request data to transmit.	
Synopsis	<pre>BufReq_ReturnType DoIP_SoAdTpCopyTxData (PduIdType TxPduId , PduInfoType * PduInfoPtr , RetryInfoType * Retry , Pdu- LengthType * AvailableDataPtr);</pre>	
Service ID	0x06	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	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.
Parameters (out)	AvailableDataPtr	- Pointer which returns remaining number of bytes to be copied.
Return Value	BufReq_ReturnType	
	BUFREQ_OK	- Data is copied.
	BUFREQ_E_BUSY	- Request postponed. No data is copied.
	BUFREQ_E_NOT_OK	- Request failed.

5.2.2.2.19. DoIP_SoAdTpRxIndication

Purpose	API to indicate that all TCP receptions from this tester are finished.	
Synopsis	<pre>void DoIP_SoAdTpRxIndication (PduIdType RxPduId , NotifResultType Result);</pre>	
Service ID	0x0A	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Pdulds. Non reentrant for the same Pduld	
Parameters (in)	RxPduId	- DoIP handle ID to be used for DoIP APIs to be called from SoAd.
	Result	- Result of the finished reception.

5.2.2.2.20. DoIP_SoAdTpStartOfReception

Purpose	API to start a reception.	
Synopsis	<pre>BufReq_ReturnType DoIP_SoAdTpStartOfReception (PduIdType RxPduId , PduLengthType TpSduLength , PduLengthType * BufferSizePtr);</pre>	
Service ID	0x09	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Pdulds. Non reentrant for the same Pduld	
Parameters (in)	RxPduId	- DoIP handle ID to be used for DoIP APIs to be called from SoAd.



	TpSduLength	- Message length shall always be zero for DoIP.
Parameters (out)	BufferSizePtr	Available Rx buffer in the DoIP module.
Return Value	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.
Description	This function is called once by SoAd if a TCP connection to a tester is established.	

5.2.2.2.21. DoIP_SoAdTpTxConfirmation

Purpose	API to confirm a TCP transmission.	
Synopsis	<pre>void DoIP_SoAdTpTxConfirmation (PduIdType TxPduId , NotifResultType Result);</pre>	
Service ID	0x07	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	TxPduId	- DoIP handle ID to be used for DoIP APIs to be called from SoAd.
	Result	- Parameter indicates the result of the transmission.

5.2.2.2.22. DoIP_SoConModeChg

Purpose	API to indicate SoAd socket connection state change.	
Synopsis	<pre>void DoIP_SoConModeChg (SoAd_SoConIdType SoConId , SoAd_SoConModeType Mode);</pre>	
Service ID	0x0B	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SoConIds. Non reentrant for the same SoConId	
Parameters (in)	SoConId	- Socket connection index.



	Mode	- New connection mode.
--	------	------------------------

5.2.2.23. DoIP_TpTransmit

Purpose	API to request Tp data transfer from the PduR to the SoAd.	
Synopsis	<code>Std_ReturnType DoIP_TpTransmit (PduIdType DoIPPduRTxId , const PduInfoType * DoIPPduRTxInfoPtr);</code>	
Service ID	0x03	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different Pdulds. Non reentrant for the same Pduld	
Parameters (in)	DoIPPduRTxId	- DoIP handle ID to be used for DoIP API calls from PduR.
	DoIPPduRTxInfoPtr	- Pointer to Tx PDU information structure which contains the length of the message to transmit.
Return Value	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

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.



5.2.3.2. 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

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

Description	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.
--------------------	---



Rationale	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

Description	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.
Rationale	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

Description	<p>The following parameters have to be manually configured:</p> <ul style="list-style-type: none"> ▶ DoIPRoutingActivation ▶ DoIP_Eid, DoIP_Gid, DoIP_LogicalAddress ▶ DoIP_Tester
Rationale	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

Description	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

Description	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(),
--------------------	---



	DolP_SoAdIfTxConfirmation(), DolP_SoAdTpStartOfReception(), DolP_SoAdTpCopyRxData(), DolP_SoAdTpRxIndication(), DolP_SoAdTpCopyTxData(), DolP_SoAdTpTxConfirmation(), DolP_SoConModeChg(), DolP_LocallIpAddrAssignmentChg() List of DolP main functions: DolP_MainFunction() DolP_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 DolP lower layer functions.
Rationale	This limitation reduces code size and execution time by eliminating the need for extensive use of exclusive areas.

5.2.3.4.6. DolP.EB_INTREQ_DolP_0006

Description	In case configuration parameter DolPTcpConnection/DolPTcpConnectionSecurityRequired is set to TRUE, corresponding TCP connection shall be secure (e.g. TLS).
--------------------	--

5.2.3.4.7. DolP.EB_INTREQ_DolP_0007

Description	If Diagnostic message is received before the route is activated, on a channel with DolPRoutingActivationType 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



Containers included

CommonPublishedInformation	1..1	Label: Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
EthIfConfigSet	1..1	Collection container for all parameters with post-build configuration classes
EthIfGeneral	1..1	General configuration parameters of the Ethernet Interface
EthIfDefensiveProgramming	1..1	Label: Defensive Programming Options Parameters for defensive programming
PublishedInformation	1..1	Label: EB Published Information Additional published parameters not covered by CommonPublishedInformation container.

Parameters included

Parameter name	Multiplicity
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT
Label	Config Variant
Multiplicity	1..1
Type	ENUMERATION
Default value	VariantPostBuild
Range	VariantPostBuild

5.3.1.1. CommonPublishedInformation

Parameters included

Parameter name	Multiplicity
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1
SwPatchVersion	1..1



Parameters included

ModuleId	1..1
VendorId	1..1
Release	1..1

Parameter Name	ArMajorVersion
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	ArMinorVersion
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	3
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ArPatchVersion
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	0
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH



Parameter Name	SwMajorVersion
Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMinorVersion
Label	Software Minor Version
Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	9
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwPatchVersion
Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	21
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ModuleId
Label	Numeric Module ID
Description	Module ID of this module from Module List
Multiplicity	1..1
Type	INTEGER_LABEL



Default value	65
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	VendorId
Label	Vendor ID
Description	Vendor ID of the dedicated implementation of this module according to the AU-TOSAR vendor list
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	Release
Label	Release Information
Multiplicity	1..1
Type	STRING_LABEL
Default value	
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

5.3.1.2. EthIfConfigSet

Containers included		
Container name	Multiplicity	Description
EthIfController	1..n	This container is a subcontainer of EthIfConfigSet and specifies the configuration parameters for an EthIf controller.
EthIfFrameOwnerConfig	1..n	Configuration of the Ethernet frame owner Note: Every Ethernet frame type that should be received needs to be listed in this container. A recommended configuration is available, which sets common Ethernet frame types together with their upper layer callback functions.
EthIfPhysController	1..255	Configuration of the EthIfPhysController



Containers included		
EthIfRxIndicationConfig	1..254	Configuration of receive callback functions
EthIfSwitch	0..n	This container contains the configuration of EthIfSwitches.
EthIfSwitchMgmtInfoIndicationConfig	0..n	Configuration of the switch management callback function Note: This configuration container is not used.
EthIfSwitchPortGroup	0..n	Configuration of EthIfSwitchPortGroups
EthIfSwitchTimeStampIndicationConfig	0..n	Configuration of switch timestamp indications Note: This configuration container is not used.
EthIfTransceiver	1..255	Configuration of the EthIfTransceiver The configuration flag EthIfTrcvSupportEnable enables this container.
EthIfTrcvLinkStateChgConfig	1..254	Specifies a link state change callback function If a link state change occurs, all listed upper layer callback functions are called.
EthIfTxConfirmationConfig	1..254	Configuration of transmit confirmation callback functions In case of a transmission confirmation, all listed upper layer callback functions are called.
EthIfEthControllerType	0..255	
EthIfEthTrcvType	0..255	
EthIfEthSwtType	0..1	

5.3.1.3. EthIfController

Parameters included	
Parameter name	Multiplicity
EthIfCtrlIdx	1..1
EthIfCtrlMtu	1..1
EthIfMaxTxBufsTotal	1..1
EthIfVlanId	0..1
EthIfEthTrcvRef	0..1
EthIfPhysControllerRef	1..1
EthIfSwitchRefOrPortGroupRef	0..1



Parameter Name	EthIfCtrlIdx	
Label	EthIf controller index	
Description	EthIf controller index	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfCtrlMtu	
Label	Maximum transmission unit	
Description	Specifies the maximum transmission unit (MTU) of the EthIfController in [bytes]. Note: If a VLAN tag is used for the EthIfController, the MTU is 4 bytes smaller than the maximum payload size of an Ethernet frame that can be transmitted on the network.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1500	
Range	<=9000 >=64	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfMaxTxBufsTotal	
Description	Limits the total number of transmit buffers. This configuration parameter is not used.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=4294967295 >=1	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfVlanId	
Label	VLAN identifier	
Description	Defines the 12-bit VLAN identifier that 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 4094 VLANs.	
Multiplicity	0..1	
Type	INTEGER	
Default value	1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfEthTrcvRef	
Label	Ethernet interface transceiver reference	
Description	Reference to the Ethernet transceiver on which this connection is transmitted/received. Connections are specified in the Socket Adaptor.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfPhysControllerRef	
Label	Ethernet interface physical controller reference	
Description	Reference to the physical controller reference in the Ethernet Driver on which this connection is transmitted/received. Connections are specified in the Socket Adaptor.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfSwitchRefOrPortGroupRef
-----------------------	-------------------------------------



Description	The choice reference allows to configure the EthIfController to reference either an EthIfSwitch or an EthIfSwitchPortGroup.	
Multiplicity	0..1	
Type	CHOICE-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.4. EthIfFrameOwnerConfig

Parameters included	
Parameter name	Multiplicity
EthIfFrameType	1..1
EthIfOwner	1..1

Parameter Name	EthIfFrameType
Description	Selects the Ethernet frame type Note: Typical values are 0x0800 for IPv4 or 0x0806 for ARP. A recommended configuration is available, which sets common Ethernet frame types. Note: 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.
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

Parameter Name	EthIfOwner
Description	Selects the owner of an Ethernet frame type. The owner is a zero-based index into the callback function configuration 'EthIfRxIndicationConfig'. This means an Ethernet frame of type IPv4 (0x800) at index 0 calls the first callback function configured in 'EthIfRxIndicationConfig'.
Multiplicity	1..1
Type	INTEGER



Default value	0
Range	<=254 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.3.1.5. EthIfPhysController

Containers included		
Container name	Multiplicity	Description
EthIfPhysCtrlRxMainFunctionPriorityProcessing	0..255	This container is a subcontainer of EthIfPhysController and specifies the configuration parameters for the priority processing.

Parameters included	
Parameter name	Multiplicity
EthIfPhysControllerIdx	1..1
EthIfEthCtrlRef	1..1
EthIfWEthCtrlRef	0..1

Parameter Name	EthIfPhysControllerIdx
Description	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.
Multiplicity	1..1
Type	INTEGER
Range	<=255 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfEthCtrlRef
Description	Reference to the controller in the Ethernet Driver on which this connection is transmitted/received. Connections are specified in the Socket Adaptor.
Multiplicity	1..1



Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	
Parameter Name	EthIfWEthCtrlRef	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.6. EthIfPhysCtrlRxMainFunctionPriorityProcessing

Parameters included		
Parameter name		Multiplicity
EthIfPhysCtrlRxIndicationIterations		1..1
EthIfPhysCtrlRxMainFunctionPeriod		1..1
EthIfPhysCtrlRxIngressFifoRef		1..1

Parameter Name	EthIfPhysCtrlRxIndicationIterations	
Description	Maximum number of Ethernet frames per Ethernet controller polled from the Ethernet driver within EthIf_MainFunctionRxPrio().	
Multiplicity	1..1	
Type	INTEGER	
Default value	10	
Range	<=65535 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfPhysCtrlRxMainFunctionPeriod	
Description	Specifies the period of the main function EthIf_MainFunctionRxPrio() in seconds. Note: Ethernet Interface does not require this information, but the BSW scheduler does.	
Multiplicity	1..1	



Type	FLOAT	
Default value	0.01	
Range	<=65535 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfPhysCtrlRxIngressFifoRef	
Description	Reference to the Ethernet controller ingress FIFO	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Range	node:when(count(node:ref(..../EthIfEthCtrlRef)/EthCtrlConfigIngress/EthCtrlConfigIngressFifo/*) > 0, node:paths(node:ref(..../EthIfEthCtrlRef)/EthCtrlConfigIngress/EthCtrlConfigIngressFifo/*), "")	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.7. EthIfRxIndicationConfig

Parameters included	
Parameter name	Multiplicity
EthIfRxIndicationFunction	1..1

Parameter Name	EthIfRxIndicationFunction
Description	Specifies the receive indication callback function Note: A typical receive indication function entry is Tcplp_RxIndication. A recommended configurations is available that adds common receive indication functions. Note: Enter Ethif_Up_RxIndicationDummy if no receive indication callback function shall be called.
Multiplicity	1..1
Type	FUNCTION-NAME
Configuration class	VariantPostBuild: VariantPostBuild



Origin	AUTOSAR_ECUC
---------------	--------------

5.3.1.8. EthIfSwitch

Parameters included	
Parameter name	Multiplicity
EthIfSwitchIdx	1..1
EthIfSwitchRef	1..1

Parameter Name	EthIfSwitchIdx
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 an Ethernet switch.
Multiplicity	1..1
Type	INTEGER
Range	<=255 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfSwitchRef
Description	Reference to an Ethernet switch, which is handled by a specific Ethernet switch driver.
Multiplicity	1..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.3.1.9. EthIfSwitchMgmtInfoIndicationConfig

Parameters included	
Parameter name	Multiplicity
EthIfSwitchMgmtInfoIndicationFunction	1..1



Parameter Name	EthIfSwitchMgmtInfoIndicationFunction	
Multiplicity	1..1	
Type	FUNCTION-NAME	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.10. EthIfSwitchPortGroup

Parameters included	
Parameter name	Multiplicity
EthIfSwitchPortGroupIdx	1..1
EthIfSwitchPortGroupRefSemantics	0..1
EthIfPortRef	1..n

Parameter Name	EthIfSwitchPortGroupIdx	
Multiplicity	1..1	
Type	INTEGER	
Range	<=255 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfSwitchPortGroupRefSemantics	
Multiplicity	0..1	
Type	ENUMERATION	
Range	ETHIF_SWITCH_PORT_GROUP_CONTROL ETHIF_SWITCH_PORT_GROUP_LINK_INFO	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfPortRef	
Multiplicity	1..n	
Type	SYMBOLIC-NAME-REFERENCE	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.11. EthIfSwitchTimeStampIndicationConfig

Parameters included	
Parameter name	Multiplicity
EthIfSwitchEgressTimeStampIndicationFunction	1..1
EthIfSwitchIngressTimeStampIndicationFunction	1..1

Parameter Name	EthIfSwitchEgressTimeStampIndicationFunction
Multiplicity	1..1
Type	FUNCTION-NAME
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

Parameter Name	EthIfSwitchIngressTimeStampIndicationFunction
Multiplicity	1..1
Type	FUNCTION-NAME
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

5.3.1.12. EthIfTransceiver

Parameters included	
Parameter name	Multiplicity
EthIfTransceiverIdx	1..1
EthIfEthTrcvRef	0..1
EthIfWEthTrcvRef	0..1

Parameter Name	EthIfTransceiverIdx
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.



Multiplicity	1..1
Type	INTEGER
Range	<=255 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfEthTrcvRef
Description	Reference to an Ethernet transceiver that is handled by a specific Ethernet transceiver driver
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfWEthTrcvRef
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.3.1.13. EthIfTrcvLinkStateChgConfig

Parameters included	
Parameter name	Multiplicity
EthIfTrcvLinkStateChgFunction	1..1

Parameter Name	EthIfTrcvLinkStateChgFunction
Description	Specifies a link state change callback function Note: Enter EthIf_Up_TrsvLinkStateChgDummy if no link state change function shall be called.
Multiplicity	1..1
Type	FUNCTION-NAME



Default value	EthSM_TrcvLinkStateChg	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.14. EthIfTxConfirmationConfig

Parameters included	
Parameter name	Multiplicity
EthIfTxConfirmationFunction	1..1

Parameter Name	EthIfTxConfirmationFunction
Description	Specifies a transmit confirmation callback function Note: Enter EthIf_Up_TxConfirmationDummy if no transmit confirmation function shall be called.
Multiplicity	1..1
Type	FUNCTION-NAME
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

5.3.1.15. EthIfEthControllerType

Parameters included	
Parameter name	Multiplicity
EthIfEthControllerBswmdlImplementationRefs	1..1
EthIfEthCtrlTypeRef	1..1

Parameter Name	EthIfEthControllerBswmdlImplementationRefs
Description	Reference to the BSW implementation of the underlying controller that contains the vendor ID and vendorApiInfix. To be configured only when the support of multiple Eth controllers is required, or when the Eth controller that contains the vendor ID and vendorApiInfix is used. Supported for Eth controller versions 4.3.-0. and above.
Multiplicity	1..1
Type	FOREIGN-REFERENCE



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfEthCtrlTypeRef	
Description	Reference to the Ethernet Controller.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.3.1.16. EthIfEthTrcvType

Parameters included	
Parameter name	Multiplicity
EthIfEthTrcvBswmdlImplementationRefs	1..1
EthIfEthTrcvTypeRef	1..1

Parameter Name	EthIfEthTrcvBswmdlImplementationRefs	
Description	Reference to the BSW implementation of the underlying transceiver that contains the vendor ID and vendorApiInfix. To be configured only when the support of multiple Eth transceivers is required, or when the Eth Transceiver that contains the vendor ID and vendorApiInfix is used. Supported for Eth transceiver version 4.3.0. and above.	
Multiplicity	1..1	
Type	FOREIGN-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfEthTrcvTypeRef	
Description	Reference to the Ethernet Transceiver.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	



5.3.1.17. EthIfEthSwtType

Parameters included	
Parameter name	Multiplicity
EthIfEthSwtBswmdImplementationRefs	1..1
EthIfEthSwtTypeRef	1..1

Parameter Name	EthIfEthSwtBswmdImplementationRefs	
Description	Reference to the BSW implementation of the underlying switch that contains the vendor ID and vendorApiInfix. To be configured only when the support of the Eth switch that contains the vendor ID and vendorApiInfix is used. Supported for Eth switch version 4.3.0. and above.	
Multiplicity	1..1	
Type	FOREIGN-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfEthSwtTypeRef	
Description	Reference to the Ethernet Switch.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.3.1.18. EthIfGeneral

Containers included		
Container name	Multiplicity	Description
EthIfSecurityEventRefs	1..1	Container for the references to IdsMEvent elements representing the security events that the EthIf module shall report to the IdsM in case the corresponding security related event occurs (and if EthIfEnableSecurityEventReporting is set to "true").
ReportToDem	1..1	Label: Production error handling

**Containers included**

		Production error handling
--	--	---------------------------

Parameters included

Parameter name	Multiplicity
EthIfDevErrorDetect	1..1
EthIfEnableRxInterrupt	1..1
EthIfEnableSignalQualityApi	1..1
EthIfEnableTxInterrupt	1..1
EthIfEnableWEthApi	0..1
EthIfGetAndResetMeasurementDataApi	1..1
EthIfGetBaudRate	1..1
EthIfGetCounterState	1..1
EthIfGetCtrlIdxList	1..1
EthIfGetTransceiverWakeupModeApi	0..1
EthIfGetVlanIdSupport	1..1
EthIfGlobalTimeSupport	1..1
EthIfMainFunctionPeriod	1..1
EthIfMainFunctionStatePeriod	0..1
EthIfMaxTrcvTotal	1..1
EthIfPortStartupActiveTime	0..1
EthIfPublicCddHeaderFile	0..255
EthIfRxIndicationIterations	1..1
EthIfSetForwardingModeApi	1..1
EthIfStartAutoNegotiation	1..1
EthIfSwitchManagementSupport	1..1
EthIfSwitchOffPortTimeDelay	0..1
EthIfTrcvLinkStateChgMainReload	1..1
EthIfVerifyConfigApi	1..1
EthIfVersionInfoApi	1..1
EthIfVersionInfoApiMacro	1..1
EthIfWakeUpSupport	1..1
EthIfEnableSecurityEventReporting	1..1



Parameters included

EthIfMiiApiEnable	1..1
EthIfSwitchingPortGroupSupport	1..1
EthIfAsyncEthTrcvModeSupport	1..1
EthIfAsyncEthCtrlModeSupport	1..1
EthIfPublicHandleTypeEnum	1..1
EthIfSupportEthAPI	1..1
EthIfMaxCtrl	1..1
EthIfMaxPhyCtrl	1..1
EthIfMaxEthSwitches	1..1
EthIfMaxSwtPorts	1..1
EthIfMaxSwtPortGroups	1..1
EthIfRelocatablePbcfgEnable	1..1
EthIfSetPhysAddrSupportEnable	1..1
EthIfTrcvSupportEnable	1..1
EthIfUpdatePhysAddrFilterSupportEnable	1..1
EthIfVirtualCtrlSupportEnable	1..1
EthIfVLANSupportEnable	1..1
EthIfGetArlTableApi	1..1
EthIfGetBufferLevelApi	1..1
EthIfSwtGetCounterValuesApi	1..1
EthIfGetPortMacAddrApi	1..1
EthIfResetConfigurationApi	1..1
EthIfStoreConfigurationApi	1..1
EthIfSetModeTimeout	0..1
EthIfInitControllersTransceivers	1..1
EthIfSwtPreProcessRxFrame	1..1
EthIfSwtAdpatTxFrame	1..1
EthIfDeviceAuthenticationApiEnable	1..1
EthIfRetransmitApiEnable	1..1

Parameter Name	EthIfDevErrorDetect
Description	Enables/disables development error detection



Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfEnableRxInterrupt
Description	Enables/disables the receive interrupt This configuration parameter is not used.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfEnableSignalQualityApi
Description	Enable/disable the APIs to read and clear the signal quality
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfEnableTxInterrupt
Description	Enables/disables the transmit interrupt This configuration parameter is not used.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfEnableWEthApi
-----------------------	---------------------------



Description	Enables/disables APIs for WEth / WEthTrcv This configuration parameter is not used.
Multiplicity	0..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfGetAndResetMeasurementDataApi
Description	Enables/disables the get and reset measurement data API
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfGetBaudRate
Description	Enables/disables the GetBaudRate API This configuration parameter is not used.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfGetCounterState
Description	Enables/disables the GetCounterState API This configuration parameter is not used.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild



Origin	AUTOSAR_ECUC
--------	--------------

Parameter Name	EthIfGetCtrlIdxList	
Description	Enables/disables the GetCtrlIdxList API This configuration parameter is not used.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfGetTransceiverWakeupModeApi	
Description	Enables/disables the EthIf_GetTransceiverWakeupMode API	
Multiplicity	0..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfGetVlanIdSupport	
Description	Enables/disables the GetVlanId API	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfGlobalTimeSupport	
Description	Enables/disables the global time APIs used amongst others by global time synchronization over Ethernet	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfMainFunctionPeriod	
Description	Specifies the period of the functions EthIf_MainFunctionRx() and EthIf_MainFunctionTx() in seconds Ethernet Interface does not require this information, but the BSW scheduler does.	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.01	
Range	<=65535 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfMainFunctionStatePeriod	
Description	Specifies the period of the main function EthIf_MainFunctionState() in seconds. Ethernet Interface does not require this information, but the BSW scheduler does.	
Multiplicity	0..1	
Type	FLOAT	
Range	<=65535 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfMaxTrcvTotal	
Description	Limits the total number of transceivers	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=255	



	>=1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfPortStartupActiveTime	
Description	Time delay after which all ports started with the API call EthIf_StartAllPorts() are switched off.	
Multiplicity	0..1	
Type	FLOAT	
Range	<=65.535 >=0.001	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfPublicCddHeaderFile	
Description	Defines the header files for callback functions that shall be included in case of CDDs. Range of characters is 1..32.	
Multiplicity	0..255	
Type	STRING	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfRxIndicationIterations	
Description	Maximum number of Ethernet frames per Ethernet controller polled from the Ethernet driver within EthIf_MainFunctionRx()	
Multiplicity	1..1	
Type	INTEGER	
Default value	10	
Range	<=65535 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfSetForwardingModeApi	
-----------------------	----------------------------------	--



Description	Enables/disables the EthIf_SetForwardingMode() API This configuration parameter is not used.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfStartAutoNegotiation	
Description	Enables/disables the StartAutoNegotiation API This configuration parameter is not used.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfSwitchManagementSupport	
Description	Enables/disables the switch management APIs to support a switch-port specific communication attribute access This configuration parameter is not used.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthIfSwitchOffPortTimeDelay	
Description	Specifies the time delay after which the mode ETHTRCV_MODE_DOWN of a EthIfSwitchPortGroup is executed. This is only used for EthIfSwtPortGroups that are not referenced by an EthIfController or the reference is of type "link-information".	
Multiplicity	0..1	
Type	FLOAT	



Range	<=65.535 >=0.001
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfTrcvLinkStateChgMainReload
Description	Specifies the frequency of transceiver link state change checks in each period of the main function EthIf_MainFunctionTx() This configuration parameter is not used.
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<=255 >=1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfVerifyConfigApi
Description	Enables/disables the EthIf_VerifyConfig API This configuration parameter is not used.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfVersionInfoApi
Description	Enables/disables the version info API
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild



Origin	AUTOSAR_ECUC
---------------	--------------

Parameter Name	EthIfVersionInfoApiMacro
Description	Enables/disables the version info API macro implementation This configuration parameter is not used.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfWakeUpSupport
Description	Configures if Ethernet wake-up is supported or not
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	EthIfEnableSecurityEventReporting
Description	Enables/disables reporting of security events to the IdsM
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthIfMiiApiEnable
Description	Enables/disables the EthIf_ReadMii() and EthIf_WriteMii() APIs
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild



Origin	Elektrobit Automotive GmbH	
Parameter Name	EthIfSwitchingPortGroupSupport	
Description	Enables/disables switching of the switch port groups feature	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	
Parameter Name	EthIfAsyncEthTrcvModeSupport	
Description	Switch between sync/async behavior of EthTrcv <ul style="list-style-type: none"> ▶ True: Asynchronous communication with EthTrcv - callback EthIf_TrcvModeIndication - is enabled. ▶ False: Asynchronous communication with EthTrcv - callback EthIf_TrcvModeIndication - is disabled. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	
Parameter Name	EthIfAsyncEthCtrlModeSupport	
Description	Switch between sync/async behavior of Eth Controllers <ul style="list-style-type: none"> ▶ True: Asynchronous communication with EthCtrl - callback EthIf_CtrlModeIndication - is enabled. ▶ False: Asynchronous communication with EthCtrl - callback EthIf_CtrlModeIndication - is disabled. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	
Parameter Name	EthIfPublicHandleTypeEnum	



Description	Configures the type Eth_BufldxType The type Eth_BufldxType represents the hardware buffer handle of an Eth hardware buffer. This configuration parameter must be configured according to the Eth driver's needs.	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	UINT8	
Range	UINT32 UINT8	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfSupportEthAPI	
Description	Configures the support of the correct version of the Eth API, according to the version of AUTOSAR ASR403: AUTOSAR 4.0.3 ASR422: AUTOSAR 4.2.2 ASR430: AUTOSAR 4.3.0 ASR430_EB: compatible with current 4.3.0 EB Eth drivers	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	ASR430	
Range	ASR403 ASR422 ASR430 ASR430_EB	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfMaxCtrl	
Description	Limits the total number of controllers	
Multiplicity	1..1	



Type	INTEGER	
Default value	1	
Range	<=255 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfMaxPhyCtrl	
Description	Limits the total number of used physical Ethernet controllers (Eth)	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=255 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfMaxEthSwitches	
Description	Limits the total number of Ethernet switches	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255 =>0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfMaxSwtPorts	
Description	Limits the total number of switch ports	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255	



	>=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthIfMaxSwtPortGroups
Description	Limits the total number of switch port groups
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<=255 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthIfRelocatablePbcfgEnable
Description	Enables/disables support for relocatable post-build configuration <ul style="list-style-type: none"> ▶ True: Post-build configuration relocatable in memory ▶ False: Post-build configuration not relocatable in memory
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthIfSetPhysAddrSupportEnable
Description	Enables/disables the use of Ethernet driver API Eth_SetPhysAddr() <ul style="list-style-type: none"> ▶ True: EthIf_SetPhysAddr() calls the Ethernet driver API Eth_SetPhysAddr(). ▶ False: EthIf_SetPhysAddr() returns without any lower layer calls. Eth_SetPhysAddr() is not required. <p>Note: Eth_SetPhysAddr() does not exist in AUTOSAR 4.0.3. Therefore, it is required to disable this configuration parameter for such Ethernet drivers.</p>
Multiplicity	1..1
Type	BOOLEAN



Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthIfTrcvSupportEnable
Description	<p>Switch to enable/disable EthTrcv support</p> <p>If disabled, all API functions accessing the EthTrcv are excluded.</p> <ul style="list-style-type: none"> ▶ True: The EthIf supports controlling the EthTrcv. ▶ False: The EthIf does not support controlling the EthTrcv. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (code): Enabling this parameter reduces the ROM consumption of the module code. ▶ Execution time reduction (code): Enabling this parameter reduces the execution time of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthIfUpdatePhysAddrFilterSupportEnable
Description	<p>Enables/disables the use of the Ethernet driver API Eth_UpdatePhysAddrFilter()</p> <ul style="list-style-type: none"> ▶ True: EthIf_UpdatePhysAddrFilter() calls the Ethernet driver API Eth_UpdatePhysAddrFilter(). ▶ False: EthIf_UpdatePhysAddrFilter() returns without any lower layer calls. Eth_UpdatePhysAddrFilter() is not required. <p>Note: Eth_UpdatePhysAddrFilter() does not exist in AUTOSAR 4.0.3. Therefore, it is required to disable this configuration parameter for such Ethernet drivers.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild



Origin	Elektrobit Automotive GmbH
---------------	----------------------------

Parameter Name	EthIfVirtualCtrlSupportEnable	
Description	<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"> ▶ True: The EthIf supports virtual EthIf Controllers. ▶ False: The EthIf does not support virtual EthIf Controllers. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code. ▶ Execution time reduction (code): Disabling this parameter reduces the execution time of the module code. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfVLANSupportEnable	
Description	<p>Switch to enable/disable VLAN support</p> <ul style="list-style-type: none"> ▶ True: The EthIf supports VLAN. ▶ False: The EthIf does not support VLAN. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfGetArlTableApi	
-----------------------	----------------------------	--



Description	Enables/disables the EthIf_GetArlTable() API	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfGetBufferLevelApi	
Description	Enables/disables the API to fetch the switch buffer utilization	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfSwtGetCounterValuesApi	
Description	Enables/disables the EthIf_SwtGetCounterValues() API	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfGetPortMacAddrApi	
Description	Enables/disables the EthIf_GetPortMacAddr() API	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfResetConfigurationApi	
Description	Enables/disables the EthIf_ResetConfiguration() API	



Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthIfStoreConfigurationApi
Description	Enables/disables the EthIf_StoreConfiguration() API
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthIfSetModeTimeout
Description	Specifies in which time duration the EthIf module shall repeat (delay) a call to set mode APIs.
Multiplicity	0..1
Type	FLOAT
Configuration class	PreCompile: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthIfInitControllersTransceivers
Description	Switch to enable/disable EthIf to initialize configured physical controllers and transceivers. This config parameter shall be disabled for AUTOSAR 4.2.2 and higher Eth drivers and transceivers.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthIfSwtPreProcessRxFrame
Description	Switch to enable/disable the call of EthSwt_EthIfPreProcessRxFrame() to allow the switch driver to preprocess received frames. This is necessary for frames



	<p>that 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"> ▶ True: EthSwt_EthIfPreProcessRxFrame() is called. ▶ False: No preprocessing is performed. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ Execution time reduction (code): Disabling this parameter reduces the execution time of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthIfSwtAdpatTxFrame
Description	<p>Switch to enable/disable the functionality to extend outgoing frames with additional switch-specific data</p> <ul style="list-style-type: none"> ▶ True: EthIf calls EthSwt APIs to modify Tx frames. ▶ False: No Tx frame modification is performed. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ Execution time reduction (code): Disabling this parameter reduces the execution time of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthIfDeviceAuthenticationApiEnable
Description	Enables/disables EthIf_EnableRelatedEthIfCtrls() and EthIf_DisableRelatedEthIfCtrls() APIs
Multiplicity	1..1
Type	BOOLEAN
Default value	false



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfRetransmitApiEnable	
Description	Enables/disables the EthIf_Retransmit() API	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.3.1.19. EthIfSecurityEventRefs

Parameters included	
Parameter name	Multiplicity
ETHIF_SEV_DROP_UNKNOWN_ETHERTYPE	0..1
ETHIF_SEV_DROP_VLAN_DOUBLE_TAG	0..1
ETHIF_SEV_DROP_INV_VLAN	0..1

Parameter Name	ETHIF_SEV_DROP_UNKNOWN_ETHERTYPE	
Description	Reference to the IdsMEvent ETHIF_SEV_DROP_UNKNOWN_ETHERTYPE: An ethernet datagram was dropped due to the unknown EtherType.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	ETHIF_SEV_DROP_VLAN_DOUBLE_TAG	
Description	Reference to the IdsMEvent ETHIF_SEV_DROP_VLAN_DOUBLE_TAG: An ethernet datagram was dropped due to double VLAN tag.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	



Parameter Name	ETHIF_SEV_DROP_INV_VLAN	
Description	Reference to the IdsMEvent ETHIF_SEV_DROP_INV_VLAN: An ethernet data-frame was dropped due to an invalid CrtlIdx/VLAN.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.3.1.20. ReportToDem

Parameters included	
Parameter name	Multiplicity
EthIfDemCtrlTestResultReportToDem	1..1
EthIfDemCtrlTestResultReportToDemDetErrorId	1..1

Parameter Name	EthIfDemCtrlTestResultReportToDem
Label	EthIfDemCtrlTestResult report to
Description	<p>Selects the handling of the production error EthIfDemCtrlTestResult</p> <ul style="list-style-type: none"> ▶ DEM: The error is reported to the Diagnostics Event Manager (Dem). ▶ DET: The error is reported to the Development Error Tracer (Det) if enabled. ▶ DISABLE: The error is not reported at all. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (code): Setting this parameter to a value of DISABLE reduces the ROM consumption of the module code. ▶ Execution time reduction (code): Setting this parameter to the value of DISABLE reduces the execution time of the module code.
Multiplicity	1..1
Type	ENUMERATION
Default value	DISABLE
Range	DEM DET DISABLE



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	
Parameter Name	EthIfDemCtrlTestResultReportToDemDetErrorId	
Label	EthIfDemCtrlTestResult Dem To Det error ID	
Description	If a production error is reported towards the Det, EthIfDemCtrlTestResultReportToDemDetErrorId defines the error ID that is reported towards the Det.	
Multiplicity	1..1	
Type	INTEGER	
Default value	145	
Range	<=255 >=0	
Configuration class	PreCompile:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.3.1.21. EthIfDefensiveProgramming

Parameters included	
Parameter name	Multiplicity
EthIfDefProgEnabled	1..1
EthIfPrecondAssertEnabled	1..1
EthIfPostcondAssertEnabled	1..1
EthIfStaticAssertEnabled	1..1
EthIfUnreachAssertEnabled	1..1
EthIfInvariantAssertEnabled	1..1

Parameter Name	EthIfDefProgEnabled
Label	Enable Defensive Programming
Description	<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"> 1. Enable development error detection 2. Enable defensive programming



	3. Enable assertions as required
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthIfPrecondAssertEnabled
Label	Enable Precondition Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>EthIfDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>EthIfDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthIfPostcondAssertEnabled
Label	Enable Postcondition Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>EthIfDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>EthIfDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfStaticAssertEnabled	
Label	Enable Static Assertions	
Description	<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"> ▶ Enable Development Error Detection (<code>EthIfDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>EthIfDefProgEnabled</code>): must be enabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfUnreachAssertEnabled	
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"> ▶ Enable Development Error Detection (<code>EthIfDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>EthIfDefProgEnabled</code>): must be enabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthIfInvariantAssertEnabled	
-----------------------	------------------------------------	--



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"> ▶ Enable Development Error Detection (<code>EthIfDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>EthIfDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.3.1.22. PublishedInformation

Parameters included	
Parameter name	Multiplicity
PbcfgMSupport	1..1
Parameter Name	
PbcfgMSupport	
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	PublishedInformation:
Origin	Elektrobit Automotive GmbH

5.3.2. Recommended configurations



5.3.2.1. EthIfRecConfiguration

Containers included	
Container name	Container definition
EthIfConfigSet	EthIfConfigSet

Parameters included	
Parameter name	Value

5.3.2.1.1. EthIfConfigSet

Containers included	
Container name	Container definition
EthIfController_0	EthIfController
EthIfFrameOwnerConfig_IPv4	EthIfFrameOwnerConfig
EthIfFrameOwnerConfig_IPv6	EthIfFrameOwnerConfig
EthIfFrameOwnerConfig_ARP	EthIfFrameOwnerConfig
EthIfPhysController_0	EthIfPhysController
EthIfRxIndicationConfig_IPv4	EthIfRxIndicationConfig
EthIfRxIndicationConfig_IPv6	EthIfRxIndicationConfig
EthIfRxIndicationConfig_ARP	EthIfRxIndicationConfig
EthIfTrcvLinkStateChgConfig_EthSM	EthIfTrcvLinkStateChgConfig
EthIfTxConfirmationConfig_0	EthIfTxConfirmationConfig

Parameters included	
Parameter name	Value

5.3.2.1.2. EthIfController_0

Parameters included	
Parameter name	Value
EthIfCtrlIdx	0



5.3.2.1.3. EthIfFrameOwnerConfig_IPv4

Parameters included	
Parameter name	Value
EthIfFrameType	2048
EthIfOwner	0

5.3.2.1.4. EthIfFrameOwnerConfig_IPv6

Parameters included	
Parameter name	Value
EthIfFrameType	34525
EthIfOwner	1

5.3.2.1.5. EthIfFrameOwnerConfig_ARP

Parameters included	
Parameter name	Value
EthIfFrameType	2054
EthIfOwner	2

5.3.2.1.6. EthIfPhysController_0

Parameters included	
Parameter name	Value
EthIfPhysControllerIdx	0

5.3.2.1.7. EthIfRxIndicationConfig_IPv4

Parameters included	
Parameter name	Value
EthIfRxIndicationFunction	TcpIp_RxIndication



5.3.2.1.8. EthIfRxIndicationConfig_IPv6

Parameters included	
Parameter name	Value
EthIfRxIndicationFunction	TcpIp_RxIndication

5.3.2.1.9. EthIfRxIndicationConfig_ARP

Parameters included	
Parameter name	Value
EthIfRxIndicationFunction	TcpIp_RxIndication

5.3.2.1.10. EthIfTrcvLinkStateChgConfig_EthSM

Parameters included	
Parameter name	Value
EthIfTrcvLinkStateChgFunction	EthSM_TrsvLinkStateChg

5.3.2.1.11. EthIfTxConfirmationConfig_0

Parameters included	
Parameter name	Value
EthIfTxConfirmationFunction	EthIf_Up_TxConfirmationDummy

5.3.2.2. EthIfRecConfigurationEthTSyn

Containers included	
Container name	Container definition
EthIfConfigSet	EthIfConfigSet
EthIfGeneral	EthIfGeneral

Parameters included	
Parameter name	Value



5.3.2.2.1. EthIfConfigSet

Containers included	
Container name	Container definition
EthIfController_0	EthIfController
EthIfFrameOwnerConfig_PTP	EthIfFrameOwnerConfig
EthIfFrameOwnerConfig_IPv4	EthIfFrameOwnerConfig
EthIfFrameOwnerConfig_IPv6	EthIfFrameOwnerConfig
EthIfFrameOwnerConfig_ARP	EthIfFrameOwnerConfig
EthIfPhysController_0	EthIfPhysController
EthIfRxIndicationConfig_PTP	EthIfRxIndicationConfig
EthIfRxIndicationConfig_IPv4	EthIfRxIndicationConfig
EthIfRxIndicationConfig_IPv6	EthIfRxIndicationConfig
EthIfRxIndicationConfig_ARP	EthIfRxIndicationConfig
EthIfTrcvLinkStateChgConfig_EthSM	EthIfTrcvLinkStateChgConfig
EthIfTrcvLinkStateChgConfig_EthTSyn	EthIfTrcvLinkStateChgConfig
EthIfTxConfirmationConfig_PTP	EthIfTxConfirmationConfig

Parameters included

Parameter name	Value

5.3.2.2.2. EthIfController_0

Parameters included	
Parameter name	Value
EthIfCtrlIdx	0

5.3.2.2.3. EthIfFrameOwnerConfig_PTP

Parameters included	
Parameter name	Value
EthIfFrameType	35063
EthIfOwner	0



5.3.2.2.4. EthIfFrameOwnerConfig_IPv4

Parameters included	
Parameter name	Value
EthIfFrameType	2048
EthIfOwner	1

5.3.2.2.5. EthIfFrameOwnerConfig_IPv6

Parameters included	
Parameter name	Value
EthIfFrameType	34525
EthIfOwner	2

5.3.2.2.6. EthIfFrameOwnerConfig_ARP

Parameters included	
Parameter name	Value
EthIfFrameType	2054
EthIfOwner	3

5.3.2.2.7. EthIfPhysController_0

Parameters included	
Parameter name	Value
EthIfPhysControllerIdx	0

5.3.2.2.8. EthIfRxIndicationConfig_PTP

Parameters included	
Parameter name	Value
EthIfRxIndicationFunction	EthTSyn_RxIndication



5.3.2.2.9. EthIfRxIndicationConfig_IPv4

Parameters included	
Parameter name	Value
EthIfRxIndicationFunction	TcpIp_RxIndication

5.3.2.2.10. EthIfRxIndicationConfig_IPv6

Parameters included	
Parameter name	Value
EthIfRxIndicationFunction	TcpIp_RxIndication

5.3.2.2.11. EthIfRxIndicationConfig_ARP

Parameters included	
Parameter name	Value
EthIfRxIndicationFunction	TcpIp_RxIndication

5.3.2.2.12. EthIfTrcvLinkStateChgConfig_EthSM

Parameters included	
Parameter name	Value
EthIfTrcvLinkStateChgFunction	EthSM_TrcvLinkStateChg

5.3.2.2.13. EthIfTrcvLinkStateChgConfig_EthTSyn

Parameters included	
Parameter name	Value
EthIfTrcvLinkStateChgFunction	EthTSyn_TrcvLinkStateChg

5.3.2.2.14. EthIfTxConfirmationConfig_PTP

Parameters included	
Parameter name	Value



Parameters included

EthIfTxConfirmationFunction	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 the API ID of function EthIf_Cbk_RxIndication() .
Value	0x10U

5.3.3.1.2. ETHIF_CBK_TXCONFIRMATION_SVCID

Purpose	Defines the API ID of function EthIf_Cbk_TxConfirmation() .
Value	0x11U

5.3.3.1.3. ETHIF_CHECKWAKEUP_SVCID

Purpose	Defines the API ID of function EthIf_CheckWakeup() .
Value	0x30U

5.3.3.1.4. ETHIF_CLEARSWITCHPORTSIGNALQUALITY_SVCID

Purpose	Defines the API ID of function EthIf_ClearSwitchPortSignalQuality() .
Value	0x1bU



5.3.3.1.5. ETHIF_CLEARTRCVSIGNALQUALITY_SVCID

Purpose	Defines the API ID of function EthIf_ClearTrcvSignalQuality() .
Value	0x19U

5.3.3.1.6. ETHIF_CTRLMODEINDICATION_SVCID

Purpose	Defines the API ID of function EthIf_CtrlModeIndication() .
Value	0x0EU

5.3.3.1.7. ETHIF_DISABLERELATEDETHIFCTRLS_SVCID

Purpose	Defines the API ID of function EthIf_DisableRelatedEthIfCtrls() .
Value	0xF1U

5.3.3.1.8. ETHIF_ENABLEEGRESSTIMESTAMP_SVCID

Purpose	Defines the API ID of function EthIf_EnableEgressTimeStamp() .
Value	0x23U

5.3.3.1.9. ETHIF_ENABLERELATEDETHIFCTRLS_SVCID

Purpose	Defines the API ID of function EthIf_EnableRelatedEthIfCtrls() .
Value	0xF0U

5.3.3.1.10. ETHIF_E_HW_NOT_INITIALIZED

Purpose	API requests called with invalid parameter.
Value	0x80U

5.3.3.1.11. ETHIF_E_INIT_FAILED

Purpose	Initialization of EthIf module failed.
Value	0x07U



5.3.3.1.12. ETHIF_E_INV_CTRL_IDX

Purpose	API requests called with invalid controller index.
Value	0x01U

5.3.3.1.13. ETHIF_E_INV_PARAM

Purpose	API requests called with invalid parameter.
Value	0x06U

5.3.3.1.14. ETHIF_E_INV_PORT_GROUP_IDX

Purpose	API requests called with invalid port group index.
Value	0x03U

5.3.3.1.15. ETHIF_E_INV_SWT_IDX

Purpose	API requests called with invalid switch index.
Value	0x90U

5.3.3.1.16. ETHIF_E_INV_TRCV_IDX

Purpose	API requests called with invalid transceiver index.
Value	0x02U

5.3.3.1.17. ETHIF_E_NOT_INITIALIZED

Purpose	API requests called before Ethif module is initialized.
Value	0x04U

5.3.3.1.18. ETHIF_E_PARAM_POINTER

Purpose	API requests called with invalid pointer in parameter list.
Value	0x05U



5.3.3.1.19. ETHIF_GETANDRESETMEASUREMENTDATA_SVCID

Purpose	Defines the API ID of function EthIf_GetAndResetMeasurementData() .
Value	0x45U

5.3.3.1.20. ETHIF_GETARLTABLE_SVCID

Purpose	Defines the API ID of function EthIf_GetArlTable() .
Value	0x29U

5.3.3.1.21. ETHIF_GETBUFFERLEVEL_SVCID

Purpose	Defines the API ID of function EthIf_GetBufferLevel() .
Value	0x2AU

5.3.3.1.22. ETHIF_GETCABLEDIAGNOSTICSRESULT_SVCID

Purpose	Defines the API ID of function EthIf_GetCableDiagnosticsResult() .
Value	0x14U

5.3.3.1.23. ETHIF_GETCONTROLLERMODE_SVCID

Purpose	Defines the API ID of function EthIf_GetControllerMode() .
Value	0x04U

5.3.3.1.24. ETHIF_GETCTRLIDXLIST_SVCID

Purpose	Defines the API ID of function EthIf_GetCtrlIdxList() .
Value	0x44U

5.3.3.1.25. ETHIF_GETCURRENTTIME_SVCID

Purpose	Defines the API ID of function EthIf_GetCurrentTime() .
Value	0x22U



5.3.3.1.26. ETHIF_GETEGRESSSTIMESTAMP_SVCID

Purpose	Defines the API ID of function EthIf_GetEgressTimeStamp() .
Value	0x24U

5.3.3.1.27. ETHIF_GETINGRESSTIMESTAMP_SVCID

Purpose	Defines the API ID of function EthIf_GetIngressTimeStamp() .
Value	0x25U

5.3.3.1.28. ETHIF_GETPHYIDENTIFIER_SVCID

Purpose	Defines the API ID of function EthIf_GetPhyIdentifier() .
Value	0x15U

5.3.3.1.29. ETHIF_GETPHYSADDR_SVCID

Purpose	Defines the API ID of function EthIf_GetPhysAddr() .
Value	0x08U

5.3.3.1.30. ETHIF_GETPORTMACADDR_SVCID

Purpose	Defines the API ID of function EthIf_GetPortMacAddr() .
Value	0x28U

5.3.3.1.31. ETHIF_GETSWITCHPORTSIGNALQUALITY_SVCID

Purpose	Defines the API ID of function EthIf_GetSwitchPortSignalQuality() .
Value	0x1aU

5.3.3.1.32. ETHIF_GETTRANSCEIVERWAKEUPMODE_SVCID

Purpose	Defines the API ID of function EthIf_GetTransceiverWakeUpMode() .
----------------	---



Value	0x2FU
--------------	-------

5.3.3.1.33. ETHIF_GETTRCVSIGNALQUALITY_SVCID

Purpose	Defines the API ID of function EthIf_GetTrcvSignalQuality() .
Value	0x18U

5.3.3.1.34. ETHIF_GETVERSIONINFO_SVCID

Purpose	Defines the API ID of function EthIf_GetVersionInfo() .
Value	0x0BU

5.3.3.1.35. ETHIF_GETVLANID_SVCID

Purpose	Defines the API ID of function EthIf_GetVlanId() .
Value	0x43U

5.3.3.1.36. ETHIF_INIT_SVCID

Purpose	Defines the API ID of function EthIf_Init() .
Value	0x01U

5.3.3.1.37. ETHIF_INSTANCE_ID

Purpose	Defines the instance number of this Ethernet Interface. Since multiple instances of Ethernet Interface are not supported, the instance ID is always zero.
Value	0U

5.3.3.1.38. ETHIF_MAINFUNCTIONRX_SVCID

Purpose	Defines the API ID of function EthIf_MainFunctionRx() .
Value	0x20U



5.3.3.1.39. ETHIF_MAINFUNCTIONSTATE_SVCID

Purpose	Defines the API ID of function EthIf_MainFunctionState() .
Value	0x05U

5.3.3.1.40. ETHIF_MAINFUNCTIONTX_SVCID

Purpose	Defines the API ID of function EthIf_MainFunctionTx() .
Value	0x21U

5.3.3.1.41. ETHIF_PROVIDETXBUFFER_SVCID

Purpose	Defines the API ID of function EthIf_ProvideTxBuffer() .
Value	0x09U

5.3.3.1.42. ETHIF_READMII_SVCID

Purpose	Defines the API ID of function EthIf_ReadMii() .
Value	0x70U

5.3.3.1.43. ETHIF_RESETCONFIGURATION_SVCID

Purpose	Defines the API ID of function EthIf_ResetConfiguration() .
Value	0x2DU

5.3.3.1.44. ETHIF_RETRANSMIT_SVCID

Purpose	Defines the API ID of function EthIf_Retransmit() .
Value	0xF2U

5.3.3.1.45. ETHIF_SETCONTROLLERMODE_SVCID

Purpose	Defines the API ID of function EthIf_SetControllerMode() .
Value	0x03U



5.3.3.1.46. ETHIF_SETCORRECTIONTIME_SVCID

Purpose	Defines the API ID of function EthIf_SetCorrectionTime() .
Value	0x26U

5.3.3.1.47. ETHIF_SETFORWARDINGMODE_SVCID

Purpose	Defines the API ID of function EthIf_SetForwardingMode() .
Value	0x41U

5.3.3.1.48. ETHIF_SETGLOBALTIME_SVCID

Purpose	Defines the API ID of function EthIf_SetGlobalTime() .
Value	0x27U

5.3.3.1.49. ETHIF_SETPHYLOOPBACKMODE_SVCID

Purpose	Defines the API ID of function EthIf_SetPhyLoopbackMode() .
Value	0x12U

5.3.3.1.50. ETHIF_SETPHYSADDR_SVCID

Purpose	Defines the API ID of function EthIf_SetPhysAddr() .
Value	0x0dU

5.3.3.1.51. ETHIF_SETPHYTESTMODE_SVCID

Purpose	Defines the API ID of function EthIf_SetPhyTestMode() .
Value	0x17U

5.3.3.1.52. ETHIF_SETPHYTXMODE_SVCID

Purpose	Defines the API ID of function EthIf_SetPhyTxMode() .
Value	0x13U



5.3.3.1.53. ETHIF_SETTRANSCEIVERWAKEUPMODE_SVCID

Purpose	Defines the API ID of function EthIf_SetTransceiverWakeUpMode() .
Value	0x2EU

5.3.3.1.54. ETHIF_STARTALLPORTS_SVCID

Purpose	Defines the API ID of function EthIf_StartAllPorts() .
Value	0x07U

5.3.3.1.55. ETHIF_STORECONFIGURATION_SVCID

Purpose	Defines the API ID of function EthIf_StoreConfiguration() .
Value	0x2CU

5.3.3.1.56. ETHIF_SWITCHMGMTINFOINDICATION_SVCID

Purpose	Defines the API ID of function EthIf_SwitchMgmtInfoIndication() .
Value	0x3AU

5.3.3.1.57. ETHIF_SWITCHPORTGROUPREQUESTMODE_SVCID

Purpose	Defines the API ID of function EthIf_SwitchPortGroupRequestMethod() .
Value	0x06U

5.3.3.1.58. ETHIF_SWTGETCOUNTERVALUES_SVCID

Purpose	Defines the API ID of function EthIf_SwtGetCounterValues() .
Value	0x40U

5.3.3.1.59. ETHIF_TRANSMIT_SVCID

Purpose	Defines the API ID of function EthIf_Transmit() .
Value	0x0AU



5.3.3.1.60. ETHIF_TRCVMODEINDICATION_SVCID

Purpose	Defines the API ID of function EthIf_TrcvModeIndication() .
Value	0x0FU

5.3.3.1.61. ETHIF_UPDATEPHYSADDRFILTER_SVCID

Purpose	Defines the API ID of function EthIf_UpdatePhysAddrFilter() .
Value	0x0cU

5.3.3.1.62. ETHIF_VERIFYCONFIG_SVCID

Purpose	Defines the API ID of function EthIf_VerifyConfig() .
Value	0x40U

5.3.3.1.63. ETHIF_WRITEMII_SVCID

Purpose	Defines the API ID of function EthIf_WriteMii() .
Value	0x71U

5.3.3.1.64. TS_RELOCATABLE_CFG_ENABLE

Purpose	Enable/disable relocateable config.
Value	ETHIF_RELOCATABLE_CFG_ENABLE

5.3.3.2. Functions

5.3.3.2.1. EthIf_Cbk_RxIndication

Purpose	Calls EthIf_RxIndication() .
Synopsis	<pre>void EthIf_Cbk_RxIndication (uint8 CtrlIdx , Eth_DataType * DataPtr , uint16 LenByte);</pre>



Service ID	0x10	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	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.
Description	This service directly calls EthIf_RxIndication() .	

5.3.3.2.2. EthIf_Cbk_TxConfirmation

Purpose	Tx confirmation callback function.	
Synopsis	<code>void EthIf_Cbk_TxConfirmation (uint8 CtrlIdx , Eth_BufIdxType BufIdx);</code>	
Service ID	0x11	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	CtrlIdx	Index of the Ethernet controller within the context of the Ethernet Interface.
	BufIdx	Index of the transmitted buffer.
Description	This service confirms the transmission of an Ethernet frame.	

5.3.3.2.3. EthIf_CheckWakeup

Purpose	Check a wake-up source.	
Synopsis	<code>Std_ReturnType EthIf_CheckWakeup (Ecum_WakeupSourceType WakeupSource);</code>	
Parameters (in)	WakeupSource	Source (transceiver) that initiated the wake-up event.
Return Value	Std_ReturnType	
	E_OK	The function was successfully executed.



	E_NOT_OK	The function could not be successfully executed.
Description	This service is called by integration code to check a wake-up source.	

5.3.3.2.4. EthIf_ClearSwitchPortSignalQuality

Purpose	Clears the stored signal quality.	
Synopsis	Std_ReturnType EthIf_ClearSwitchPortSignalQuality (uint8 SwitchIdx , uint8 SwitchPortIdx);	
Service ID	none	
Sync/Async	synchronous	
Reentrancy	Reentrant for different SwitchIdx and SwitchPortIdx. Non reentrant for the same SwitchIdx and SwitchPortIdx.	
Parameters (in)	SwitchIdx	Switch index within the context of the Ethernet Interface.
	SwitchPortIdx	Switch port index within the context of the Ethernet Interface.
Return Value	Std_ReturnType	
	E_OK	The request was accepted.
	E_NOT_OK	The request was not accepted.
Description	This function clears the stored signal quality of the link for the indexed switch port.	

5.3.3.2.5. EthIf_CtrlModeIndication

Purpose	Callback function to indicate a controller mode change.	
Synopsis	void EthIf_CtrlModeIndication (uint8 CtrlIdx , Eth_ModeType CtrlMode);	
Service ID	0x0E	
Sync/Async	Synchronous	
Reentrancy	Non reentrant for the same CtrlIdx. Reentrant for different CtrlIdx.	
Parameters (in)	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
Description	Called asynchronously when the mode is read out. Triggered by the previous Eth_SetControllerMode() call. Can be called directly within the trigger functions.	

5.3.3.2.6. EthIf_DisableRelatedEthIfCtrls

Purpose	Within a group of EthIfControllers that reference the same PhyController, disable all but one received as a parameter.	
Synopsis	Std_ReturnType EthIf_DisableRelatedEthIfCtrls (uint8 CtrlIdx);	
Parameters (in)	CtrlIdx	Index of the controller within the context of the EthIf.
Return Value	Std_ReturnType	
	E_OK	
	E_NOT_OK	{}

5.3.3.2.7. EthIf_EnableEgressTimeStamp

Purpose	Service to enable the egress time stamp.	
Synopsis	void EthIf_EnableEgressTimeStamp (uint8 CtrlIdx , Eth_BufIdx- Type BufIdx);	
Service ID	0x23	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	CtrlIdx	EthIf controller index.
	BufIdx	Index of the used transmit buffer.

5.3.3.2.8. EthIf_EnableRelatedEthIfCtrls

Purpose	Within a group of EthIfControllers that reference the same PhyController, enable all but one received as a parameter.	
Synopsis	Std_ReturnType EthIf_EnableRelatedEthIfCtrls (uint8 CtrlIdx);	



Parameters (in)	CtrlIdx	Index of the controller within the context of the EthIf.
Return Value	Std_ReturnType	
	E_OK	
	E_NOT_OK	{}

5.3.3.2.9. EthIf_GetAndResetMeasurementData

Purpose	Reads and resets measurement data.					
Synopsis	<pre>Std_ReturnType EthIf_GetAndResetMeasurementData (EthIf_MeasurementIdxType MeasurementIdx, boolean MeasurementResetNeeded, uint32 * MeasurementDataPtr);</pre>					
Parameters (in)	MeasurementIdx	Index to select specific measurement data: <ul style="list-style-type: none"> ▶ 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.				
Parameters (out)	MeasurementDataPtr	Pointer to the data buffer where to copy measurement data.				
Return Value	Std_ReturnType <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">E_OK</td> <td style="padding: 2px;">The function was successfully executed.</td> </tr> <tr> <td style="padding: 2px;">E_NOT_OK</td> <td style="padding: 2px;">The function could not be successfully executed.</td> </tr> </table>		E_OK	The function was successfully executed.	E_NOT_OK	The function could not be successfully executed.
E_OK	The function was successfully executed.					
E_NOT_OK	The function could not be successfully executed.					



Description	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_GetAr1Table

Purpose	Service to obtain the address resolution table of a switch.	
Synopsis	Std_ReturnType EthIf_GetAr1Table (uint8 SwitchIdx , uint16 * numberOfElements , Eth_MacVlanType * arlTableListPointer);	
Service ID	0x29	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch driver.
Parameters (in,out)	numberOfElements	<ul style="list-style-type: none"> ▶ in: Maximum number of elements that can be written into the ARL table. ▶ out: Number of elements that are currently available in the EthSwitch module.
Parameters (out)	ArlTable	Returns the ARL table of the switch consisting of a list of structs with MAC address, VLAN ID, and port.
Return Value		

5.3.3.2.11. EthIf_GetBufferLevel

Purpose	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.	
Synopsis	Std_ReturnType EthIf_GetBufferLevel (uint8 SwitchIdx , uint32 * SwitchBufferLevelPtr);	
Service ID	0x2A	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch driver.
Parameters (out)	SwitchBufferLevelPtr	The interpretation of this value is switch-dependent.
Return Value		



5.3.3.2.12. EthIf_GetControllerMode

Purpose	Gets the Ethernet controller mode.	
Synopsis	Std_ReturnType EthIf_GetControllerMode (uint8 CtrlIdx , Eth_-ModeType * CtrlModePtr);	
Service ID	0x04	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	CtrlIdx	Index of the controller within the context of the EthIf.
Parameters (out)	CtrlModePtr	Address to set the current controller mode to.
Return Value	Std_ReturnType E_OK Service execution was successful. E_NOT_OK Service execution failed.	
Description	This service gets the Ethernet controller mode.	

5.3.3.2.13. EthIf_GetCtrlIdxList

Purpose	Gets the active Ethernet controllers.	
Synopsis	Std_ReturnType EthIf_GetCtrlIdxList (uint8 * NumberOfCtrlIdx , uint8 * CtrlIdxListPtr);	
Parameters (in,out)	NumberOfCtrlIdx	<ul style="list-style-type: none"> ▶ in: Maximum number of controllers in CtrlIdxListPtr, 0 to return the number of controllers but without filling CtrlIdxListPtr. ▶ out: Number of active controllers.
Parameters (out)	CtrlIdxListPtr	Pointer to a list of active controller indexes.
Return Value	Std_ReturnType E_OK The function was successfully executed. E_NOT_OK The function could not be successfully executed.	
Description	This service returns the number and index of all active Ethernet controllers.	



5.3.3.2.14. EthIf_GetCurrentTime

Purpose	Service to get the current hardware time.	
Synopsis	<pre>Std_ReturnType EthIf_GetCurrentTime (uint8 CtrlIdx , Eth_- Time StampQualType * timeQualPtr , Eth_TimeStampType * timeS- tampPtr);</pre>	
Parameters (in)	CtrlIdx	EthIf controller index.
Parameters (out)	timeQualPtr	Pointer containing the quality of the hard- ware time stamp.
	timeStampPtr	Pointer containing the current time.
Return Value	<pre>Std_ReturnType E_OK E_NOT_OK</pre>	
	E_OK	Service execution was successful.
	E_NOT_OK	Service execution failed.

5.3.3.2.15. EthIf_GetEgressTimeStamp

Purpose	Service to get the egress time stamp.	
Synopsis	<pre>Std_ReturnType EthIf_GetEgressTimeStamp (uint8 CtrlIdx , Eth_- BufIdxType BufIdx , Eth_TimeStampQualType * timeQualPtr , Eth_- Time StampType * timeStampPtr);</pre>	
Parameters (in)	CtrlIdx	EthIf controller index.
	BufIdx	Index of the used transmit buffer.
Parameters (out)	timeQualPtr	Pointer containing the quality of the hard- ware time stamp.
	timeStampPtr	Pointer containing the egress time stamp.
Return Value	<pre>Std_ReturnType E_OK E_NOT_OK</pre>	
	E_OK	Operation was successful.
	E_NOT_OK	Operation was not successful.

5.3.3.2.16. EthIf_GetIngressTimeStamp

Purpose	Service to get the ingress time stamp.	
Synopsis	<pre>Std_ReturnType EthIf_GetIngressTimeStamp (uint8 CtrlIdx , Eth_DataType * DataPtr , Eth_TimeStampQualType * timeQualPtr , Eth_TimeStampType * timeStampPtr);</pre>	



Parameters (in)	CtrlIdx	EthIf controller index.
	DataPtr	Pointer to the Rx data buffer of the requested frame.
Parameters (out)	timeQualPtr	Pointer containing the quality of the hardware time stamp.
	timeStampPtr	Pointer containing the ingress time stamp.
Return Value	Std_ReturnType	
	E_OK	Operation was successful.
	E_NOT_OK	Operation was not successful.

5.3.3.2.17. EthIf_GetPhysAddr

Purpose	Gets the local Ethernet physical address.	
Synopsis	<code>void EthIf_GetPhysAddr (uint8 CtrlIdx , uint8 * PhysAddrPtr);</code>	
Service ID	0x08	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	CtrlIdx	Index of the controller the transceiver is connected to within the context of the EthIf.
Parameters (out)	PhysAddrPtr	Address to write the physical address to.
Description	This service gets the local Ethernet physical device address.	

5.3.3.2.18. EthIf_GetPortMacAddr

Purpose	Service to obtain the port over which this MAC address can be reached.	
Synopsis	<code>Std_ReturnType EthIf_GetPortMacAddr (const uint8 * MacAddrPtr , uint8 * SwitchIdxPtr , uint8 * PortIdxPtr);</code>	
Service ID	0x28	
Parameters (in)	MacAddrPtr	MAC address for which a switch port is searched over which the node with this MAC address can be reached.
Parameters (out)	SwitchIdxPtr	Pointer to the switch index.
	PortIdxPtr	Pointer to the port index.



Return Value	
---------------------	--

5.3.3.2.19. EthIf_GetSwitchPortSignalQuality

Purpose	Obtains the switch port signal quality.	
Synopsis	<pre>Std_ReturnType EthIf_GetSwitchPortSignalQuality (uint8 SwitchIdx , uint8 SwitchPortIdx , EthIf_SignalQualityResultType * ResultPtr);</pre>	
Service ID	none	
Sync/Async	synchronous	
Reentrancy	Reentrant for different SwitchIdx and SwitchPortIdx. Non reentrant for the same SwitchIdx and SwitchPortIdx.	
Parameters (in)	SwitchIdx	Switch index within the context of the Ethernet Interface.
	SwitchPortIdx	Switch port index within the context of the Ethernet Interface.
Parameters (out)	ResultPtr	Pointer to the memory where the signal quality shall be stored.
Return Value	Std_ReturnType	
	E_OK	The request was accepted.
	E_NOT_OK	The request was not accepted.
Description	This function obtains the signal quality of the link for the indexed switch port.	

5.3.3.2.20. EthIf_GetTransceiverWakeupMode

Purpose	Gets the transceiver wake-up mode.	
Synopsis	<pre>Std_ReturnType EthIf_GetTransceiverWakeupMode (uint8 TrcvIdx , EthTrcv_WakeupModeType * TrcvWakeupModePtr);</pre>	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Ethernet Interface.
Parameters (out)	TrcvWakeupModePtr	<p>Pointer where transceiver wake-up mode is written:</p> <ul style="list-style-type: none"> ▶ EHTHTRCV_WUM_DISABLE: The transceiver wake-up is disabled.



		▶ ETHTRCV_WUM_ENABLE: The transceiver wake-up is enabled.
Return Value	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	Transceiver wake-up mode could not be obtained.
Description	This service returns the wake-up mode of the indexed transceiver.	

5.3.3.2.21. EthIf_GetVersionInfo

Purpose	Get version information of the Ethernet Interface.	
Synopsis	<code>void EthIf_GetVersionInfo (Std_VersionInfoType * VersionInfoPtr);</code>	
Service ID	0x0b	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	VersioninfoPtr	Pointer where to store the version information of this module.
Description	<p>This service returns the version information of this module. The version information includes:</p> <ul style="list-style-type: none"> ▶ Module ID ▶ Vendor ID ▶ Vendor-specific version numbers 	

5.3.3.2.22. EthIf_GetVlanId

Purpose	Gets the Ethernet controller VLAN identifier.	
Synopsis	<code>Std_ReturnType EthIf_GetVlanId (uint8 CtrlIdx , uint16 * VlanIdPtr);</code>	
Parameters (in)	CtrlIdx	Index of the Ethernet controller within the context of the Ethernet Interface.
Parameters (out)	VlanIdPtr	Pointer to store the VLAN identifier (VID) of the Ethernet controller. 0 if the the Eth-



		ernet controller represents no virtual network (VLAN).
Return Value	Std_ReturnType	
	E_OK	The function was successfully executed.
	E_NOT_OK	The function could not be successfully executed.
Description	This service returns the VLAN identifier of the requested Ethernet controller.	

5.3.3.2.23. EthIf_Init

Purpose	Initializes the EthIf module.	
Synopsis	<code>void EthIf_Init (const EthIf_ConfigType * CfgPtr);</code>	
Service ID	0x01	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	CfgPtr	Address of the post-build configuration data structure.
Description	This service initializes the EthIf module. It shall be the first function of the module that is called.	

5.3.3.2.24. EthIf_IsValidConfig

Purpose	Checks the compatibility of the post-build configuration.	
Synopsis	<code>Std_ReturnType EthIf_IsValidConfig (const void * voidConfigPtr);</code>	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	voidConfigPtr	Pointer to the configuration data of the EthIf module.
Return Value	Result of compatibility check	
	E_OK	Provided configuration is compatible.
	E_NOT_OK	Provided configuration is not compatible.
Description	This service checks the compatibility of the post-build configuration against the source code.	



5.3.3.2.25. EthIf_MainFunctionRx

Purpose	Main function for Rx tasks.
Synopsis	<code>void EthIf_MainFunctionRx (void);</code>
Service ID	0x20
Sync/Async	Synchronous
Reentrancy	Non reentrant
Description	This service performs the periodic tasks for Ethernet reception (polling the Ethernet driver).

5.3.3.2.26. EthIf_MainFunctionRxPrio

Purpose	Main function for priority receive processing tasks.
Synopsis	<code>void EthIf_MainFunctionRxPrio (uint8 PCtrlIngrFifoIdx);</code>
Service ID	0xXX
Sync/Async	Synchronous
Reentrancy	Non reentrant
Description	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_<EthIfCtr>_Prio()</code> .

5.3.3.2.27. EthIf_MainFunctionState

Purpose	Function to update the mode state and transceiver link state of EthIf objects.
Synopsis	<code>void EthIf_MainFunctionState (void);</code>
Service ID	0x05
Sync/Async	Asynchronous
Reentrancy	Non reentrant
Description	The function polls the link state of the used communication hardware (Ethernet transceiver, Ethernet switch ports). For active objects, it reads out/calculates the transceiver link state and reports it to upper layers (State Manager or BswM).



5.3.3.2.28. EthIf_MainFunctionTx

Purpose	Main function for Tx tasks.
Synopsis	<code>void EthIf_MainFunctionTx (void);</code>
Service ID	0x21
Sync/Async	Synchronous
Reentrancy	Non reentrant
Description	This service contains currently no functionality.

5.3.3.2.29. EthIf_ProvideTxBuffer

Purpose	Provide an Ethernet Tx buffer.	
Synopsis	<code>BufReq_ReturnType EthIf_ProvideTxBuffer (uint8 CtrlIdx , Eth_FrameType FrameType , uint8 Priority , Eth_BufIdxType * BufIdxPtr , EthIf_Uint8TypePtr * BufPtr , uint16 * LenBytePtr);</code>	
Service ID	0x09	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	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).
Parameters (in,out)	LenBytePtr	Address that holds the requested buffer length and after return of the call it holds the actual size of the returned Tx buffer.
Parameters (out)	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.
Return Value	Buffer request status	
	BUFREQ_OK	Service execution was successful.
	BUFREQ_E_NOT_OK	Service execution failed.
	BUFREQ_E_BUSY	Service currently not available.



Description	This service returns an Ethernet driver Tx buffer with the required length.
--------------------	---

5.3.3.2.30. EthIf_ReadMii

Purpose	Service to read from an Ethernet transceiver register.	
Synopsis	Std_ReturnType EthIf_ReadMii (uint8 CtrlIdx , uint8 RegIdx , uint16 * RegValPtr);	
Service ID	0x70	
Parameters (in)	CtrlIdx	EthIf controller index.
	RegIdx	Register index of the TrcVldx assigned to the CtrlIdx.
Parameters (out)	RegValPtr	Pointer containing the value of the RegIdx.
Return Value		

5.3.3.2.31. EthIf_ResetConfiguration

Purpose	Service to reset the configuration of the learned MAC/port tables of a switch in a persistent manner. This is used e.g. by a CDD. The statically configured entries shall still remain.	
Synopsis	Std_ReturnType EthIf_ResetConfiguration (uint8 SwitchIdx);	
Service ID	0x2D	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch driver.
Return Value		

5.3.3.2.32. EthIf_Retransmit

Purpose	Retransmit a received/transmitted frame.	
Synopsis	Std_ReturnType EthIf_Retransmit (uint8 CtrlIdx , uint8 OrigCtrlIdx , Eth_BufIdxType * BufIdxPtr , Eth_DataType * DataPtr , Eth_FrameType FrameType , uint16 LenByte , const Eth_RetransmitInfoType *const RetransmitInfoPtr);	
Parameters (in)	CtrlIdx	Index of the controller where frame shall be retransmitted 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, the priority needs to be set to 0xFF.
Parameters (in,out)	BufIdxPtr	Pointer to the index of the buffer to transmit.
Return Value	Std_ReturnType	
	E_OK	Retransmission of the buffer was successful.
Description	The provided CtrlIdx values refer to non-compatible EthIf controllers (e.g. origin EthIf controller uses VLAN and EthIf controller uses no VLAN or vice versa).	
	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 a buffer index (parameter BufIdxPtr).	

5.3.3.2.33. EthIf_RxIndication

Purpose	Rx indication function.
Synopsis	<pre>void EthIf_RxIndication (uint8 CtrlIdx , Eth_FrameType FrameType , boolean isBroadcast , const uint8 * PhysAddrPtr , Eth_DataType * DataPtr , uint16 LenByte);</pre>
Service ID	0x10
Sync/Async	Synchronous



Reentrancy	Non reentrant	
Parameters (in)	CtrlIdx	Index of the controller within the context of the EthIf.
	FrameType	Ethernet frame type.
	isBroadcast	Indicates whether the 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.
Description	This service is called by the Ethernet driver if a reception is indicated.	

5.3.3.2.34. EthIf_SetControllerMode

Purpose	Sets the Ethernet controller mode.	
Synopsis	Std_ReturnType EthIf_SetControllerMode (uint8 CtrlIdx , Eth_ModeType CtrlMode);	
Service ID	0x03	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	CtrlIdx	Index of the controller within the context of the EthIf.
	CtrlMode	Mode to set the controller to.
Return Value	Std_ReturnType	
	E_OK	Service execution was successful.
	E_NOT_OK	Service execution failed.
Description	This service sets the Ethernet controller mode.	

5.3.3.2.35. EthIf_SetCorrectionTime

Purpose	Service to perform the time correction time stamp.
Synopsis	void EthIf_SetCorrectionTime (uint8 CtrlIdx , const Eth_TimeIntDiffType * timeOffsetPtr , const Eth_RateRatioType * rateRatioPtr);



Service ID	0x26	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	CtrlIdx	EthIf controller index.
	timeOffsetPtr	Pointer to adjust the hardware time.
	rateRatioPtr	Pointer to set the rate ratio.

5.3.3.2.36. EthIf_SetForwardingMode

Purpose	Sets the frame forwarding mode.	
Synopsis	Std_ReturnType EthIf_SetForwardingMode (uint8 SwitchIdx , boolean mode);	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch driver.
	mode	Frame forwarding mode: ▶ TRUE: Forwarding enabled. ▶ FALSE: Forwarding disabled.
Return Value	Std_ReturnType	
	E_OK	Stopping of frame forwarding succeeded.
	E_NOT_OK	Stopping of frame forwarding did not succeed.
Description	This service sets the frame forwarding mode.	

5.3.3.2.37. EthIf_SetGlobalTime

Purpose	Service to set the global time.	
Synopsis	Std_ReturnType EthIf_SetGlobalTime (uint8 CtrlIdx , const Eth_TimeStampType * timeStampPtr);	
Service ID	0x27	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	CtrlIdx	EthIf controller index.



	timeStampPtr	Pointer containing the new time.
Return Value		

5.3.3.2.38. EthIf_SetPhysAddr

Purpose	Set the physical source address of the controller.	
Synopsis	<code>void EthIf_SetPhysAddr (uint8 CtrlIdx , const uint8 * PhysAddrPtr);</code>	
Service ID	0x80	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	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.
Description	This service sets the physical source address used by the indexed controller.	

5.3.3.2.39. EthIf_SetTransceiverWakeupMode

Purpose	Sets the transceiver wake-up mode.	
Synopsis	<code>Std_ReturnType EthIf_SetTransceiverWakeupMode (uint8 TrcvIdx , EthTrcv_WakeupModeType TrcvWakeUpMode);</code>	
Parameters (in)	TrcvIdx	Index of the transceiver within the context of the Ethernet Interface.
	TrcvWakeUpMode	Transceiver wake-up mode: <ul style="list-style-type: none">▶ ETHTRCV_WUM_DISABLE: Disable the transceiver wake-up.▶ ETHTRCV_WUM_ENABLE: Enable the transceiver wake-up.▶ ETHTRCV_WUM_CLEAR: Clears the transceiver wake-up reason.
Return Value	<code>Std_ReturnType</code>	
	E_OK	The request was successful.



	E_NOT_OK	The transceiver wake-up could not be changed or the wake-up reason could not be cleared.
Description	This service enables/disables the wake-up mode or clears the wake-up reason of the specified transceiver.	

5.3.3.2.40. EthIf_StartAllPorts

Purpose	Requests to start all configured ports.	
Synopsis	Std_ReturnType EthIf_StartAllPorts (void);	
Return Value	Std_ReturnType	
	E_OK	Success.
	E_NOT_OK	Port mode could not be started.
Description	This service requests to start all configured ports.	

5.3.3.2.41. EthIf_StoreConfiguration

Purpose	Service to store the configuration of the learned MAC/port tables of a switch in a persistent manner. It is used e.g. by a CDD.	
Synopsis	Std_ReturnType EthIf_StoreConfiguration (uint8 SwitchIdx);	
Service ID	0x2C	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch driver.
Return Value		

5.3.3.2.42. EthIf_SwitchMgmtInfoIndication

Purpose	Callback function to indicate switch management info.
Synopsis	void EthIf_SwitchMgmtInfoIndication (uint8 CtrlIdx , Eth_DataType * DataPtr , EthSwt_MgmtInfoType * MgmtInfoPtr);
Service ID	0x3A
Sync/Async	Synchronous



Reentrancy	Non reentrant	
Parameters (in)	CtrlIdx	Index of an Ethernet Interface controller.
	DataPtr	Ethernet data pointer where the management information belongs.
Parameters (out)	MgmtInfoPtr	Pointer to management information.
Description	Ingress Switch management info indication redirected call to upper layers that registered for the call.	

5.3.3.2.43. EthIf_SwitchPortGroupRequestMode

Purpose	Requests the mode of EthIfSwtPortGroup.	
Synopsis	Std_ReturnType EthIf_SwitchPortGroupRequestMode (EthIf_SwitchPortGroupIdxType PortGroupIdx , EthTrcv_ModeType PortMode);	
Parameters (in)	PortGroupIdx	Index of the port group within the context of the Ethernet Interface.
	PortMode	<p>Requested port mode:</p> <ul style="list-style-type: none"> ▶ ETHTRCV_MODE_DOWN: Disables the port group. ▶ ETHTRCV_MODE_ACTIVE: Enables the port group.
Return Value	Std_ReturnType	
	E_OK	Success.
	E_NOT_OK	Port group mode could not be changed.
Description	This service requests 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

Purpose	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 <ul style="list-style-type: none"> ▶ dropped packets due to a buffer overrun, ▶ dropped packets due to CRC errors, etc.
----------------	--



Synopsis	<code>Std_ReturnType EthIf_SwtGetCounterValues (uint8 SwitchIdx , uint8 SwitchPortIdx , Eth_CounterType * CounterPtr);</code>	
Service ID	0x40	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch driver.
Return Value		

5.3.3.2.45. EthIf_Transmit

Purpose	Transmit a Tx buffer on Ethernet.	
Synopsis	<code>Std_ReturnType EthIf_Transmit (uint8 CtrlIdx , Eth_BufIdxType BufIdx , Eth_FrameType FrameType , boolean TxConfirmation , uint16 LenByte , uint8 * PhysAddrPtr);</code>	
Service ID	0x0a	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	CtrlIdx	Index of the controller within the context of the EthIf.
	BufIdx	Index of the buffer to transmit.
	FrameType	Ethernet type of the related payload for the transmission.
	TxConfirmation	Activates a transmission confirmation.
	LenByte	Length of the payload contained in the Tx buffer to transmit.
	PhysAddrPtr	Address that holds the destination address the data shall be transmitted to.
Return Value		
Description	This service unlocks a previously provided Tx buffer for transmission.	

5.3.3.2.46. EthIf_TrcvModeIndication

Purpose	Callback function to indicate a transceiver mode change.
Synopsis	<code>void EthIf_TrcvModeIndication (uint8 TrcvIdx , EthTrcv_ModeType TrcvMode);</code>
Service ID	0x0F



Sync/Async	Synchronous	
Reentrancy	Non reentrant for the same CtrlIdx. Reentrant for different CtrlIdx.	
Parameters (in)	TrcvIdx	Index of the Ethernet transceiver within the context of the Ethernet Interface
	TrcvMode	Notified Ethernet transceiver mode: ▶ ETHTRCV_MODE_DOWN ▶ ETHTRCV_MODE_ACTIVE
Description	Called asynchronously when the mode is read out. Triggered by the previous Eth_SetTransceiverMode() call. Can be called directly within the trigger functions.	

5.3.3.2.47. EthIf_TxConfirmation

Purpose	Tx confirmation function.	
Synopsis	<pre>void EthIf_TxConfirmation (uint8 CtrlIdx , Eth_BufIdxType BufIdx , Std_ReturnType Result);</pre>	
Service ID	0x11	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	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.
Description	This service confirms the frame transmission by the indexed controller.	

5.3.3.2.48. EthIf_UpdatePhysAddrFilter

Purpose	Updates the physical address filter.
Synopsis	<pre>Std_ReturnType EthIf_UpdatePhysAddrFilter (uint8 CtrlIdx , const uint8 * PhysAddrPtr , Eth_FilterActionType Action);</pre>
Service ID	0x81
Sync/Async	Synchronous
Reentrancy	Reentrant



Parameters (in)	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.
Return Value	Std_ReturnType	
	E_OK	Filter was successfully changed.
	E_NOT_OK	Filter could not be changed.
Description	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

Purpose	Verifies the switch configuration.	
Synopsis	Std_ReturnType EthIf_VerifyConfig (uint8 SwitchIdx , boolean * Result);	
Parameters (in)	SwitchIdx	Index of the switch within the context of the Ethernet Switch driver.
Parameters (out)	Result	Pointer to the result of verification: ▶ TRUE: Configuration verified ok. ▶ FALSE: Configuration values found corrupted.
Return Value	Std_ReturnType	
	E_OK	Configuration verification succeeded.
	E_NOT_OK	Configuration verification did not succeed.
Description	This service verifies the switch configuration depending on the hardware architecture, hardware capability, and the intended accuracy of this verification. Forwarded to EthSwt_VerifyConfig().	

5.3.3.2.50. EthIf_WriteMii

Purpose	Service to write into an Ethernet transceiver register.
----------------	---



Synopsis	<code>Std_ReturnType EthIf_WriteMii (uint8 CtrlIdx , uint8 RegIdx , uint16 RegVal);</code>	
Service ID	0x71	
Parameters (in)	CtrlIdx	EthIf controller index.
	RegIdx	Register index of the TrcvIdx assigned to the CtrlIdx.
	RegVal	Value that shall be written to the RegIdx.
Return Value		

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

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.

5.3.4.2. 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

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

Description	{code=[EthIf, EthIf_MainFunctionRx()]}
-------------	--

5.3.4.4.2. lim.EthIf.EB_INTREQ_EthIf_0002

Description	{code=EthIf_Init()}
Rationale	The reinitialization process resets 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

Description	{code=[EthIf_GetControllerMode(), Eth_GetControllerMode()]}
Rationale	EthIf_GetControllerMode()



5.3.4.4.4. lim.EthIf.EB_INTREQ_EthIf_0004

Description	{code=[EthIf_Init(), EthIf_Init()]}
Rationale	EthIf_Init() EthIf_Init()

5.3.4.4.5. lim.EthIf.EB_INTREQ_EthIf_0005

Description	{code=[EthIf_MainFunctionRx_PrioProcessing[name](), EthIf_MainFunctionRx()]}
-------------	--

5.3.4.4.6. lim.EthIf.EB_INTREQ_EthIf_0006

Description	The config parameter EthIfInitControllersTransceivers shall be enabled only for Eth drivers and transceivers below AUTOSAR 4.2.2. The functions Eth_ControllerInit() and EthTrcv_TransceiverInit() shall be called by integration code for EB Eth drivers and transceivers for AUTOSAR 4.2.2 or higher.
-------------	---

5.3.4.4.7. lim.EthIf.EB_INTREQ_EthIf_0007

Description	If EB Eth drivers and transceivers for AUTOSAR 4.3.0 are used, then the config parameter EthIfSupportEthAPI needs to be set to ASR430_EB. A compilation error stating a syntax issue with EthIf_TxConfirmation() is an indication that the EthIfSupportEthAPI is not properly configured.
-------------	---

5.3.4.4.8. lim.EthIf.EB_INTREQ_EthIf_0008

Description	If EthIfSwitch and/or EthIfSwitchPortGroup elements are configured, then the EthTrcv configuration parameter EthTrcvGeneral/EthTrcvGetLinkStateApi must be set and at least one of EthSwtPorts, configured in EthSwt/EthSwtConfig/*/EthSwtPort must have an EthSwtPortTrcvRef defined.
-------------	--

5.3.4.4.9. lim.EthIf.EB_INTREQ_EthIf_0009

Description	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. the same SPI. Preemption of an SPI cannot 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 the SPI. It shall be ensured that the 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 the EthIf configuration.

5.3.4.4.10. lim.EthIf.EB_INTREQ_EthIf_0010

Description	If the config parameter EthIfRetransmitApiEnable is enabled, the DataPtr parameter used in the EthIf_Retransmit() API shall provide space for a VLAN header in front of the used buffer.
--------------------	--

5.3.4.4.11. lim.EthIf.EB_INTREQ_EthIf_0011

Description	If Eth drivers of a version below AUTOSAR 4.3.0 are used, then EthCtrlIdx of all referenced EthCtrl must be 0-based and consecutive.
--------------------	--

5.3.4.4.12. lim.EthIf.EB_INTREQ_EthIf_0012

Description	If support of drivers that contain vendor ID and vendorApiInfix is required, then every driver that requires it (Eth, EthTrcv, or EthSwt) needs to have a valid BswModuleDescription which shall be referenced from EthIf to provide it with vendor ID and vendorApiInfix. Support is not available for AUTOSAR versions below 4.3.0.
--------------------	---

5.3.4.4.13. lim.EthIf.EB_INTREQ_EthIf_0013

Description	{}
--------------------	----

5.3.4.4.14. lim.EthIf.EB_INTREQ_EthIf_0014

Description	If support of multiple drivers is required, then all the drivers must have an unique combination of VendorId and/or VendorApiInfix defined in driver's BswModuleDescription file.
--------------------	---

5.3.4.4.15. lim.EthIf.EB_INTREQ_EthIf_0015

Description	EthIf requires some optional API of Eth, EthTrcv and EthSwt to be configured. If EthTrcv is present in the configuration following APIs need to be enabled: EthTr-
--------------------	--



	cv_SetTransceiverMode() EthTrcv_GetTransceiverMode() EthTrcv_GetLinkState() EthTrcv_SetPhyTestMode() EthTrcv_GetCableDiagnosticsResult() EthTrcv_GetPhyIdentifier() If EthSwt is present in the configuration: EthSwt_SetSwitchPortMode() and EthSwt_GetSwitchPortMode() need to be enabled if port management is expected. This list may not be exhaustive.
Rationale	Although lower layer APIs are defined as optional in the AUTOSAR specification, the absence of them may cause serious and unwanted side effects in EthIf core functionalities.

5.4. EthSM

5.4.1. Configuration parameters

Containers included		
Container name	Multiplicity	Description
CommonPublishedInformation	1..1	Label: Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
EthSMConfig	1..1	This container is used to provide a name to the configuration of the AUTOSAR EthSM module.
EthSMGeneral	1..1	This container contains the global parameter of the Ethernet State Manager.
ReportToDem	1..1	Label: Production error handling Production error handling
EthSMDefensiveProgramming	1..1	Label: Defensive Programming Options Parameters for defensive programming
EthSMNetwork	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.
PublishedInformation	1..1	Label: EB Published Information Additional published parameters not covered by CommonPublishedInformation container.



Parameters included

Parameter name	Multiplicity
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT
Label	Config Variant
Multiplicity	1..1
Type	ENUMERATION
Default value	VariantPostBuild
Range	VariantPostBuild

5.4.1.1. CommonPublishedInformation

Parameters included

Parameter name	Multiplicity
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1
SwPatchVersion	1..1
ModuleId	1..1
VendorId	1..1
Release	1..1

Parameter Name	ArMajorVersion
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	ArMinorVersion
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	3
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ArPatchVersion
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	0
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMajorVersion
Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMinorVersion
Label	Software Minor Version
Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.

Multiplicity	1..1
Type	INTEGER_LABEL
Default value	6
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwPatchVersion
Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	16
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ModuleId
Label	Numeric Module ID
Description	Module ID of this module from Module List
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	143
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	VendorId
Label	Vendor ID
Description	Vendor ID of the dedicated implementation of this module according to the AU-TOSAR vendor list
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH



Parameter Name	Release
Label	Release Information
Multiplicity	1..1
Type	STRING_LABEL
Default value	
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

5.4.1.2. EthSMConfig

5.4.1.3. EthSMGeneral

Parameters included	
Parameter name	Multiplicity
EthSMDevErrorDetect	1..1
EthSMDummyMode	1..1
EthSMMainFunctionPeriod	1..1
EthSMVersionInfoApi	1..1
EthSMSingleNetworkOptEnable	1..1
EthSMMaxNetworks	1..1
EthSMMultiCoreSupport	1..1
EthSMDevAuthSupport	1..1
EthSMRelocatablePbcfgEnable	1..1

Parameter Name	EthSMDevErrorDetect	
Description	Enables and disables the development error detection and notification mechanism.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	AUTOSAR_ECUC
---------------	--------------

Parameter Name	EthSMDummyMode	
Description	This configuration parameter is not used. Feature is not supported. Disables the API to the Ethlf. 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.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthSMMainFunctionPeriod	
Description	Specifies the period in seconds that the MainFunction has to be triggered with.	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.01	
Range	<=1 >=0.005	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthSMVersionInfoApi	
Description	Enables and disables the version info API.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthSMSingleNetworkOptEnable	
Description	Optimizes code for the purpose of using a single Ethernet network only.	



	<ul style="list-style-type: none"> ▶ True: The EthSM supports a single Ethernet network only. ▶ False: The EthSM supports multiple Ethernet networks. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (code): Enabling this parameter reduces the ROM consumption of the module code. ▶ Execution time reduction (code): Enabling this parameter reduces the execution time of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthSMMMaxNetworks
Description	Number of Ethernet networks supported by the EthSM.
Multiplicity	1..1
Type	INTEGER
Default value	1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthSMMMultiCoreSupport
Description	Enables module Multi core support.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthSMDevAuthSupport
Description	Enables module Device Authentication support.
Multiplicity	1..1
Type	BOOLEAN



Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthSMRelocatablePbcfgEnable
Description	Enables/disable support for relocatable postbuild configuration. <ul style="list-style-type: none"> ▶ True: Postbuild configuration relocatable in memory. ▶ False: Postbuild configuration not relocatable in memory.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.4.1.4. ReportToDem

Parameters included	
Parameter name	Multiplicity
EthSMDemCtrlTestResultReportToDem	1..1
EthSMDemCtrlTestResultReportToDemDetErrorId	1..1

Parameter Name	EthSMDemCtrlTestResultReportToDem
Label	EthSMDemCtrlTestResult report to
Description	Selects the handling of the production error EthSMDemCtrlTestResult. <ul style="list-style-type: none"> ▶ DEM: The error is reported to the Diagnostics Event Manager (Dem). ▶ DET: The error is reported to the Development Error Tracer (Det) if enabled. ▶ DISABLE: The error is not reported at all. Optimization Effect: <ul style="list-style-type: none"> ▶ ROM reduction (code): Setting this parameter to a value of DISABLE reduces the ROM consumption of the module code. ▶ Execution time reduction (code): Setting this parameter to a value of DISABLE reduces the execution time of the module code.



Multiplicity	1..1
Type	ENUMERATION
Default value	DISABLE
Range	DEM
	DET
	DISABLE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthSMDemCtrlTestResultReportToDemDetErrorId	
Label	EthSMDemCtrlTestResult Dem To Det error ID	
Description	If a production error is reported towards the Det, EthSMDemCtrlTestResultReportToDemDetErrorId defines the error ID which is reported towards the Det.	
Multiplicity	1..1	
Type	INTEGER	
Default value	129	
Range	<=255	
	>=0	
Configuration class	PreCompile:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.4.1.5. EthSMDefensiveProgramming

Parameters included	
Parameter name	Multiplicity
EthSMDefProgEnabled	1..1
EthSMPrecondAssertEnabled	1..1
EthSMPostcondAssertEnabled	1..1
EthSMStaticAssertEnabled	1..1
EthSMUnreachAssertEnabled	1..1
EthSMInvariantAssertEnabled	1..1

Parameter Name	EthSMDefProgEnabled



Label	Enable Defensive Programming
Description	<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"> 1. Enable development error detection 2. Enable defensive programming 3. Enable assertions as required
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthSMPrecondAssertEnabled
Label	Enable Precondition Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>EthSMDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>EthSMDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthSMPostcondAssertEnabled
Label	Enable Postcondition Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>EthSMDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>EthSMDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthSMStaticAssertEnabled
Label	Enable Static Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>EthSMDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>EthSMDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	EthSMUnreachAssertEnabled
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"> ▶ Enable Development Error Detection (<code>EthSMDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>EthSMDefProgEnabled</code>): must be enabled
Multiplicity	1..1



Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	EthSMInvariantAssertEnabled	
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"> ▶ Enable Development Error Detection (<code>EthSMDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>EthSMDefProgEnabled</code>): must be enabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.4.1.6. EthSMNetwork

Containers included		
Container name	Multiplicity	Description
EthSMDemEventParameter- Refs	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.

Parameters included	
Parameter name	Multiplicity



Parameters included

EthSMEthIfControllerRef	1..1
EthSMComMNetworkHandleRef	1..1
EthSMDevAuthNoComNotificationEnable	1..1

Parameter Name	EthSMEthIfControllerRef	
Description	ID of the Ethernet controller assigned to the configured network handle. Reference to one of the controllers of EthIf configuration.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthSMComMNetworkHandleRef	
Description	Unique handle to identify one certain Ethernet network. Reference to one of the network handles configured for the ComM.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	EthSMDevAuthNoComNotificationEnable	
Description	Enables DevAuth support for this EthSMNetwork.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.4.1.7. EthSMDemEventParameterRefs

Parameters included	
Parameter name	Multiplicity



Parameters included

ETHSM_E_LINK_DOWN	0..1
-----------------------------------	------

Parameter Name	ETHSM_E_LINK_DOWN
Description	<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"> ▶ <code>EthSMDemCtrlTestResultReportToDem</code>: Select DEM to enable the reporting of <code>ETHSM_E_LINK_DOWN</code>. <p>Further notes:</p> <ul style="list-style-type: none"> ▶ Activation: This error is reported when the transceiver switches to "down" while communication has already been established and is requested because of communication request. ▶ 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>. ▶ Trigger debounce: None. The error is reported on first occurrence. ▶ Rate of diagnostic checks: Checked on every <code>EthSM_Mainfunction()</code> call.
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.4.1.8. PublishedInformation

Parameters included	
Parameter name	Multiplicity
PbcfgMSupport	1..1

Parameter Name	PbcfgMSupport
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	PublishedInformation:
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_CtrlModeIndication().
Value	0x09U

5.4.2.1.2. ETHSM_DEVAUTHMODEINDICATION_SVCID

Purpose	Defines API service ID of function EthSM_DevAuthNoComModeIndication().
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 EthSM_GetCurrentComMode() .
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 EthSM_GetVersionInfo() .
Value	0x02U

5.4.2.1.15. ETHSM_INIT_SVCID

Purpose	Defines API service ID of function EthSM_Init() .
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.
----------------	--



Value	0U
--------------	----

5.4.2.1.17. ETHSM_MAINFUNCTION_SVCID

Purpose	Defines API service ID of function EthSM_MainFunction() .
Value	0x01U

5.4.2.1.18. ETHSM_REQUESTCOMMODE_SVCID

Purpose	Defines API service ID of function EthSM_RequestComMode() .
Value	0x05U

5.4.2.1.19. ETHSM_TCPIPMODEINDICATION_SVCID

Purpose	Defines API service ID of function EthSM_TcpIpModeIndication() .
Value	0x08U

5.4.2.1.20. ETHSM_TRCVLINKSTATECHG_SVCID

Purpose	Defines API service ID of function EthSM_TrcvLinkStateChg().
Value	0x06U

5.4.2.1.21. ETHSM_TRCVMODEINDICATION_SVCID

Purpose	Defines API service ID of function EthSM_TrcvModeIndication().
Value	0x10U

5.4.2.2. Functions

5.4.2.2.1. EthSM_GetCurrentComMode

Purpose	Gets the current communication mode.
----------------	--------------------------------------



Synopsis	Std_ReturnType EthSM_GetCurrentComMode (NetworkHandleType NetworkHandle , ComM_ModeType * ComM_ModePtr);	
Service ID	0x04	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	NetworkHandle	- Handle of the affected network
Parameters (out)	ComM_ModePtr	- Address to write the current Com mode to.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service gets the current communication mode of the network.	

5.4.2.2.2. EthSM_GetVersionInfo

Purpose	Get version information of the Ethernet State Manager.	
Synopsis	void EthSM_GetVersionInfo (Std_VersionInfoType * versioninfo);	
Service ID	0x02	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	versioninfo	- Pointer where to store the version information of this module.
Description	<p>This service returns the version information of this module. The version information includes:</p> <ul style="list-style-type: none"> ▶ Module Id ▶ Vendor Id ▶ Vendor specific version numbers 	

5.4.2.2.3. EthSM_Init

Purpose	EthSM_Init() initializes the EthSM module.
Synopsis	Std_ReturnType EthSM_Init (const EthSM_ConfigType * ConfigPtr);



Service ID	0x07	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ConfigPtr	- Address of the post-build configuration data structure.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	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

Purpose	Checks compatibility of the post-build-time configuration.	
Synopsis	Std_ReturnType EthSM_IsValidConfig (const void * voidConfigPtr) ;	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	voidConfigPtr	Pointer to the configuration data of the EthSM module.
Return Value	Result of compatibility check	
	E_OK	Provided configuration is compatible.
	E_NOT_OK	Provided configuration is not compatible.
Description	This service checks the compatibility of the post-build-time configuration against the source code.	

5.4.2.2.5. EthSM_MainFunction

Purpose	EthSM main function.	
Synopsis	void EthSM_MainFunction (void) ;	
Service ID	0x01	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Description	This service operate the effects of the EthSM state machine.	



5.4.2.2.6. EthSM_RequestComMode

Purpose	Service to set the requested Com mode.	
Synopsis	Std_ReturnType EthSM_RequestComMode (NetworkHandleType NetworkHandle , ComM_ModeType ComM_Mode);	
Service ID	0x05	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	NetworkHandle	- Handle of the affected network.
	ComM_Mode	- Com mode requested.
Return Value	E_OK: Service execution was successful. E_NOT_OK: Service execution failed.	
Description	This service sets the Com mode requested by the upper layer.	

5.4.2.2.7. EthSM_TcpIpModeIndication

Purpose	TcpIp mode indication.	
Synopsis	void EthSM_TcpIpModeIndication (uint8 CtrlIdx , TcpIp_StateType TcpIpState);	
Service ID	0x08	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CtrlIdx	- Controller index which changed the TcpIp state.
	TcpIpState	- New state the TcpIp has changed to.
Description	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

Protected data structures	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.
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. 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

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

Description	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.
Rationale	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
CommonPublishedInformation	1..1	Label: Common Published Information

**Containers included**

		Common container, aggregated by all modules. It contains published information about vendor and versions.
PublishedInformation	1..1	Label: EB Published Information Additional published parameters not covered by Common-PublishedInformation container.
SdConfig	1..n	This container contains the configuration parameters and sub containers of the AUTOSAR Service Discovery module.
SdGeneral	1..1	This container lists the general configuration parameters for the Service Discovery module.
SdDefensiveProgramming	1..1	Label: Defensive Programming Options Parameters for defensive programming

Parameters included

Parameter name	Multiplicity
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT	
Label	Config Variant	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	VariantPostBuild	
Range	VariantPostBuild	
Configuration class	VariantPostBuild:	VariantPostBuild

5.5.1.1. CommonPublishedInformation**Parameters included**

Parameter name	Multiplicity
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1



Parameters included

SwPatchVersion	1..1
ModuleId	1..1
VendorId	1..1
Release	1..1

Parameter Name	ArMajorVersion
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	ArMinorVersion
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	2
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ArPatchVersion
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:



Origin	Elektrobit Automotive GmbH
---------------	----------------------------

Parameter Name	SwMajorVersion
Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMinorVersion
Label	Software Minor Version
Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	4
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwPatchVersion
Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	16
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ModuleId
Label	Numeric Module ID
Description	Module ID of this module from Module List

Multiplicity	1..1
Type	INTEGER_LABEL
Default value	171
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	VendorId
Label	Vendor ID
Description	Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	Release
Label	Release Information
Multiplicity	1..1
Type	STRING_LABEL
Default value	
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

5.5.1.2. PublishedInformation

Parameters included	
Parameter name	Multiplicity
PbcfgMSupport	1..1

Parameter Name	PbcfgMSupport
Label	PbcfgM support
Description	Specifies whether or not the Sd can use the PbcfgM module for post-build support.



Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

5.5.1.3. SdConfig

Containers included		
Container name	Multiplicity	Description
SdInstance	0..n	This container represents an instance of the SD; i.e. the SD configuration for a certain link.

5.5.1.4. SdInstance

Containers included		
Container name	Multiplicity	Description
SdClientService	0..n	This container specifies all parameters used by Client services.
SdClientTimer	0..n	This container specifies all timers used by the Service Discovery module for Client Services.
SdInstanceDemEventParameterRefs	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.
SdInstanceMulticastRxPdu	1..1	This container specifies the received PDU.
SdInstanceTxPdu	1..1	This container specifies the transmitted PDU.
SdInstanceUnicastRxPdu	1..1	This container specifies the received PDU.
SdServerService	0..n	This container specifies all parameters used by Server services.
SdServerTimer	0..n	This container specifies all timers used by the Service Discovery module for Server Services.



Parameters included

Parameter name	Multiplicity
SdInstanceHostname	0..1
SdMaximumRemoteNodes	1..1
SdSubscribeEventgroupRetryMax	1..1
SdSubscribeEventgroupRetryDelay	0..1

Parameter Name	SdInstanceHostname	
Description	Configuration parameter to specify the Hostname.	
Multiplicity	0..1	
Type	STRING	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdMaximumRemoteNodes	
Description	Maximum number of remote nodes supported by this SdInstance.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65533 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	SdSubscribeEventgroupRetryMax	
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	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdSubscribeEventgroupRetryDelay	
----------------	--	--



Description	Time in seconds when a subscription to an event group shall be retriggered, if no SubscribeEventGroupAck or SubscribeEventGroupNack was received.	
Multiplicity	0..1	
Type	FLOAT	
Default value	0.01	
Range	<=50 >=0.001	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

5.5.1.5. SdClientService

Containers included		
Container name	Multiplicity	Description
SdClientCapabilityRecord	0..n	Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service.
SdConsumedEventGroup	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.
SdBlacklistedVersions	0..n	
SdConsumedMethods	0..1	Container element for representing the data path for accessing the server methods.

Parameters included	
Parameter name	Multiplicity
SdClientServiceAutoRequire	1..1
SdClientServiceHandleId	1..1
SdClientServiceId	1..1
SdClientServiceInstanceId	1..1
SdClientServiceMajorVersion	1..1
SdClientServiceMinorVersion	1..1
SdVersionDrivenFindBehavior	1..1



Parameters included

SdClientServiceTcpRef	0..1
SdClientServiceTimerRef	1..1
SdClientServiceUdpRef	0..1

Parameter Name	SdClientServiceAutoRequire	
Description	If existing and set to true, this Service will be set to "required" on start.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdClientServiceHandleId	
Description	The HandleId by which the BswM can identify this Client Service Instance.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=65535 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdClientServiceId	
Description	Id to identify the service. This is unique for the service interface.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=65534 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdClientServiceInstancId	
-----------------------	---------------------------------	--



Description	Configuration parameter to specify Instance Id of the service as used in SD entries.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65534 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdClientServiceMajorVersion	
Description	Major version number of the Service as used in the SD entries.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=254 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdClientServiceMinorVersion	
Description	Minor version number of the Service as used in the SD Service Entries.	
Multiplicity	1..1	
Type	INTEGER	
Default value	4294967295	
Range	<=4294967295 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdVersionDrivenFindBehavior	
Description	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"> ▶ 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.
Multiplicity	1..1
Type	ENUMERATION
Default value	EXACT_OR_ANY_MINOR_VERSION
Range	EXACT_OR_ANY_MINOR_VERSION MINIMUM_MINOR_VERSION
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SdClientServiceTcpRef
Description	Reference to the SoAdSocketConnection representing the data path (TCP) for communication with methods.
Multiplicity	0..1
Type	REFERENCE
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SdClientServiceTimerRef
Description	The reference of the SdClientTimer container for this service.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SdClientServiceUdpRef
Description	Reference to the SoAdSocketConnection representing the data path (UDP) for communication with methods.
Multiplicity	0..1
Type	REFERENCE
Configuration class	PostBuild: VariantPostBuild



Origin	AUTOSAR_ECUC
---------------	--------------

5.5.1.6. SdClientCapabilityRecord

Parameters included	
Parameter name	Multiplicity
SdClientServiceCapabilityRecordKey	1..1
SdClientServiceCapabilityRecordValue	0..1

Parameter Name	SdClientServiceCapabilityRecordKey
Description	Defines a CapabilityRecord key.
Multiplicity	1..1
Type	STRING
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SdClientServiceCapabilityRecordValue
Description	Defines the corresponding CapabilityRecord value.
Multiplicity	0..1
Type	STRING
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.5.1.7. SdConsumedEventGroup

Containers included		
Container name	Multiplicity	Description
SdClientCapabilityRecord	0..n	Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service.

Parameters included	
Parameter name	Multiplicity
SdConsumedEventGroupAutoRequire	1..1



Parameters included

SdConsumedEventGroupHandleId	1..1
SdConsumedEventGroupId	1..1
SdConsumedEventGroupMulticastGroupRef	0..n
SdConsumedEventGroupTimerRef	1..1
SdConsumedEventGroupMulticastActivationRef	0..1
SdConsumedEventGroupTcpActivationRef	0..1
SdConsumedEventGroupUdpActivationRef	0..1

Parameter Name	SdConsumedEventGroupAutoRequire	
Description	If existing and set to true, this EventGroup will be set to "required" on start.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdConsumedEventGroupHandleId	
Description	The HandleId by which the BswM can identify this EventGroup.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=65535 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdConsumedEventGroupId	
Description	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.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=65534	



	>=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC
Parameter Name	SdConsumedEventGroupMulticastGroupRef
Description	Reference to the SoAdSocketConnectionGroup representing the multicast data path (UDP).
Multiplicity	0..n
Type	REFERENCE
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC
Parameter Name	SdConsumedEventGroupTimerRef
Description	The reference of the SdClientTimer container for this eventGroup.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC
Parameter Name	SdConsumedEventGroupMulticastActivationRef
Description	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.
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC
Parameter Name	SdConsumedEventGroupTcpActivationRef
Description	The reference of the Routing Group for activation of the data path for receiving TCP events.
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC



Parameter Name	SdConsumedEventGroupUdpActivationRef	
Description	The reference of the Routing Group for activation of the data path for receiving UDP events.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.5.1.8. SdClientCapabilityRecord

Parameters included		
Parameter name		Multiplicity
SdClientServiceCapabilityRecordKey		1..1
SdClientServiceCapabilityRecordValue		0..1

Parameter Name	SdClientServiceCapabilityRecordKey	
Description	Defines a CapabilityRecord key.	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdClientServiceCapabilityRecordValue	
Description	Defines the corresponding CapabilityRecord value.	
Multiplicity	0..1	
Type	STRING	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.5.1.9. SdBlacklistedVersions

Parameters included		
Parameter name		Multiplicity



Parameters included

SdBlacklistedMinorVersion	1..1
---	------

Parameter Name	SdBlacklistedMinorVersion	
Description	Blacklisted MinorVersions.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=4294967294 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.5.1.10. SdConsumedMethods

Parameters included

Parameter name	Multiplicity
SdClientServiceActivationRef	1..1

Parameter Name	SdClientServiceActivationRef	
Description	Reference to a SoAdRoutingGroupRef to activate/deactivate the data path for the methods.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.5.1.11. SdClientTimer

Parameters included

Parameter name	Multiplicity
SdClientTimerInitialFindDelayMax	0..1
SdClientTimerInitialFindDelayMin	0..1
SdClientTimerInitialFindRepetitionsBaseDelay	0..1



Parameters included

SdClientTimerInitialFindRepetitionsMax	0..1
SdClientTimerRequestResponseMaxDelay	0..1
SdClientTimerRequestResponseMinDelay	0..1
SdClientTimerTTL	1..1

Parameter Name	SdClientTimerInitialFindDelayMax	
Description	Max value in [s] to delay randomly the transmission of a find message. This parameter is mandatory for ClientService.	
Multiplicity	0..1	
Type	FLOAT	
Default value	0.0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdClientTimerInitialFindDelayMin	
Description	Min value in [s] to delay randomly the transmission of a find message. This parameter is mandatory for ClientService.	
Multiplicity	0..1	
Type	FLOAT	
Default value	0.0	
Range	<=4294967294 >=0.0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdClientTimerInitialFindRepetitionsBaseDelay	
Description	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.	
Multiplicity	0..1	
Type	FLOAT	
Default value	0.01	
Range	<=Infinity	



	>=0.0
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SdClientTimerInitialFindRepetitionsMax	
Description	Configuration for the maximum number of find repetitions. This parameter is mandatory for ClientService.	
Multiplicity	0..1	
Type	INTEGER	
Default value	3	
Range	<=10 >=0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdClientTimerRequestResponseMaxDelay	
Description	Maximum allowable response delay to entries received by multicast in seconds. This parameter is mandatory for ConsumedEventGroups.	
Multiplicity	0..1	
Type	FLOAT	
Default value	0.0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdClientTimerRequestResponseMinDelay	
Description	Minimum allowable response delay to the find message in seconds. This parameter is mandatory for ConsumedEventGroups.	
Multiplicity	0..1	
Type	FLOAT	
Default value	0.0	
Range	<=4294967294 >=0.0	
Configuration class	PostBuild:	VariantPostBuild



Origin	AUTOSAR_ECUC
---------------	--------------

Parameter Name	SdClientTimerTTL	
Description	Time to live for find and subscribe messages.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=16777215 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.5.1.12. SdInstanceDemEventParameterRefs

Parameters included	
Parameter name	Multiplicity
SD_E_MALFORMED_MSG	0..1
SD_E_OUT_OF_RES	0..1
SD_E_SUBSCR_NACK_RECV	0..1
SD_E_SERVER_NOT_AVAILABLE	0..1

Parameter Name	SD_E_MALFORMED_MSG	
Description	Reference to the DemEventParameter which shall be issued when the SD Instance received malformed message.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SD_E_OUT_OF_RES	
Description	Reference to the DemEventParameter which shall be issued when the SD Instance does not have enough resources to handle client.	
Multiplicity	0..1	



Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SD_E_SUBSCR_NACK_RECV	
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	SD_E_SERVER_NOT_AVAILABLE	
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	SdRxPdulId	
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	



Parameter Name	SdRxPduRef	
Description	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.5.1.14. SdInstanceTxPdu

Parameters included		
Parameter name		Multiplicity
SdTxPduRef		1..1

Parameter Name	SdTxPduRef	
Description	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.5.1.15. SdInstanceUnicastRxPdu

Parameters included		
Parameter name		Multiplicity
SdRxPduld		1..1
SdRxPduRef		1..1

Parameter Name	SdRxPduld	
Description	ID of the PDU that will be received via the API Sd_SoAdlfRxIndication().	
Multiplicity	1..1	
Type	INTEGER	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	SdRxPduRef	
Description	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

5.5.1.16. SdServerService

Containers included		
Container name	Multiplicity	Description
SdEventHandler	0..n	Container Element for representing an EventGroup as part of the Service Instance.
SdProvidedMethods	0..1	Container element for representing the needed elements of the data path for the methods provided by the service.
SdServerCapabilityRecord	0..n	Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service.

Parameters included	
Parameter name	Multiplicity
SdServerServiceAutoAvailable	1..1
SdServerServiceHandleId	1..1
SdServerServiceId	1..1
SdServerServiceInstanceId	1..1
SdServerServiceMajorVersion	1..1
SdServerServiceMinorVersion	1..1
SdServerServiceTcpRef	0..1
SdServerServiceTimerRef	1..1
SdServerServiceUdpRef	0..1



Parameter Name	SdServerServiceAutoAvailable	
Description	If existing and set to true, this Service will be set to "Available" on start.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdServerServiceHandleId	
Description	The HandleId by which the BswM can identify this Server Service Instance.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65535 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdServerServiceId	
Description	Id to identify the service. This is unique for the service interface.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=65534 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdServerServiceInstanceld	
Description	Configuration parameter to specify Instance Id of the Service implemented by the Server Service.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65534	



	>=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SdServerServiceMajorVersion	
Description	Major version number of the Service as used in SD Entries.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=254 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdServerServiceMinorVersion	
Description	Minor version number of the Service as used e.g. in Offer Service entries.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=4294967294 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdServerServiceTcpRef	
Description	Reference to SoAdSocketConnectionGroup used for methods.	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdServerServiceTimerRef	
Description	The reference of the SdServerTimer container for this service.	



Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SdServerServiceUdpRef
Description	Reference to SoAdSocketConnectionGroup used for methods.
Multiplicity	0..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.5.1.17. SdEventHandler

Containers included		
Container name	Multiplicity	Description
SdEventHandlerMulticast	0..1	The subcontainer including the Routing Group for Activation of Events sent over Multicast.
SdEventHandlerTcp	0..1	The subcontainer including the Routing Groups for Activation and Trigger Transmit for Events sent over TCP.
SdEventHandlerUdp	0..1	The subcontainer including the Routing Groups for Activation and Trigger Transmit for Events sent over UDP.
SdServerCapabilityRecord	0..n	Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service.

Parameters included	
Parameter name	Multiplicity
SdEventHandlerEventGroupId	1..1
SdEventHandlerHandleId	1..1
SdEventHandlerMulticastThreshold	1..1
SdEventHandlerTimerRef	1..1

Parameter Name	SdEventHandlerEventGroupId
-----------------------	-----------------------------------



Description	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.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65534 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdEventHandlerHandleId	
Description	The HandleId by which the BswM can identify this EventGroup.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=65535 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdEventHandlerMulticastThreshold	
Description	Specifies the number of subscribed clients that trigger the Server to change the transmission of events to Multicast.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdEventHandlerTimerRef	
Description	The reference of the SdServerTimer container for this EventGroup.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	AUTOSAR_ECUC
--------	--------------

5.5.1.18. SdEventHandlerMulticast

Parameters included	
Parameter name	Multiplicity
SdEventActivationRef	0..1
SdMulticastEventSoConRef	0..1

Parameter Name	SdEventActivationRef	
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	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdMulticastEventSoConRef	
Description	Reference to the SoAdSocketConnection representing the multicast data path (UDP).	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.5.1.19. SdEventHandlerTcp

Parameters included	
Parameter name	Multiplicity
SdEventActivationRef	0..1
SdEventTriggeringRef	0..1
Parameter Name	
SdEventActivationRef	



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	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdEventTriggeringRef	
Description	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.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.5.1.20. SdEventHandlerUdp

Parameters included		
Parameter name		Multiplicity
<u>SdEventActivationRef</u>		0..1
<u>SdEventTriggeringRef</u>		0..1

Parameter Name	SdEventActivationRef	
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	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdEventTriggeringRef	
Description	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.	



Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.5.1.21. SdServerCapabilityRecord

Parameters included	
Parameter name	Multiplicity
SdServerCapabilityRecordKey	1..1
SdServerCapabilityRecordValue	0..1

Parameter Name	SdServerCapabilityRecordKey
Description	Defines a CapabilityRecord key.
Multiplicity	1..1
Type	STRING
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SdServerCapabilityRecordValue
Description	Defines the corresponding CapabilityRecord value.
Multiplicity	0..1
Type	STRING
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.5.1.22. SdProvidedMethods

Parameters included	
Parameter name	Multiplicity
SdServerServiceActivationRef	1..1
Parameter Name	SdServerServiceActivationRef



Description	Reference to a SoAdRoutingGroup to activated and deactivate the data path for methods of the service.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.5.1.23. SdServerCapabilityRecord

Parameters included	
Parameter name	Multiplicity
SdServerCapabilityRecordKey	1..1
SdServerCapabilityRecordValue	0..1

Parameter Name	SdServerCapabilityRecordKey
Description	Defines a CapabilityRecord key.
Multiplicity	1..1
Type	STRING
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

Parameter Name	SdServerCapabilityRecordValue
Description	Defines the corresponding CapabilityRecord value.
Multiplicity	0..1
Type	STRING
Configuration class	PostBuild:
Origin	AUTOSAR_ECUC

5.5.1.24. SdServerTimer

Parameters included	
Parameter name	Multiplicity
SdServerTimerInitialOfferDelayMax	0..1



Parameters included

SdServerTimerInitialOfferDelayMin	0..1
SdServerTimerInitialOfferRepetitionBaseDelay	0..1
SdServerTimerInitialOfferRepetitionsMax	0..1
SdServerTimerOfferCyclicDelay	0..1
SdServerTimerRequestResponseMaxDelay	1..1
SdServerTimerRequestResponseMinDelay	1..1
SdServerTimerTTL	1..1

Parameter Name	SdServerTimerInitialOfferDelayMax	
Description	Max value in [s] to delay randomly the first offer. This parameter is mandatory for ServerService.	
Multiplicity	0..1	
Type	FLOAT	
Default value	0.0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdServerTimerInitialOfferDelayMin	
Description	Min value in [s] to delay randomly the first offer. This parameter is mandatory for ServerService.	
Multiplicity	0..1	
Type	FLOAT	
Range	<=4294967294 >=0.0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdServerTimerInitialOfferRepetitionBaseDelay	
Description	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.	
Multiplicity	0..1	
Type	FLOAT	



Range	<=Infinity >=0.0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdServerTimerInitialOfferRepetitionsMax	
Description	Configure the maximum amount of offer repetition. This parameter is mandatory for ServerService.	
Multiplicity	0..1	
Type	INTEGER	
Default value	3	
Range	<=10 >=0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdServerTimerOfferCyclicDelay	
Description	Interval between cyclic offers in the main phase. This parameter is mandatory for ServerService.	
Multiplicity	0..1	
Type	FLOAT	
Range	<=Infinity >=0.0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdServerTimerRequestResponseMaxDelay	
Description	Maximum allowable response delay to entries received by multicast in seconds.	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	



Parameter Name	SdServerTimerRequestResponseMinDelay	
Description	Minimum allowable response delay to entries received by multicast in seconds.	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.0	
Range	<=4294967294 >=0.0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdServerTimerTTL	
Description	Time to live for offer service.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=16777215 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.5.1.25. SdGeneral

Containers included		
Container name	Multiplicity	Description
VendorSpecific	1..1	Contains the vendor specific configuration parameters of the AUTOSAR Sd module.

Parameters included	
Parameter name	Multiplicity
SdHeaderFileInclusion	0..n
SdDevErrorDetect	1..1
SdMainFunctionCycleTime	1..1
SdSubscribeEventgroupRetryEnable	1..1
SdVersionInfoApi	1..1



Parameters included

SdRelocatablePbcfgEnable	1..1
SdCallOutEventMapping	1..1
SdMaxInstanceCount	1..1
SdMaxClientServiceCount	1..1
SdMaxServerServiceCount	1..1
SdMaxClientServiceEventGroupCount	1..1
SdMaxServerServiceEventCount	1..1
SdMaxConsumedEventGroupMulticastReferences	1..1

Parameter Name	SdHeaderFileInclusion
Multiplicity	0..n
Type	STRING
Origin	AUTOSAR_ECUC

Parameter Name	SdDevErrorDetect
Description	Enables and disables the development error detection and notification mechanism.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SdMainFunctionCycleTime
Description	This parameter defines the cycle time in seconds of the periodic calling of Sd main function.
Multiplicity	1..1
Type	FLOAT
Default value	0.005
Range	<=1.0 >=1.0E-4
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC



Parameter Name	SdSubscribeEventgroupRetryEnable	
Description	Switch to enable or disable retry functionality to subscribe to Eventgroups of ServerServices with TTL set to 0xFFFFFFF.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdVersionInfoApi	
Description	Enables and disables the version info API.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdRelocatablePbcfgEnable	
Description	Enables/disable support for relocatable postbuild configuration. ▶ True: Postbuild configuration relocatable in memory. ▶ False: Postbuild configuration not relocatable in memory.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SdCallOutEventMapping	
Description	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.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	SdMaxInstanceCount
Description	Maximum number of instances that can be defined in the configuration.
Multiplicity	1..1
Type	INTEGER
Default value	8
Configuration class	VariantPostBuild:
Origin	Elektrobit Automotive GmbH

Parameter Name	SdMaxClientServiceCount
Description	Maximum number of client services that can be defined in the configuration.
Multiplicity	1..1
Type	INTEGER
Default value	32
Configuration class	VariantPostBuild:
Origin	Elektrobit Automotive GmbH

Parameter Name	SdMaxServerServiceCount
Description	Maximum number of client services that can be defined in the configuration.
Multiplicity	1..1
Type	INTEGER
Default value	32
Configuration class	VariantPostBuild:
Origin	Elektrobit Automotive GmbH

Parameter Name	SdMaxClientServiceEventGroupCount
Description	Maximum number of client services that can be defined in the configuration.
Multiplicity	1..1
Type	INTEGER
Default value	32
Configuration class	VariantPostBuild:



Origin	Elektrobit Automotive GmbH
---------------	----------------------------

Parameter Name	SdMaxServerServiceEventCount	
Description	Maximum number of server services that can be defined in the configuration.	
Multiplicity	1..1	
Type	INTEGER	
Default value	32	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	SdMaxConsumedEventGroupMulticastReferences	
Description	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	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.5.1.26. VendorSpecific

Containers included		
Container name	Multiplicity	Description
ProductionErrorHandler	1..1	
Sd_SendDataBackServerNotAvailable	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>
Sd_SendDataBackMalformedMsg	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>
Sd_SendDataBackSubscribeNackReceived	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
SdTxFrameBufferSize	1..1
SdTxOptionsBufferSize	1..1
SdRxOptionsBufferSize	1..1

Parameter Name	SdTxFrameBufferSize
Description	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.
Multiplicity	1..1
Type	INTEGER
Default value	1408
Range	<=1408 >=40
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SdTxOptionsBufferSize
Description	Defines the maximum number of options transmitted within a Sd control messages.
Multiplicity	1..1
Type	INTEGER
Default value	40
Range	<=250 >=3
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SdRxOptionsBufferSize
Description	Defines the maximum number of options being processes in a received Sd control messages.
Multiplicity	1..1
Type	INTEGER



Default value	120
Range	<=250 >=3
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.5.1.27. ProductionErrorHandler

Parameters included	
Parameter name	Multiplicity
Enable_SD_E_MALFORMED_MSG	1..1
SD_E_MALFORMED_MSG_ReportToDemDetErrorId	1..1
Enable_SD_E_OUT_OF_RES	1..1
SD_E_OUT_OF_RES_ReportToDemDetErrorId	1..1
Enable_SD_E_SUBSCR_NACK_RECV	1..1
SD_E_SUBSCR_NACK_RECV_ReportToDemDetErrorId	1..1
Enable_SD_E_SERVER_NOT_AVAILABLE	1..1
SD_E_SERVER_NOT_AVAILABLE_ReportToDemDetErrorId	1..1

Parameter Name	Enable_SD_E_MALFORMED_MSG
Label	Enable SD_E_MALFORMED_MSG
Description	<p>Enables DEM reporting in SD.</p> <ul style="list-style-type: none"> ▶ DEM: All errors are reported to the Diagnostics Event Manager (Dem). ▶ DET: All errors are reported to the Development Error Tracer (Det) if enabled. ▶ DISABLE: Production errors are not reported at all. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code. ▶ Execution time reduction (code): Disabling this parameter reduces the execution time of the module code.
Multiplicity	1..1



Type	ENUMERATION	
Default value	DISABLE	
Range	DEM DET DISABLE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	SD_E_MALFORMED_MSG_ReportToDemDetErrorId	
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	Enable_SD_E_OUT_OF_RES	
Label	Enable SD_E_OUT_OF_RES	
Description	<p>Enables DEM reporting in SD.</p> <ul style="list-style-type: none"> ▶ DEM: All errors are reported to the Diagnostics Event Manager (Dem). ▶ DET: All errors are reported to the Development Error Tracer (Det) if enabled. ▶ DISABLE: Production errors are not reported at all. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code. ▶ Execution time reduction (code): Disabling this parameter reduces the execution time of the module code. 	
Multiplicity	1..1	
Type	ENUMERATION	



Default value	DISABLE
Range	DEM DET DISABLE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SD_E_OUT_OF_RES_ReportToDemDetErrorId
Label	SD_E_OUT_OF_RES 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	130
Range	<=255
Configuration class	PreCompile: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	Enable_SD_E_SUBSCR_NACK_RECV
Label	Enable SD_E_SUBSCR_NACK_RECV
Description	<p>Enables DEM reporting in SD.</p> <ul style="list-style-type: none"> ▶ DEM: All errors are reported to the Diagnostics Event Manager (Dem). ▶ DET: All errors are reported to the Development Error Tracer (Det) if enabled. ▶ DISABLE: Production errors are not reported at all. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code. ▶ Execution time reduction (code): Disabling this parameter reduces the execution time of the module code.
Multiplicity	1..1
Type	ENUMERATION
Default value	DISABLE



Range	DEM DET DISABLE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SD_E_SUBSCR_NACK_RECV_ReportToDemDetErrorId
Label	SD_E_SUBSCR_NACK_RECV 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	131
Range	<=255
Configuration class	PreCompile: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	Enable_SD_E_SERVER_NOT_AVAILABLE
Label	Enable SD_E_SERVER_NOT_AVAILABLE
Description	<p>Enables DEM reporting in SD.</p> <ul style="list-style-type: none"> ▶ DEM: All errors are reported to the Diagnostics Event Manager (Dem). ▶ DET: All errors are reported to the Development Error Tracer (Det) if enabled. ▶ DISABLE: Production errors are not reported at all. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code. ▶ Execution time reduction (code): Disabling this parameter reduces the execution time of the module code.
Multiplicity	1..1
Type	ENUMERATION
Default value	DISABLE
Range	DEM



	DET
	DISABLE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	SD_E_SERVER_NOT_AVAILABLE_ReportToDemDetErrorId
Label	SD_E_SERVER_NOT_AVAILABLE 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	133
Range	<=255
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.5.1.28. Sd_SendDataBackServerNotAvailable

Parameters included	
Parameter name	Multiplicity
Sd_ServerNotAvailableCallBack	1..1

Parameter Name	Sd_ServerNotAvailableCallBack
Description	<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>Syntax: (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>Note: User defined header files can be added to configuration container Sd-HeaderFileInclusion.</p>



Multiplicity	1..1
Type	FUNCTION-NAME
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.5.1.29. Sd_SendDataBackMalformedMsg

Parameters included	
Parameter name	Multiplicity
Sd_MalformedMsgCallBack	1..1

Parameter Name	Sd_MalformedMsgCallBack
Description	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. Note: User defined header files can be added to configuration container Sd-HeaderFileInclusion.
Multiplicity	1..1
Type	FUNCTION-NAME
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.5.1.30. Sd_SendDataBackSubscribeNackReceived

Parameters included	
Parameter name	Multiplicity
Sd_SubscribeNackReceivedCallBack	1..1

Parameter Name	Sd_SubscribeNackReceivedCallBack
Description	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>Note: User defined header files can be added to configuration container Sd-HeaderFileInclusion.</p>
Multiplicity	1..1
Type	FUNCTION-NAME
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.5.1.31. SdDefensiveProgramming

Parameters included	
Parameter name	Multiplicity
SdDefProgEnabled	1..1
SdPrecondAssertEnabled	1..1
SdPostcondAssertEnabled	1..1
SdStaticAssertEnabled	1..1
SdUnreachAssertEnabled	1..1
SdInvariantAssertEnabled	1..1

Parameter Name	SdDefProgEnabled
Label	Enable Defensive Programming
Description	<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"> 1. Enable development error detection 2. Enable defensive programming 3. Enable assertions as required
Multiplicity	1..1
Type	BOOLEAN
Default value	false



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	SdPrecondAssertEnabled	
Label	Enable Precondition Assertions	
Description	<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"> ▶ Enable Development Error Detection (<code>SdDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>SdDefProgEnabled</code>): must be enabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	SdPostcondAssertEnabled	
Label	Enable Postcondition Assertions	
Description	<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"> ▶ Enable Development Error Detection (<code>SdDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>SdDefProgEnabled</code>): must be enabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	SdStaticAssertEnabled	
Label	Enable Static Assertions	
Description	Enables handling of static assertion checks reported from the module Sd.	



	<p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Enable Development Error Detection (<code>SdDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>SdDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	SdUnreachAssertEnabled
Label	Enable Unreachable Code Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>SdDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>SdDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	SdInvariantAssertEnabled
Label	Enable Invariant Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>SdDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>SdDefProgEnabled</code>): must be enabled
Multiplicity	1..1



Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: 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

Purpose	Defines API id of function Sd_ClientService_mainTimersFunction.
Value	0x16U

5.5.2.1.6. SD_CLIENTSERVICEMAITTTL_SVCID

Purpose	Defines API id of function Sd_ClientService_mainTTL.
Value	0x17U

5.5.2.1.7. SD_CLIENTSERVICESETSTATE_SVCID

Purpose	Defines API id of function Sd_ClientServiceSetState.
Value	0x08U

5.5.2.1.8. SD_CLIENTSERVICESINITAFTERTHALT_SVCID

Purpose	Defines API id of function Sd_ClientServices_init_after_halt.
Value	0x1AU

5.5.2.1.9. SD_CLIENTSERVICE_COMSMEVENT_OFFER_SVCID

Purpose	Defines API id of function Sd_ClientService_ComSM_event_Offer.
Value	0x13U

5.5.2.1.10. SD_CLIENTSERVICE_COMSMEVENT_START_SVCID

Purpose	Defines API id of function Sd_ClientService_ComSM_event_start.
Value	0x10U

5.5.2.1.11. SD_CLIENTSERVICE_COMSMEVENT_STOPOFFER_SVCID

Purpose	Defines API id of function Sd_ClientService_ComSM_event_Stopoffer.
----------------	--



Value	0x14U
--------------	-------

5.5.2.1.12. SD_CLIENTSERVICE_COMSMEVENT_STOP_SVCID

Purpose	Defines API id of function Sd_ClientService_ComSM_event_stop.
Value	0x11U

5.5.2.1.13. SD_CLIENTSERVICE_COMSMEVENT_TIMEOUT_SVCID

Purpose	Defines API id of function Sd_ClientService_ComSM_event_timeout.
Value	0x12U

5.5.2.1.14. SD_CLIENTSERVICE_COMSM_EVENT_STOP_SVCID

Purpose	Defines API id of function Sd_ClientService_ComSM_event_stop.
Value	0x1FU

5.5.2.1.15. SD_CLIENTSERVICE_COMSM_EVENT_TIMEOUT

Purpose	Defines API id of function Sd_ClientService_ComSM_event_timeout.
Value	0x0BU

5.5.2.1.16. SD_CLIENTSERVICE_COMSM_SOCONMODECHECK_SVCID

Purpose	Defines API id of function Sd_ClientService_ComSM_SoConModeCheck.
Value	0x15U

5.5.2.1.17. SD_CLIENTSERVICE_CONSUMEDEVENTGROUPSETSTATE_SVCID

Purpose	Defines API id of function Sd_ClientService_consumedEventGroupSetState.
----------------	---



Value	0x1EU
--------------	-------

5.5.2.1.18. SD_CLIENTSERVICE_EVENT_OFFER_SERVICE_SVCID

Purpose	Defines API id of function Sd_ClientService_eventOfferService.
Value	0x1CU

5.5.2.1.19. SD_CLIENTSERVICE_PROCESSOFFERS_SVCID

Purpose	Defines API id of function Sd_ClientService_ProcessOffers.
Value	0x1DU

5.5.2.1.20. SD_CLIENTSERVICE_TTL_RUN_OUT

Purpose	Defines API id of function Sd_ClientService_TTL_Run_Out.
Value	0x0CU

5.5.2.1.21. SD_COMPAREADDRESS_SVCID

Purpose	Defines API id of function Sd_LocallpAddrAssignmentChg.
Value	0x04U

5.5.2.1.22. SD_CONNECTION_SETTING_FAILED

Purpose	service called with invalid mode request
Value	0x0BU

5.5.2.1.23. SD_CONSUMEDEVENTGROUPSETSTATE_SVCID

Purpose	Defines API id of function Sd_ConsumedEventGroupSetState.
----------------	---



Value	0x09U
--------------	-------

5.5.2.1.24. SD_CONTROL_SOCKET_IMPOSSIBLE

Purpose	service called with invalid mode request
Value	0x0AU

5.5.2.1.25. SD_CONTROL_SOCKET_NOT_RELEASED

Purpose	service called with invalid mode request
Value	0x09U

5.5.2.1.26. SD_E_CLIENT_IMPOSSIBLE_STATE

Purpose	Initialization failed error.
Value	0x07U

5.5.2.1.27. SD_E_COUNT_OF_RETRY_SUBSCRIPTION_EXCEEDED

Purpose	Initialization failed error.
Value	0x06U

5.5.2.1.28. SD_E_DIVISOR_IS_NEGATIVE

Purpose	Divisor is negative error.
Value	0x0DU

5.5.2.1.29. SD_E_INVALID_ARG

Purpose	Initialization failed error.
----------------	------------------------------



Value	0x03U
--------------	-------

5.5.2.1.30. SD_E_INV_ID

Purpose	API service called with invalid Id.
Value	0x04U

5.5.2.1.31. SD_E_INV_MODE

Purpose	API service called with invalid mode request.
Value	0x05U

5.5.2.1.32. SD_E_IPADDR_ASSIGNMENT_NOT_FINISHED

Purpose	IP Address assignment not finished error.
Value	0x0CU

5.5.2.1.33. SD_E_NOT_INITIALIZED

Purpose	API service called before initializing the module.
Value	0x01U

5.5.2.1.34. SD_E_OUT_OF_RES

Purpose	Initialization failed error.
Value	0x08U

5.5.2.1.35. SD_E_PARAM_POINTER

Purpose	Null pointer has been passed as an argument.
Value	0x02U



5.5.2.1.36. SD_E_RETRY_INFINITE_TTL_NOT_INFINITE

Purpose	IP Address assignment not finished error.
Value	0x0EU

5.5.2.1.37. SD_E_RETRY_PROCESS_BIGGER_THAN_TTL

Purpose	Divisor is negative error.
Value	0x0FU

5.5.2.1.38. SD_E_SHOULD_NOT_HAPPEN

Purpose	Definition of defensive programming SD_E_SHOULD_NOT_HAPPEN.
Value	0xffU

5.5.2.1.39. SD_GETRANDOM_SVCID

Purpose	Defines API id of function Sd_Get_Random.
Value	0x60U

5.5.2.1.40. SD_GETVERSIONINFO_SVCID

Purpose	Defines API id of function Sd_GetVersionInfo.
Value	0x02U

5.5.2.1.41. SD_HANDLERESPONSETIMERS_SVCID

Purpose	Defines API id of function Sd_HandleResponseTimers.
Value	0x62U

5.5.2.1.42. SD_INIT_SVCID

Purpose	Defines API id of function Sd_Init.
----------------	-------------------------------------



Value	0x01U
--------------	-------

5.5.2.1.43. SD_INSTANCEMAINFUNCTION_SVCID

Purpose	Defines API id of function Sd_Instance_MainFunction.
Value	0x0AU

5.5.2.1.44. SD_INSTANCE_ID

Purpose	AUTOSAR module identification.
Value	0U

5.5.2.1.45. SD_INTERNAL_API_ID

Purpose	API ID of module internal functions to be used in assertions.
Value	DET_INTERNAL_API_ID

5.5.2.1.46. SD_INVARIANT_ASSERT

Purpose	Report an assertion violation to Det.
Value	DET_INVARIANT_ASSERT((Condition), SD_MODULE_ID, SD_INSTANCE_ID, (Apild))

5.5.2.1.47. SD_INVARIANT_ASSERT_NO_EVAL

Purpose	Report an assertion violation to Det.
Value	DET_INVARIANT_ASSERT_NO_EVAL((Condition), SD_MODULE_ID, SD_INSTANCE_ID, (Apild))

5.5.2.1.48. SD_LOCALIPADDRASSIGNMENTCHG_SVCID

Purpose	Defines API id of function Sd_LocalIpAddrAssignmentChg.
----------------	---



Value	0x05U
--------------	-------

5.5.2.1.49. SD_MAINFUNCTION_SVCID

Purpose	Defines API id of function Sd_MainFunction.
Value	0x06U

5.5.2.1.50. SD_POSTCONDITION_ASSERT

Purpose	Report an assertion violation to Det.
Value	DET_POSTCONDITION_ASSERT((Condition), SD_MODULE_ID, SD_INSTANCE_ID, (Apild))

5.5.2.1.51. SD_POSTCONDITION_ASSERT_NO_EVAL

Purpose	Report an assertion violation to Det.
Value	DET_POSTCONDITION_ASSERT_NO_EVAL((Condition), SD_MODULE_ID, SD_INSTANCE_ID, (Apild))

5.5.2.1.52. SD_PRECONDITION_ASSERT

Purpose	Report an assertion violation to Det.
Value	DET_PRECONDITION_ASSERT((Condition), SD_MODULE_ID, SD_INSTANCE_ID, (Apild))

5.5.2.1.53. SD_PRECONDITION_ASSERT_NO_EVAL

Purpose	Report an assertion violation to Det.
Value	DET_PRECONDITION_ASSERT_NO_EVAL((Condition), SD_MODULE_ID, SD_INSTANCE_ID, (Apild))

5.5.2.1.54. SD_RECEIVEGETENTRYOPTIONS_SVCID

Purpose	Defines API id of function Sd_Receive_getEntryOptions.
----------------	--



Value	0x50U
--------------	-------

5.5.2.1.55. SD_RESETREMOTENODERXSTATE_SVCID

Purpose	Defines API id of function Sd_ResetRemoteNodeRxState.
Value	0x61U

5.5.2.1.56. SD_RXINDICATION_SVCID

Purpose	Defines API id of function Sd_RxIndication.
Value	0x42U

5.5.2.1.57. SD_SENDQUEUEOPTIONS_SVCID

Purpose	Defines API id of function Sd_Send_queueOptions.
Value	0x43U

5.5.2.1.58. SD_SEND_FLUSHBUFFER_SVCID

Purpose	Defines API id of function Sd_Send_flushBuffer.
Value	0x45U

5.5.2.1.59. SD_SERVERSERVICEADDEVENTGROUPSUBS_SVCID

Purpose	Defines API id of function Sd_ServerService_addEventGroupSubscriber.
Value	0x29U

5.5.2.1.60. SD_SERVERSERVICEADDMULTEVENTGROUPSUBS_SVCID

Purpose	Defines API id of function Sd_ServerService_addMulticastEventGroupSubscriber.
----------------	---



Value	0x28U
--------------	-------

5.5.2.1.61. SD_SERVERSERVICEEVENTSTOPSUBSCRIBE_SVCID

Purpose	Defines API id of function Sd_ServerService_eventStopSubscribe.
Value	0x27U

5.5.2.1.62. SD_SERVERSERVICEHANDLETTL_SVCID

Purpose	Defines API id of function Sd_ClientService_mainTTL.
Value	0x24U

5.5.2.1.63. SD_SERVERSERVICEMAINFUNCTION_SVCID

Purpose	Defines API id of function Sd_ClientService_mainFunction.
Value	0x25U

5.5.2.1.64. SD_SERVERSERVICEMAITIMERS_SVCID

Purpose	Defines API id of function Sd_ClientService_mainTimersFunction.
Value	0x23U

5.5.2.1.65. SD_SERVERSERVICEREMOVEEVENTGROUPSUBS_SVCID

Purpose	Defines API id of function Sd_ServerService_removeEventGroupSubscriber.
Value	0x2AU

5.5.2.1.66. SD_SERVERSERVICERESETREMOTECONNECTION_SVCID

Purpose	Defines API id of function Sd_ServerService_resetRemoteConnection.
----------------	--



Value	0x26U
--------------	-------

5.5.2.1.67. SD_SERVERSERVICESETSTATE_SVCID

Purpose	Defines API id of function Sd_ServerServiceSetState.
Value	0x07U

5.5.2.1.68. SD_SERVERSERVICE_COMSMEVENT_START_SVCID

Purpose	Defines API id of function Sd_ServerService_ComSM_event_start.
Value	0x20U

5.5.2.1.69. SD_SERVERSERVICE_COMSMEVENT_STOPHALT_SVCID

Purpose	Defines API id of function Sd_ServerService_ComSM_event_stophalt.
Value	0x21U

5.5.2.1.70. SD_SERVERSERVICE_COMSMEVENT_TIMEOUT_SVCID

Purpose	Defines API id of function Sd_ServerService_ComSM_event_timeout.
Value	0x22U

5.5.2.1.71. SD_STATIC_ASSERT

Purpose	Report an static assertion violation to Det.
Value	DET_STATIC_ASSERT(expr)

5.5.2.1.72. SD_UNREACHABLE_CODE_ASSERT

Purpose	Report an unreachable code assertion violation to Det.
Value	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 Sd_ClientServiceSetState (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 Sd_ConsumedEventGroupSetState (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



Description	This API function sets the Client EventGroup state.
--------------------	---

5.5.2.2.3. Sd_GetVersionInfo

Purpose	Get version information of the Sd module.	
Synopsis	<pre>void Sd_GetVersionInfo (Std_VersionInfoType * Sd_VersionInfo);</pre>	
Service ID	0x0B	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	versioninfo	Pointer to where to store the version information of this module.
Description	<p>This service returns the version information of this module. The version information includes:</p> <ul style="list-style-type: none"> ➤ Module Id ➤ Vendor Id ➤ Vendor specific version numbers 	

5.5.2.2.4. Sd_Init

Purpose	Initializes the Sd module.
Synopsis	<pre>void Sd_Init (const Sd_ConfigType * ConfigPtr);</pre>
Service ID	0x01
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Description	This service initializes Sd module internal variables.

5.5.2.2.5. Sd_IsValidConfig

Purpose	Checks compatibility of the post-build-time configuration.
Synopsis	<pre>Std_ReturnType Sd_IsValidConfig (const void * voidConfigPtr);</pre>



Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	voidConfigPtr	Pointer to the configuration data of the Sd module.
Return Value	Result of compatibility check	
	E_OK	Provided configuration is compatible.
	E_NOT_OK	Provided configuration is not compatible.
Description	This service checks the compatibility of the post-build-time configuration against the source code.	

5.5.2.2.6. Sd_LocalIpAddrAssignmentChg

Purpose	Local IP address assignment change indication.	
Synopsis	<pre>void Sd_LocalIpAddrAssignmentChg (SoAd_SoConIdType SoConId , TcpIp_IpAddrStateType State);</pre>	
Service ID	0x05	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SoConId	Socket connection index specifying the socket connection where the IP address assignment has changed.
	State	State of IP address assignment.
Description	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

Purpose	Main Function of module Sd.
Synopsis	<pre>void Sd_MainFunction (void);</pre>
Service ID	0x06
Sync/Async	Synchronous
Reentrancy	Non-Reentrant



Description	This API triggers all periodic actions to be performed by Sd.
--------------------	---

5.5.2.2.8. Sd_RxIndication

Purpose	Sd Rx Indication.	
Synopsis	<code>void Sd_RxIndication (PduIdType RxPduId , PduInfoType * PduInfoPtr);</code>	
Service ID	0x42	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	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.
Description	Indication of a received I-PDU from a lower layer communication interface module.	

5.5.2.2.9. Sd_ServerServiceSetState

Purpose	Sets the Server Service instance state.	
Synopsis	<code>Std_ReturnType Sd_ServerServiceSetState (uint16 ServiceServiceHandleId , Sd_ServerServiceSetStateType ServerServiceState);</code>	
Service ID	0x07	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	ServiceServiceHandleId	ID to identify the Server Service instance.
	ServerServiceState	The state the Server 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 Server Service instance state.	



5.5.2.2.10. Sd_StartRandom

Purpose	Initialize the random number generator.	
Synopsis	<code>void Sd_StartRandom (uint32 Seed);</code>	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	Seed	Start value for number generator
Parameters (in,out)	Seed	Start value for number generator
Description	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

Protected data structures	All shared data that shall be protected from mutual access.
Recommended locking mechanism	<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

Description	The integrator must assure that the following functions do not interrupt each other or themselves: <ul style="list-style-type: none">▶ Sd_MainFunction▶ Sd_RxIndication▶ Sd_LocallpAddrAssignmentChg▶ Sd_StartRandom
Rationale	This limitation reduces code size and execution time.



5.5.3.4.2. Sd.EB.IntReq.EB_INTREQ_Sd_0002

Description	The integrator must assure that the following functions do not interrupt each other or themselves: <ul style="list-style-type: none">▶ Sd_ServerServiceSetState▶ Sd_ClientServiceSetStat▶ Sd_ConsumedEventGroupSetState
Rationale	This limitation reduces code size and execution time.

5.5.3.4.3. Sd.EB.IntReq.EB_INTREQ_Sd_0003

Description	The integrator must assure that all Client services using the same SocketConnection are provided on the same endpoint.
Rationale	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
CommonPublishedInformation	1..1	Label: Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
PublishedInformation	1..1	Label: EB Published Information Additional published parameters not covered by CommonPublishedInformation container.
SoAdDefensiveProgramming	1..1	Label: Defensive Programming Options Parameters for defensive programming
SoAdGeneral	1..1	This container contains all global configuration parameters of SoAd.

**Containers included**

SoAdBswModules	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).
SoAdConfig	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
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT
Label	Config Variant
Multiplicity	1..1
Type	ENUMERATION
Default value	VariantPostBuild
Range	VariantPostBuild

5.6.1.1. CommonPublishedInformation**Parameters included**

Parameter name	Multiplicity
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1
SwPatchVersion	1..1
ModuleId	1..1

Parameters included

VendorId	1..1
Release	1..1

Parameter Name	ArMajorVersion
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	ArMinorVersion
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	2
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ArPatchVersion
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	2
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMajorVersion
-----------------------	-----------------------



Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMinorVersion
Label	Software Minor Version
Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	8
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwPatchVersion
Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	22
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ModuleId
Label	Numeric Module ID
Description	Module ID of this module from Module List
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	56



Configuration class	PublishedInformation:	
Origin	Elektrobit Automotive GmbH	

Parameter Name	VendorId
Label	Vendor ID
Description	Vendor ID of the dedicated implementation of this module according to the AU-TOSAR vendor list
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	Release
Label	Release Information
Multiplicity	1..1
Type	STRING_LABEL
Default value	
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

5.6.1.2. PublishedInformation

Parameters included	
Parameter name	Multiplicity
PbcfgMSupport	1..1

Parameter Name	PbcfgMSupport
Label	PbcfgM support
Description	Specifies whether or not the SoAd can use the PbcfgM module for post-build support.
Multiplicity	1..1
Type	BOOLEAN
Default value	true



Configuration class	PublishedInformation:	
Origin	Elektrobit Automotive GmbH	

5.6.1.3. SoAdDefensiveProgramming

Parameters included	
Parameter name	Multiplicity
SoAdDefProgEnabled	1..1
SoAdPrecondAssertEnabled	1..1
SoAdPostcondAssertEnabled	1..1
SoAdStaticAssertEnabled	1..1
SoAdUnreachAssertEnabled	1..1
SoAdInvariantAssertEnabled	1..1

Parameter Name	SoAdDefProgEnabled
Label	Enable Defensive Programming
Description	<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"> 1. Enable development error detection 2. Enable defensive programming 3. Enable assertions as required
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SoAdPrecondAssertEnabled
Label	Enable Precondition Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>SoAdDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>SoAdDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SoAdPostcondAssertEnabled
Label	Enable Postcondition Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>SoAdDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>SoAdDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SoAdStaticAssertEnabled
Label	Enable Static Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>SoAdDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>SoAdDefProgEnabled</code>): must be enabled
Multiplicity	1..1



Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	SoAdUnreachAssertEnabled	
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"> ▶ Enable Development Error Detection (<code>SoAdDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>SoAdDefProgEnabled</code>): must be enabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	SoAdInvariantAssertEnabled	
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"> ▶ Enable Development Error Detection (<code>SoAdDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>SoAdDefProgEnabled</code>): must be enabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	Elektrobit Automotive GmbH
---------------	----------------------------

5.6.1.4. SoAdGeneral

Containers included		
Container name	Multiplicity	Description
SoAdSecurityEventRefs	1..1	Container for the references to IdsMEvent elements representing the security events that the SoAd module shall report to the IdsM in case the corresponding security related event occurs (and if SoAdEnableSecurityEventReporting is set to "true").
VendorSpecific	1..1	Contains the vendor specific configuration parameters of the AUTOSAR SoAd module.

Parameters included	
Parameter name	Multiplicity
SoAdDevErrorDetect	1..1
SoAdGetAndResetMeasurementDataApi	1..1
SoAdIPv6AddressEnabled	1..1
SoAdMainFunctionPeriod	1..1
SoAdSoConMax	1..1
SoAdRoutingGroupMax	1..1
SoAdVersionInfoApi	1..1
SoAdTIsEnabled	1..1
SoAdEnableSecurityEventReporting	1..1
SoAdMainFunctionPeriodTx	1..1
SoAdEnableMainFunctionTx	1..1

Parameter Name	SoAdDevErrorDetect	
Description	Pre-processor switch for enabling development error detection support.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	



Parameter Name	SoAdGetAndResetMeasurementDataApi	
Description	Pre-processor switch for enabling the Get and Reset Measurement Data API.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdIPv6AddressEnabled	
Description	Allows for increased memory allocation to store IPv6 addresses. ▶ true: Enables support for IPv6 addresses ▶ false: Only IPv4 addresses are supported	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdMainFunctionPeriod	
Description	Determines the frequency at which the SoAd_MainFunction() is called in [s].	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSoConMax	
Description	Specifies the maximum number of SoAd socket connections. 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).	
Multiplicity	1..1	
Type	INTEGER	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	SoAdRoutingGroupMax	
Description	Specifies the maximum number of SoAd routing groups. 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).	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	SoAdVersionInfoApi	
Description	Activates the SoAd_GetVersionInfo() API. ▶ true: Enables the SoAd_GetVersionInfo() API. ▶ false: SoAd_GetVersionInfo() API is not included.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	SoAdTlsEnabled	
Description	▶ true: Enables support for TLS extension. ▶ false: Disables support for TLS extension.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	SoAdEnableSecurityEventReporting
-----------------------	---



Description	Enables/disables reporting of security events to the IdsM	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	SoAdMainFunctionPeriodTx	
Description	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).	
Multiplicity	1..1	
Type	FLOAT	
Configuration class	PreCompile:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	SoAdEnableMainFunctionTx	
Description	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.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.6.1.5. SoAdSecurityEventRefs

Parameters included	
Parameter name	Multiplicity

**Parameters included**

SOAD_SEV_DROP_PDU_RX_TCP	0..1
SOAD_SEV_DROP_PDU_RX_UDP	0..1
SOAD_SEV_DROP_MSG_RX_UDP_LENGTH	0..1
SOAD_SEV_DROP_MSG_RX_UDP_SOCKET	0..1
SOAD_SEV_REJECTED_TCP_CONNECTION	0..1

Parameter Name	SOAD_SEV_DROP_PDU_RX_TCP	
Description	Reference to the IdsMEvent SOAD_SEV_DROP_PDU_RX_TCP: SoAd dropped a PDU. The PDU violates stack configuration and was received via a TCP socket.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SOAD_SEV_DROP_PDU_RX_UDP	
Description	Reference to the IdsMEvent SOAD_SEV_DROP_PDU_RX_UDP: SoAd dropped a PDU. The PDU violates stack configuration and was received via a UDP socket	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SOAD_SEV_DROP_MSG_RX_UDP_LENGTH	
Description	Reference to the IdsMEvent SOAD_SEV_DROP_MSG_RX_UDP_LENGTH: SoAd dropped a message. The message contains at least one PDU which violates stack configuration and was received via a UDP socket. The violation relates to the length of the PDUs compared to the overall length of the message	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SOAD_SEV_DROP_MSG_RX_UDP_SOCKET	
-----------------------	--	--



Description	Reference to the IdsMEvent SOAD_SEV_DROP_MSG_RX_UDP_SOCKET: SoAd received a UDP message which violates stack configuration and was dropped. No suitable socket connection matching to configuration was found.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SOAD_SEV_REJECTED_TCP_CONNECTION	
Description	Reference to the IdsMEvent SOAD_SEV_REJECTED_TCP_CONNECTION: SoAd rejected a TCP connection. The connection request violates stack configuration.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.6.1.6. VendorSpecific

Parameters included		
Parameter name	Multiplicity	
SoAdDataMemSize	0..1	
SoAdDataMemSizeExtension	1..1	
SoAdRelocatableCfgEnable	1..1	
SoAdRouteMax	1..1	
SoAdTimerType	1..1	
SoAdSoConIdType	1..1	
SoAdRoutingGroupIdType	1..1	
SoAdReportInvPDUHeaderIdToDetEnable	1..1	
SoAdSocketRouteDestFanInEnable	1..1	

Parameter Name	SoAdDataMemSize
Description	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. This parameter shall be used to reserve RAM at precompile time to be used for configuration changes at post-build time.
Multiplicity	0..1
Type	INTEGER
Default value	0
Configuration class	Link: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SoAdDataMemSizeExtension
Description	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). Optimization Effect: ▶ ROM reduction (config): Disabling this parameter reduces the ROM consumption of the module configuration.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SoAdRelocatableCfgEnable
Description	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).
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH



Parameter Name	SoAdRouteMax
Description	<p>This parameter defines the maximum number of SoAdPduRouteDest containers and the maximum number of SoAdSocketRouteDest containers.</p> <ul style="list-style-type: none"> ▶ INDEX_UINT8: The maximum number of SoAdPduRouteDest containers is 255 and the maximum number of SoAdSocketRouteDest containers is 255. ▶ INDEX_UINT16: The maximum number of SoAdPduRouteDest containers is 65535 and the maximum number of SoAdSocketRouteDest containers is 65535. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (config): The smaller the index, the smaller the ROM consumption of the module configuration.
Multiplicity	1..1
Type	ENUMERATION
Default value	INDEX_UINT16
Range	INDEX_UINT8 INDEX_UINT16
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SoAdTimerType
Description	<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>Example:</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"> ▶ UINT8: values up to 2.55 s ▶ UINT16: values up to 655.35 s (~11 min) ▶ UINT32: values up to 42949672.95 s (well over one year) <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (config): The smaller the type, the smaller the ROM consumption of the module configuration.



Multiplicity	1..1
Type	ENUMERATION
Default value	UINT8
Range	UINT8 UINT16 UINT32
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SoAdSoConIdType
Description	<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>Possible values:</p> <ul style="list-style-type: none"> ▶ UINT8: The maximum number of connections is 254. ▶ UINT16: The maximum number of connections is 65534. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (config): The smaller the type, the smaller the ROM consumption of the module configuration. ▶ RAM reduction (config): The smaller the type, the smaller the RAM consumption of the module.
Multiplicity	1..1
Type	ENUMERATION
Default value	UINT16
Range	UINT8 UINT16
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SoAdRoutingGroupIdType
Description	<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>The number of routing groups can be higher than the maximum of SoAd- RoutingGroupId if specific routing is used.</p> <p>Possible values:</p> <ul style="list-style-type: none"> ▶ UINT8: The maximum number of connections is 254. ▶ UINT16: The maximum number of connections is 65534. ▶ UINT32: The maximum number of connections is 4294967295. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (config): The smaller the type, the smaller the ROM con- sumption of the module configuration. ▶ RAM reduction (config): The smaller the type, the smaller the RAM con- sumption of the module.
Multiplicity	1..1
Type	ENUMERATION
Default value	UINT16
Range	UINT8 UINT16 UINT32
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SoAdReportInvPDUHeaderIdToDetEnable
Description	<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"> ▶ true: Report of SOAD_E_INV_PDUHEADER_ID to DET. ▶ false: No report to DET. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (code): Setting this parameter to FALSE reduces the ROM consumption of the module code. ▶ Execution time reduction (code): Setting this parameter to FALSE re- duces the execution time of the module code.
Multiplicity	1..1

Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	SoAdSocketRouteDestFanInEnable	
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"> ▶ true: Global PDUs can be referenced by multiple SoAdSocketRouteDest (SoAdRxUpperLayerType = 'IF'). ▶ false: Global PDUs can only be referenced by one SoAdSocketRouteDest (SoAdRxUpperLayerType = 'IF'). 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.6.1.7. SoAdBswModules

Parameters included	
Parameter name	Multiplicity
SoAdIf	1..1
SoAdTp	1..1
SoAdIfTriggerTransmit	1..1
SoAdIfTxConfirmation	1..1
SoAdSoConModeChg	1..1
SoAdLocallIpAddrAssignmentChg	1..1
SoAdUseCallerInfix	1..1
SoAdUseTypeInfix	1..1
SoAdBswModuleRef	1..1

Parameter Name	SoAdIf
----------------	---------------



Description	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).	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdTp	
Description	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).	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdIfTriggerTransmit	
Description	Specifies if the BSW module supports the TriggerTransmit API or not. If the value is true, the API is supported.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdIfTxConfirmation	
Description	Specifies if the BSW module supports the TxConfirmation API or not. If the value is true, the API is supported.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	AUTOSAR_ECUC
---------------	--------------

Parameter Name	SoAdSoConModeChg	
Description	Specifies if the BSW module supports the SoConModeChg API or not. If the value is true, the API is supported.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdLocallpAddrAssigmentChg	
Description	Specifies if the BSW module supports the LocallpAddrAssigmentChg API or not. If the value is true, the API is supported.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdUseCallerInfix	
Description	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.	
	This parameter is not used.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	



Parameter Name	SoAdUseTypeInfix
Description	<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"> ▶ true: TypeInfix is used. ▶ false: TypeInfix is not used. <p>This parameter is not used.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SoAdBswModuleRef
Description	<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>This parameter is not used.</p>
Multiplicity	1..1
Type	FOREIGN-REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.6.1.8. SoAdConfig

Containers included		
Container name	Multiplicity	Description
SoAdPduRoute	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.
SoAdSocketConnection-Group	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.
SoAdSocketRoute	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.
SoAdRoutingGroup	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.

5.6.1.9. SoAdPduRoute

Containers included

Container name	Multiplicity	Description
SoAdPduRouteDest	1..n	Describes the upper layer destination PDU for a message received on a Tcplp socket.

Parameters included

Parameter name	Multiplicity
SoAdTxPduld	1..1
SoAdTxPduRef	1..1
SoAdTxUpperLayerType	1..1
SoAdTxPduCollectionSemantics	1..1
SoAdSkipIfTxConfirmation	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



Parameter Name	SoAdTxPduRef	
Description	Reference to the global PDU structure.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdTxUpperLayerType	
Description	Specifies the upper layer interface type (must be "IF" in case of multiple PduRoutes).	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	IF	
Range	IF TP	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdTxPduCollectionSemantics	
Description	<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>This parameter shall only be set to SOAD_COLLECT_LAST_IS_BEST if the related upper layer is configured with SoAdIfTriggerTransmit set to TRUE.</p>	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	SOAD_COLLECT_QUEUED	
Range	SOAD_COLLECT_LAST_IS_BEST SOAD_COLLECT_QUEUED	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	SoAdSkipIfTxConfirmation	
Description	<p>Enables or disables the Tx confirmation for the SoAd_IfTransmit() call.</p> <ul style="list-style-type: none"> ▶ TRUE: Tx confirmation gets skipped for this PDU. ▶ FALSE: The Tx confirmation API is called for this PDU if the transmission was successful. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ Run-time reduction: Enabling this parameter reduces the run-time consumption of the module's Tx confirmations. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	FALSE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.6.1.10. SoAdPduRouteDest

Parameters included	
Parameter name	Multiplicity
SoAdTxPduHeaderId	0..1
SoAdTxSocketConnOrSocketConnBundleRef	1..1
SoAdTxRoutingGroupRef	0..n
SoAdTxUdpTriggerMode	1..1
SoAdTxUdpTriggerTimeout	0..1

Parameter Name	SoAdTxPduHeaderId
Description	ID to be sent on the TCP/IP connection if the PDU header option is enabled.
Multiplicity	0..1
Type	INTEGER
Default value	0



Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdTxSocketConnOrSocketConnBundleRef
Description	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.
Multiplicity	1..1
Type	CHOICE-REFERENCE
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

Parameter Name	SoAdTxRoutingGroupRef
Description	Reference to the routing group.
Multiplicity	0..n
Type	REFERENCE
Configuration class	PostBuild:
Origin	AUTOSAR_ECUC

Parameter Name	SoAdTxUdpTriggerMode
Description	<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>This parameter will be enabled if SoAdTxUpperLayerType is IF and referenced SoAdSocketConnectionGroup has SoAdSocketnPduUdpTxBufferMin enabled.</p>
Multiplicity	1..1
Type	ENUMERATION
Default value	TRIGGER_ALWAYS
Range	TRIGGER_ALWAYS TRIGGER_NEVER



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdTxUdpTriggerTimeout	
Description	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.	
Multiplicity	0..1	
Type	FLOAT	
Default value	0.1	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.6.1.11. SoAdSocketConnectionGroup

Containers included		
Container name	Multiplicity	Description
SoAdSocketConnection	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>
SoAdSocketProtocol	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
SoAdPduHeaderEnable	1..1
SoAdResourceManagementEnable	1..1
SoAdSocketLocalAddressRef	1..1
SoAdSocketLocalPort	1..1
SoAdSocketAutomaticSoConSetup	1..1
SoAdSocketSoConModeChgNotification	1..1

**Parameters included**

SoAdSocketIpAddrAssignmentChgNotification	1..1
SoAdSocketTpRxBufferMin	0..1
SoAdSocketMsgAcceptanceFilterEnabled	1..1
SoAdSocketFramePriority	0..1

Parameter Name	SoAdPduHeaderEnable	
Description	Enables the transmission of the PDU header (ID, length) on this socket connection.	
	<ul style="list-style-type: none"> ▶ true: Send PDU header before data. ▶ false: Send data only. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdResourceManagementEnable	
Description	<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"> ▶ true: resource management option enabled. ▶ false: resource management option disabled. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketLocalAddressRef	
Description	Local IP address and interface used for this connection.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketLocalPort	
Description	Local UDP or TCP port used for this connection. If this parameter is set to 0, SoAd requests Tcplp to select an ephemeral port.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketAutomaticSoConSetup	
Description	Specifies if the setup of the socket connection shall be done automatically or manually. <ul style="list-style-type: none"> ▶ true: Setup done automatically. ▶ false: Setup done manually via SoAd_OpenSoCon() and SoAd_CloseSoCon(). 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketSoConModeChgNotification	
Description	Specifies if the SoCon mode change notification callback function of the upper layer shall be called in case of SoCon mode change.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketIpAddrAssignmentChgNotification	



Description	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.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketTpRxBufferMin	
Description	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.	
Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketMsgAcceptanceFilterEnabled	
Description	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.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketFramePriority	



Description	Specifies the priority of the Ethernet frames.	
Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.6.1.12. SoAdSocketConnection

Containers included		
Container name	Multiplicity	Description
SoAdSocketRemoteAddress	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
SoAdSocketId	1..1
SoAdTlsConnectionRef	0..1

Parameter Name	SoAdSocketId	
Description	Socket connection identifier used as SoConId in the interaction with upper layers.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdTlsConnectionRef	
Description	Reference to the TLS connection.	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PostBuild:	VariantPostBuild



Origin	AUTOSAR_ECUC
--------	--------------

5.6.1.13. SoAdSocketRemoteAddress

Parameters included	
Parameter name	Multiplicity
SoAdSocketRemotelpAddress	1..1
SoAdSocketRemotePort	1..1

Parameter Name	SoAdSocketRemotelpAddress
Description	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.
Multiplicity	1..1
Type	STRING
Default value	ANY
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SoAdSocketRemotePort
Description	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.
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.6.1.14. SoAdSocketProtocol

Containers included		
Container name	Multiplicity	Description



Containers included

SoAdSocketUdp	1..1	Specifies that UDP is used as transport protocol for the socket connection group and parameters only related to UDP socket connections.
SoAdSocketTcp	1..1	Specifies that TCP is used as transport protocol for the socket connection group and parameters only related to TCP socket connections

5.6.1.15. SoAdSocketUdp

Parameters included

Parameter name	Multiplicity
SoAdSocketUdpAliveSupervisionTimeout	0..1
SoAdSocketUdpListenOnly	1..1
SoAdSocketUdpRetryEnabled	1..1
SoAdSocketUdpStrictHeaderLenCheckEnabled	1..1
SoAdSocketUdpTriggerTimeout	0..1
SoAdSocketnPduUdpTxBufferMin	0..1
SoAdDatagramTIsConnectionRef	0..1
SoAdTxBufferPoolSize	1..1
SoAdEnableShadowBufferSize	1..1

Parameter Name	SoAdSocketUdpAliveSupervisionTimeout
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	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SoAdSocketUdpListenOnly
Description	Specifies if the socket connection group is used only for reception or used for both reception and transmission.



	<ul style="list-style-type: none"> ▶ true: This UDP port cannot transmit data. ▶ false: This UDP port can send and receive data.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SoAdSocketUdpRetryEnabled
Description	Specifies if a UdpTxBuffer shall be used to retry a UDP transmission in case of TCPIP_E_PHYS_ADDR_MISS or not. <ul style="list-style-type: none"> ▶ true: A UdpTxBuffer is used to retry a UDP transmission. ▶ false: A UDP transmission is not retried, if not successful.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SoAdSocketUdpStrictHeaderLenCheckEnabled
Description	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.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SoAdSocketUdpTriggerTimeout
Description	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.



Multiplicity	0..1	
Type	FLOAT	
Default value	0.1	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketnPduUdpTxBufferMin	
Description	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. 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.	
Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdDatagramTlsConnectionRef	
Description	Reference to the dTLS connection configuration.	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdTxBufferSize	
Description	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.	
	This parameter will be enabled if SoAdSocketnPduUdpTxBufferMin is enabled.	



	Setting this parameter to 0 will disable the buffer pool and create a buffer for each socket connection.
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	PostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	SoAdEnableShadowBufferSize
Description	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.
Multiplicity	1..1
Type	INTEGER
Default value	200
Configuration class	PostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.6.1.16. SoAdSocketTcp

Parameters included	
Parameter name	Multiplicity
SoAdSocketTcpImmediateTpTxConfirmation	1..1
SoAdSocketTcpInitiate	1..1
SoAdSocketTcpKeepAlive	1..1
SoAdSocketTcpKeepAliveInterval	0..1
SoAdSocketTcpKeepAliveProbesMax	0..1

**Parameters included**

SoAdSocketTcpKeepAliveTime	0..1
SoAdSocketTcpNoDelay	0..1
SoAdSocketTcpTxQuota	0..1

Parameter Name	SoAdSocketTcplImmediateTpTxConfirmation	
Description	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.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketTcplInitiate	
Description	Specifies the initiator for this TCP connection. <ul style="list-style-type: none"> ▶ true: This TCP connection is initiated by this module. ▶ false: This TCP connection is to be initiated in the listen mode. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketTcpKeepAlive	
Description	Specifies to use the keep-alive mechanism for this connection. It will not be defined for UDP sockets.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketTcpKeepAliveInterval	
	0..1	



Description	Specifies the interval in seconds between subsequent keep-alive probes.	
Multiplicity	0..1	
Type	FLOAT	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketTcpKeepAliveProbesMax	
Description	Maximum number of times that TCP retransmits an individual data segment before aborting the connection.	
Multiplicity	0..1	
Type	INTEGER	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketTcpKeepAliveTime	
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	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdSocketTcpNoDelay	
Description	Specifies not to use the congestion control mechanism for this connection. It will not be defined for UDP sockets. ▶ true: This TCP connection will NOT use congestion control. ▶ false: This TCP connection will use congestion control. If the optional parameter is not enabled, the default behavior configured for Tcplp via the parameter TcplpTcpNagleEnabled is applied. Note: This parameter must not be set to FALSE if TcplpTcpNagleEnabled is set to FALSE.	
Multiplicity	0..1	
Type	BOOLEAN	



Default value	true
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SoAdSocketTcpTxQuota
Description	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. Rationale: prohibits that a socket connection consumes all available transmit buffers at the Tcplp and blocks transmissions via other socket connections. If the optional parameter is not enabled, the amount of data is not limited.
Multiplicity	0..1
Type	INTEGER
Default value	0
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.6.1.17. SoAdSocketRoute

Containers included		
Container name	Multiplicity	Description
SoAdSocketRouteDest	1..n	Describes the upper layer destination PDU for a message received on a Tcplp socket.

Parameters included	
Parameter name	Multiplicity
SoAdRxPduHeaderId	0..1
SoAdRxSocketConnOrSocketConnBundleRef	1..1

Parameter Name	SoAdRxPduHeaderId
Description	ID contained in the packet received on the TCP/IP connection if the PDU header option is enabled.
Multiplicity	0..1
Type	INTEGER



Default value	0
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SoAdRxSocketConnOrSocketConnBundleRef	
Description	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.	
Multiplicity	1..1	
Type	CHOICE-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.6.1.18. SoAdSocketRouteDest

Parameters included	
Parameter name	Multiplicity
SoAdRxPduRef	1..1
SoAdRxPduld	0..1
SoAdRxUpperLayerType	1..1
SoAdRxRoutingGroupRef	0..n

Parameter Name	SoAdRxPduRef	
Description	Reference to the global PDU structure.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	SoAdRxPduld	
Description	This unique identifier is used for a receive cancellation request from an upper layer of the SoAd.	
Multiplicity	0..1	



Type	INTEGER
Default value	0
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SoAdRxUpperLayerType
Description	Specifies the upper layer interface type (must be "IF" in case of multiple RxPdus).
Multiplicity	1..1
Type	ENUMERATION
Default value	IF
Range	IF TP
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SoAdRxRoutingGroupRef
Description	Reference to the routing group.
Multiplicity	0..n
Type	REFERENCE
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.6.1.19. SoAdRoutingGroup

Parameters included	
Parameter name	Multiplicity
SoAdRoutingGroupId	1..1
SoAdRoutingGroupsEnabledAtInit	1..1
SoAdRoutingGroupTxTriggerable	1..1

Parameter Name	SoAdRoutingGroupId
Description	Unique ID of the Routing Group.
Multiplicity	1..1



Type	INTEGER
Default value	0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SoAdRoutingGroupsEnabledAtInit
Description	If set to true, this routing group is enabled after initializing the SoAd module (i.e. enabled in the SoAd_Init function).
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	SoAdRoutingGroupTxTriggerable
Description	Specifies the IF-TxPDUs related to the PduRouteDest containers.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.6.2. Recommended configurations

5.6.2.1. SoAdRecConfigurationDflt

Containers included	
Container name	Container definition
PduR	SoAdBswModules
Sd	SoAdBswModules
DolP	SoAdBswModules



Parameters included

Parameter name	Value
----------------	-------

5.6.2.1.1. PduR

Parameters included

Parameter name	Value
SoAdIf	true
SoAdTp	true
SoAdIfTriggerTransmit	false
SoAdIfTxConfirmation	true
SoAdSoConModeChg	false
SoAdLocallpAddrAssigmentChg	false
SoAdUseCallerInfix	true
SoAdUseTypeInfix	true

5.6.2.1.2. Sd

Parameters included

Parameter name	Value
SoAdIf	true
SoAdTp	false
SoAdIfTriggerTransmit	false
SoAdIfTxConfirmation	false
SoAdSoConModeChg	false
SoAdLocallpAddrAssigmentChg	true
SoAdUseCallerInfix	true
SoAdUseTypeInfix	true

5.6.2.1.3. DoIP

Parameters included

Parameter name	Value
----------------	-------



Parameters included

SoAdIf	true
SoAdTp	true
SoAdIfTriggerTransmit	false
SoAdIfTxConfirmation	true
SoAdSoConModeChg	false
SoAdLocallpAddrAssigmentChg	true
SoAdUseCallerInfix	true
SoAdUseTypeInfix	true

5.6.3. Application programming interface (API)

5.6.3.1. Type definitions

5.6.3.1.1. SoAd_DtlsConnectionStateType

Purpose	DTLS connection state type.
Type	uint8

5.6.3.2. Macro constants

5.6.3.2.1. SOAD_CHANGEPARAMETER_SVCID

Purpose	Defines the API ID of function SoAd_ChangeParameter() .
Value	0xF1U

5.6.3.2.2. SOAD_CLOSESOCON_SVCID

Purpose	Defines the API ID of function SoAd_CloseSoCon() .
Value	0x09U



5.6.3.2.3. SOAD_COPYTXDATA_SVCID

Purpose	Defines the API ID of function SoAd_CopyTxData() .
Value	0x13U

5.6.3.2.4. SOAD_DISABLEROUTING_SVCID

Purpose	Defines the API ID of function SoAd_DisableRouting() .
Value	0x0FU

5.6.3.2.5. SOAD_DISABLESPECIFICROUTING_SVCID

Purpose	Defines the API ID of function SoAd_DisableSpecificRouting() .
Value	0x21U

5.6.3.2.6. SOAD_DTLSSTATECHG_SVCID

Purpose	Defines the API ID of function SoAd_DtlsStateChg() .
Value	0xF4U

5.6.3.2.7. SOAD_DTLS_STATE_CLOSED

Purpose	DTLS connection state CLOSED.
Value	0U

5.6.3.2.8. SOAD_DTLS_STATE_CONNECTED

Purpose	DTLS connection state CONNECTED.
Value	2U

5.6.3.2.9. SOAD_DTLS_STATE_CONNECTING

Purpose	DTLS connection state CONNECTING.
Value	1U



5.6.3.2.10. SOAD_ENABLEROUTING_SVCID

Purpose	Defines the API ID of function SoAd_EnableRouting() .
Value	0x0EU

5.6.3.2.11. SOAD_ENABLESPECIFICROUTING_SVCID

Purpose	Defines the API ID of function SoAd_EnableSpecificRouting() .
Value	0x20U

5.6.3.2.12. SOAD_E_INV_ARG

Purpose	API requests called with invalid argument.
Value	0x03U

5.6.3.2.13. SOAD_E_INV_LENGTH_IN_PDUHEADER

Purpose	Error - Length in PduHeader exceeds the possible maximum.
Value	0xFDU

5.6.3.2.14. SOAD_E_INV_METADATA

Purpose	Error - Invalid meta data.
Value	0x09U

5.6.3.2.15. SOAD_E_INV_PDUHEADER_ID

Purpose	Error - Unknown PduHeader ID.
Value	0x05U

5.6.3.2.16. SOAD_E_INV_PDUID

Purpose	Error - Unknown PDU ID.
---------	-------------------------



Value	0x06U
--------------	-------

5.6.3.2.17. SOAD_E_INV_SOCKETID

Purpose	Error - Unknown socket address.
Value	0x07U

5.6.3.2.18. SOAD_E_NOBUFS

Purpose	Error - No buffer space available.
Value	0x04U

5.6.3.2.19. SOAD_E_NOTINIT

Purpose	API service called before initializing the module.
Value	0x01U

5.6.3.2.20. SOAD_E_PARAM_POINTER

Purpose	API requests called with NULL pointer.
Value	0x02U

5.6.3.2.21. SOAD_E_TRIGGERTXBUF

Purpose	Error - Insufficient TriggerTransmitBuffer.
Value	0xFEU

5.6.3.2.22. SOAD_E_TWO_PDUS_SAME_SOCON

Purpose	Error - Two PDUs cannot be configured for the same socket connection simultaneously.
Value	0xFBUs



5.6.3.2.23. SOAD_E_TWO_SOCON_SAME_PDU

Purpose	Error - Two socket connections cannot use the same PDU within the same Socon-group.
Value	0xFCU

5.6.3.2.24. SOAD_GETANDRESETMEASUREMENTDATA_SVCID

Purpose	Defines the API ID of function SoAd_GetAndResetMeasurementData() .
Value	0x45U

5.6.3.2.25. SOAD_GETLOCALADDR_SVCID

Purpose	Defines the API ID of function SoAd_GetLocalAddr() .
Value	0x0CU

5.6.3.2.26. SOAD_GETPHYSADDR_SVCID

Purpose	Defines the API ID of function SoAd_GetPhysAddr() .
Value	0x0DU

5.6.3.2.27. SOAD_GETREMOTEADDR_SVCID

Purpose	Defines the API ID of function SoAd_GetRemoteAddr() .
Value	0x1CU

5.6.3.2.28. SOAD_GETSOCONID_SVCID

Purpose	Defines the API ID of function SoAd_GetSoConId() .
Value	0x07U

5.6.3.2.29. SOAD_GETSOCONMODE_SVCID

Purpose	Defines the API ID of function SoAd_GetSoConMode() .
----------------	--



Value	0xF0U
--------------	-------

5.6.3.2.30. SOAD_GETVERSIONINFO_SVCID

Purpose	Defines the API ID of function SoAd_GetVersionInfo() .
Value	0x02U

5.6.3.2.31. SOAD_IFROUTINGGROUPTRANSMIT_SVCID

Purpose	Defines the API ID of function SoAd_IfRoutingGroupTransmit() .
Value	0x1DU

5.6.3.2.32. SOAD_IFSPECIFICROUTINGGROUPTRANSMIT_SVCID

Purpose	Defines the API ID of function SoAd_IfSpecificRoutingGroupTransmit() .
Value	0x1FU

5.6.3.2.33. SOAD_IFTRANSMIT_SVCID

Purpose	Defines the API ID of function SoAd_IfTransmit() .
Value	0x03U

5.6.3.2.34. SOAD_INIT_SVCID

Purpose	Defines the API ID of function SoAd_Init() .
Value	0x01U

5.6.3.2.35. SOAD_INSTANCE_ID

Purpose	Defines the instance number of this Socket Adaptor. Because multiple instances of the Socket Adaptor are not supported, the instance ID is always zero.
Value	0U



5.6.3.2.36. SOAD_INTERNAL_SVCID

Purpose	Defines the API ID of internal functions.
Value	0xFFU

5.6.3.2.37. SOAD_INVALID_ULFUNCID

Purpose	Invalid upper layer function array index.
Value	0xFFU

5.6.3.2.38. SOAD_ISCONNECTIONREADY_SVCID

Purpose	Defines the API ID of function SoAd_IsConnectionReady() .
Value	0xF3U

5.6.3.2.39. SOAD_LOCALIPADDRASSIGNMENTCHG_SVCID

Purpose	Defines the API ID of function SoAd_LocallpAddrAssignmentChg() .
Value	0x18U

5.6.3.2.40. SOAD_MAINFUNCTIONTX_SVCID

Purpose	Defines the API ID of function SoAd_MainFunctionTx() .
Value	0xF2U

5.6.3.2.41. SOAD_MAINFUNCTION_SVCID

Purpose	Defines the API ID of function SoAd_MainFunction() .
Value	0x19U

5.6.3.2.42. SOAD_OPENSOCON_SVCID

Purpose	Defines the API ID of function SoAd_OpenSoCon() .
---------	---



Value	0x08U
--------------	-------

5.6.3.2.43. SOAD_READDHCPHOSTNAMEOPTION_SVCID

Purpose	Define API id of function SoAd_ReadDhcpHostNameOption() .
Value	0x1AU

5.6.3.2.44. SOAD_RELEASEIPADDRASSIGNMENT_SVCID

Purpose	Defines the API ID of function SoAd_ReleaselpAddrAssignment() .
Value	0x0BU

5.6.3.2.45. SOAD_RELEASEREMOTEADDR_SVCID

Purpose	Defines the API ID of function SoAd_ReleaseRemoteAddr() .
Value	0x23U

5.6.3.2.46. SOAD_REQUESTIPADDRASSIGNMENT_SVCID

Purpose	Defines the API ID of function SoAd_RequestIpAddrAssignment() .
Value	0x0AU

5.6.3.2.47. SOAD_RXINDICATION_SVCID

Purpose	Defines the API ID of function SoAd_TcpIpRxIndication().
Value	0x12U

5.6.3.2.48. SOAD_SETREMOTEADDR_SVCID

Purpose	Defines the API ID of function SoAd_SetRemoteAddr() .
Value	0x10U



5.6.3.2.49. SOAD_SETUNIQUEREMOTEADDR_SVCID

Purpose	Defines the API ID of function SoAd_SetUniqueRemoteAddr() .
Value	0x1EU

5.6.3.2.50. SOAD_TCPCACCEPTED_SVCID

Purpose	Defines the API ID of function SoAd_TcpAccepted() .
Value	0x15U

5.6.3.2.51. SOAD_TCPCONNECTED_SVCID

Purpose	Defines the API ID of function SoAd_TcpConnected() .
Value	0x16U

5.6.3.2.52. SOAD_TCPIPEVENT_SVCID

Purpose	Defines the API ID of function SoAd_TcpIpEvent() .
Value	0x17U

5.6.3.2.53. SOAD_TPCANCELRECEIVE_SVCID

Purpose	Defines the API ID of function SoAd_CancelReceive() .
Value	0x06U

5.6.3.2.54. SOAD_TPCANCELTRANSMIT_SVCID

Purpose	Defines the API ID of function SoAd_CancelTransmit() .
Value	0x05U

5.6.3.2.55. SOAD_TPCHANGEPARAMETER_SVCID

Purpose	Defines the API ID of function SoAd_TpChangeParameter() .
Value	0x11U



5.6.3.2.56. SOAD_TPTRANSMIT_SVCID

Purpose	Defines the API ID of function SoAd_TpTransmit() .
Value	0x04U

5.6.3.2.57. SOAD_TXCONFIRMATION_SVCID

Purpose	Defines the API ID of function SoAd_TxConfirmation() .
Value	0x14U

5.6.3.2.58. SOAD_WROTEDHCPPHOSTNAMEOPTION_SVCID

Purpose	Define API id of function SoAd_WriteDhcpHostNameOption() .
Value	0x1BU

5.6.3.2.59. SoAdIf_Transmit

Purpose	
Value	SoAd_IfTransmit

5.6.3.2.60. SoAdTp_CancelReceive

Purpose	
Value	SoAd_TpCancelReceive

5.6.3.2.61. SoAdTp_CancelTransmit

Purpose	
Value	SoAd_TpCancelTransmit

5.6.3.2.62. SoAdTp_Transmit

Purpose	
----------------	--



Value	SoAd_TpTransmit
--------------	-----------------

5.6.3.3. Functions

5.6.3.3.1. SoAd_ChangeParameter

Purpose	Change a parameter of the TCP/IP stack.	
Synopsis	Std_ReturnType SoAd_ChangeParameter (SoAd_SoConIdType SoConId , uint8 ParameterId , uint8 * ParameterValuePtr);	
Service ID	0x11	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SoConIds. Non reentrant for the same SoConId.	
Parameters (in)	SoConId	Socket connection index specifying the socket connection for which the parameter shall be changed
	ParameterId	Identifier of the parameter to be changed
	ParameterValuePtr	Pointer to the memory containing the new parameter value
Return Value	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
Description	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.	

5.6.3.3.2. SoAd_CloseSoCon

Purpose	Close a socket connection.
Synopsis	Std_ReturnType SoAd_CloseSoCon (SoAd_SoConIdType SoConId , boolean Abort);
Service ID	0x09
Sync/Async	Asynchronous
Reentrancy	Reentrant



Parameters (in)	SoConId	Socket connection index specifying the socket connection to be closed
	Abort	<ul style="list-style-type: none"> ▶ TRUE: Socket connection is immediately terminated. ▶ FALSE: Socket connection is terminated if no other upper layer uses this socket connection.
Return Value	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
Description	This service closes the socket connection specified by the SoConId.	

5.6.3.3.3. SoAd_CopyTxData

Purpose	Copy transmission data.	
Synopsis	BufReq_ReturnType SoAd_CopyTxData (TcpIp_SocketIdType SocketId , uint8 * BufPtr , uint16 BufLength);	
Service ID	0x13	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	SocketId	Socket handle identifying the local socket resource
	BufPtr	Pointer to buffer for transmission data
	BufLength	Length of provided data buffer
Return Value		
Description	This service requests to copy data for transmission to the buffer indicated. This call is triggered by Tcplp_Transmit(). Note: The call to SoAd_CopyTxData() may happen in the context of Tcplp_Transmit().	

5.6.3.3.4. SoAd_DisableRouting

Purpose	Disable routing of a group of PDUs.
Synopsis	Std_ReturnType SoAd_DisableRouting (SoAd_RoutingGroupIdType Id);



Service ID	0x0F	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	<code>Id</code>	Identification of the routing group
Return Value	<code>Std_ReturnType</code>	
	<code>E_OK</code>	The request was successful.
	<code>E_NOT_OK</code>	The request was not successful.
Description	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.	

5.6.3.3.5. `SoAd_DisableSpecificRouting`

Purpose	Disable routing of a group of PDUs on a specific socket connection.	
Synopsis	<code>Std_ReturnType SoAd_DisableSpecificRouting (SoAd_RoutingGroupType Id , SoAd_SoConIdType SoConId);</code>	
Service ID	0x21	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	<code>Id</code>	Identification of the routing group
	<code>SoConId</code>	Identification of the socket connection
Return Value	<code>Std_ReturnType</code>	
	<code>E_OK</code>	The request was successful.
	<code>E_NOT_OK</code>	The request was not successful.
Description	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.	

5.6.3.3.6. `SoAd_DtlsConnectionStateChg`

Purpose	
----------------	--



Synopsis	<code>void SoAd_DtlsConnectionStateChg (TcpIp_SocketIdType SocketId , const TcpIp_SockAddrType * RemoteAddrPtr , SoAd_DtlsConnectionStateType State);</code>
-----------------	--

5.6.3.3.7. SoAd_EnableRouting

Purpose	Enable routing of a group of PDUs.	
Synopsis	<code>Std_ReturnType SoAd_EnableRouting (SoAd_RoutingGroupIdType Id);</code>	
Service ID	0x0E	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	<code>Id</code>	Identification of the routing group
Return Value	Std_ReturnType E_OK The request was successful. E_NOT_OK The request was not successful.	
Description	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.	

5.6.3.3.8. SoAd_EnableSpecificRouting

Purpose	Enable routing of a group of PDUs on a specific socket connection.	
Synopsis	<code>Std_ReturnType SoAd_EnableSpecificRouting (SoAd_RoutingGroupIdType Id , SoAd_SoConIdType SoConId);</code>	
Service ID	0x20	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	<code>Id</code>	Identification of the routing group
	<code>SoConId</code>	Identification of the socket connection
Return Value	Std_ReturnType E_OK The request was successful. E_NOT_OK The request was not successful.	



Description	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.
--------------------	---

5.6.3.3.9. SoAd_GetAndResetMeasurementData

Purpose	Reads and resets measurement data.	
Synopsis	<pre>Std_ReturnType SoAd_GetAndResetMeasurementData (SoAd_MeasurementIdxType MeasurementIdx , boolean MeasurementResetNeeded , uint32 * MeasurementDataPtr);</pre>	
Parameters (in)	MeasurementIdx	<p>Index to select specific measurement data:</p> <ul style="list-style-type: none"> ▶ SOAD_MEAS_DROP_TCP (0x01) - Measurement index of dropped PDUs caused by invalid destination TCP port. ▶ SOAD_MEAS_DROP_UDP (0x02) - Measurement index of dropped PDUs caused by invalid destination UDP port. ▶ SOAD_MEAS_RESERVED_1 (0x03-0x7F) - Reserved by AUTOSAR. ▶ SOAD_MEAS_RESERVED_2 (0x80-0xEF) - Vendor-specific range. ▶ SOAD_MEAS_RESERVED_3 (0xF0-0xFE) - Reserved by AUTOSAR (future use). ▶ SOAD_MEAS_ALL (0xFF) - Represents all measurement indexes.
	MeasurementResetNeeded	Flag to trigger a reset of the measurement data
Parameters (out)	MeasurementDataPtr	Pointer to the data buffer where to copy measurement data
Return Value	Std_ReturnType	



	E_OK	The function was successfully executed.
	E_NOT_OK	The function was not successfully executed.
Description	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.	

5.6.3.3.10. SoAd_GetLocalAddr

Purpose	Get the current local IP address.	
Synopsis	<pre>Std_ReturnType SoAd_GetLocalAddr (SoAd_SoConIdType SoConId , TcpIp_SockAddrType * LocalAddrPtr , uint8 * NetmaskPtr , TcpIp_SockAddrType * DefaultRouterPtr);</pre>	
Service ID	0x0C	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SoConId	Socket connection index representing the SoAd socket connection for which the actual local IP address shall be obtained
Parameters (out)	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)
Return Value	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
Description	Obtains the local address (IP address and port) actually used for the SoAd socket connection specified by the SoConId, the netmask, and default router.	



5.6.3.3.11. SoAd_GetPhysAddr

Purpose	Get the physical IP address.	
Synopsis	<code>Std_ReturnType SoAd_GetPhysAddr (SoAd_SoConIdType SoConId , uint8 * PhysAddrPtr);</code>	
Service ID	0x0D	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SoConId	Socket connection index representing the SoAd socket connection for which the physical source address of the related Ethlf controller shall be obtained
Parameters (out)	PhysAddrPtr	Pointer to the memory where the physical source address (MAC address) in network byte order is stored
Return Value	Std_ReturnType E_OK The request was successful. E_NOT_OK The request was not successful.	
Description	Obtains the physical source address of the Ethlf controller used by the SoAd socket connection specified by the SoConId.	

5.6.3.3.12. SoAd_GetRemoteAddr

Purpose	Get the remote address of a socket connection.	
Synopsis	<code>Std_ReturnType SoAd_GetRemoteAddr (SoAd_SoConIdType SoConId , TcpIp_SockAddrType * IpAddrPtr);</code>	
Service ID	0x1C	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SoConId	Socket connection index representing the SoAd socket connection from which the remote address shall be obtained
Parameters (out)	IpAddrPtr	Pointer to the memory where the remote address is stored
Return Value	Std_ReturnType E_OK The request was successful.	



	E_NOT_OK	The request was not successful.
Description	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.	

5.6.3.3.13. SoAd_GetSoConId

Purpose	Get socket connection ID for given PDU ID.	
Synopsis	Std_ReturnType SoAd_GetSoConId (PduIdType TxPduId , SoAd_SoConIdType * SoConIdPtr);	
Service ID	0x07	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	TxPduId	Transmit the PDU ID specifying the SoAd socket connection for which the socket connection index shall be returned.
Parameters (out)	SoConIdPtr	Pointer to the memory receiving the socket connection index
Return Value	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
Description	Returns the socket connection index related to the specified transmit PDU ID. In case a fan-out is configured for TxPduld (i.e. multiple SoAdPduRouteDest specified), E_NOT_OK shall be returned.	

5.6.3.3.14. SoAd_GetSoConMode

Purpose	Get socket connection mode for the given socket connection index.	
Synopsis	Std_ReturnType SoAd_GetSoConMode (SoAd_SoConIdType SoConId , SoAd_SoConModeType * ModePtr);	
Service ID	0xF0	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SoConId	Index of the socket connection for which the mode shall be returned



Parameters (out)	ModePtr	Pointer to the memory where the socket connection mode shall be stored
Return Value	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
Description	Returns the socket connection mode related to the specified socket connection.	

5.6.3.3.15. SoAd_GetVersionInfo

Purpose	Get version information of the SoAd module.	
Synopsis	<code>void SoAd_GetVersionInfo (Std_VersionInfoType * Versioninfo);</code>	
Service ID	0x02	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (out)	Versioninfo	Pointer to where to store the version information of this module
Description	This service returns the version information of this module. The version information includes: <ul style="list-style-type: none"> ▶ Module ID ▶ Vendor ID ▶ Vendor-specific version numbers 	

5.6.3.3.16. SoAd_IfRoutingGroupTransmit

Purpose	Initiate transmission of IF-TxPDUs belonging to a given routing group.	
Synopsis	<code>Std_ReturnType SoAd_IfRoutingGroupTransmit (SoAd_RoutingGroupType Id);</code>	
Service ID	0x1D	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	Id	Identification of the routing group
Return Value	Std_ReturnType	



	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
Description	Triggers the transmission of all IF-TxPDUs identified by the parameter Id after requesting the data from the related upper layer.	

5.6.3.3.17. SoAd_IfSpecificRoutingGroupTransmit

Purpose	Initiate transmission of IF-TxPDUs belonging to a given routing group on a specific socket connection.	
Synopsis	Std_ReturnType SoAd_IfSpecificRoutingGroupTransmit (SoAd_RoutingGroupIdType Id , SoAd_SoConIdType SoConId);	
Service ID	0x1F	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	Id	Identification of the routing group
	SoConId	Identification of the socket connection
Return Value	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
Description	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.	

5.6.3.3.18. SoAd_IfTransmit

Purpose	Transfer L-PDU.	
Synopsis	Std_ReturnType SoAd_IfTransmit (PduIdType SoAdSrcPduId , const PduInfoType * SoAdSrcPduInfoPtr);	
Service ID	0x03	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	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
Return Value	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).
Description	This service is used to request the transfer of L-PDU.	

5.6.3.3.19. SoAd_Init

Purpose	Initialize the SoAd module.	
Synopsis	<code>void SoAd_Init (const SoAd_ConfigType * SoAdConfigPtr);</code>	
Service ID	0x01	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	SoAdConfigPtr	Points to the implementation-specific structure
Description	This function initializes the SoAd module.	

5.6.3.3.20. SoAd_IsConnectionReady

Purpose	Checks if ARP entry and IpSec SA exist in Tcplp.	
Synopsis	<code>TcpIp_ReturnType SoAd_IsConnectionReady (SoAd_SoConIdType SoConId , const TcpIp_SockAddrType * RemoteAddrPtr);</code>	
Service ID	0x1F	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	SoConId	- Identification of the socket connection
	RemoteAddrPtr	- Pointer to the remote address



Return Value	Tcplp_ReturnType
	TCPPIP_E_PENDING ARP entry, IpSec SA or (D)TLS connection establishment in progress.
	TCPPIP_E_NOT_OK Transmission not allowed/possible.
	TCPPIP_OK ARP and IpSec SA exist, data can be sent
Description	Calls Tcplp with the correct SocketId and RemoteAddr of the corresponding SoConId. Passes the return value of Tcplp to the upper layer.

5.6.3.3.21. SoAd_IsValidConfig

Purpose	Checks the compatibility of the post-build configuration.	
Synopsis	<code>Std_ReturnType SoAd_IsValidConfig (const void * SoAdConfigPtr);</code>	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	SoAdConfigPtr	Pointer to the configuration data of the SoAd module.
Return Value		
Description	This service checks the compatibility of the post-build configuration against the source code.	

5.6.3.3.22. SoAd_LocallipAddrAssignmentChg

Purpose	Indicates an IP address change.	
Synopsis	<code>void SoAd_LocallipAddrAssignmentChg (TcpIp_LocalAddrIdType IpAddrId , TcpIp_IpAddrStateType State);</code>	
Service ID	0x18	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	IpAddrId	IP address Identifier, representing an IP address specified in the Tcplp module configuration (e.g. static IPv4 address on EthIf controller 0).
	State	State of IP address assignment:



	<ul style="list-style-type: none"> ▶ TCPIP_IPADDR_STATE_ASSIGNED ▶ TCPIP_IPADDR_STATE_ONHOLD ▶ TCPIP_IPADDR_STATE_UNASSIGNED
Description	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).

5.6.3.3.23. SoAd_MainFunction

Purpose	Main function of the SoAd.
Synopsis	<code>void SoAd_MainFunction (void);</code>
Service ID	0x19
Sync/Async	Synchronous
Reentrancy	Non reentrant
Description	This function is the main function for scheduling SoAd.

5.6.3.3.24. SoAd_MainFunctionTx

Purpose	Main function containing only the transmit part of the SoAd.
Synopsis	<code>void SoAd_MainFunctionTx (void);</code>
Service ID	0xF2
Sync/Async	Synchronous
Reentrancy	Non reentrant
Description	Transmit part of the main function if enabled to be external

5.6.3.3.25. SoAd_OpenSoCon

Purpose	Open a socket connection.
Synopsis	<code>Std_ReturnType SoAd_OpenSoCon (SoAd_SoConIdType SoConId);</code>
Service ID	0x08
Sync/Async	Asynchronous



Reentrancy	Reentrant	
Parameters (in)	SoConId	Socket connection index specifying the socket connection to be opened
Return Value	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
Description	This service opens the socket connection specified by the SoConId.	

5.6.3.3.26. SoAd_ReadDhcpHostNameOption

Purpose	Read the DHCP host name.	
Synopsis	Std_ReturnType SoAd_ReadDhcpHostNameOption (SoAd_SoConIdType SoConId , uint8 * Length , uint8 * Data);	
Service ID	0x1A	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SoConIds. Non reentrant for the same SoConId.	
Parameters (in)	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.
Return Value	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 read the currently configured host name option in the DHCP submodule of the TCP/IP stack.	

5.6.3.3.27. SoAd_ReleaseIpAddrAssignment

Purpose	Release IP address assignment.
----------------	--------------------------------



Synopsis	<code>Std_ReturnType SoAd_ReleaseIpAddrAssignment (SoAd_SoConIdType SoConId);</code>	
Service ID	0x0B	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	SoConId	Socket connection index specifying the socket connection for which the IP address shall be released
Return Value	Std_ReturnType E_OK The request was accepted. E_NOT_OK The request was not accepted.	
Description	This API service releases the local IP address assignment used for the socket connection specified by the SoConId.	

5.6.3.3.28. SoAd_ReleaseRemoteAddr

Purpose	Release the remote address of a socket connection.	
Synopsis	<code>void SoAd_ReleaseRemoteAddr (SoAd_SoConIdType SoConId);</code>	
Service ID	0x23	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SoConIds. Non reentrant for the same SoConId.	
Parameters (in)	SoConId	Index of the socket connection for which the remote address shall be released
Description	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.	

5.6.3.3.29. SoAd_RequestIpAddrAssignment

Purpose	Request IP address assignment.	
Synopsis	<code>Std_ReturnType SoAd_RequestIpAddrAssignment (SoAd_SoConIdType SoConId , TcpIp_IpAddrAssignmentType Type , TcpIp_SockAddrType * LocalIpAddrPtr);</code>	
Service ID	0x0A	
Sync/Async	Asynchronous	



Reentrancy	Reentrant for different SoConlds. Non reentrant for the same SoConld.	
Parameters (in)	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 SoConld. Note: This parameter is only used if the parameter Type is set to TCPIP_IPADDR_ASSIGNMENT_STATIC.
Return Value	Std_ReturnType	
	E_OK	The request was accepted.
	E_NOT_OK	The request was not accepted.
Description	This API service initiates the local IP address assignment to be used for the socket connection specified by the SoConld.	

5.6.3.3.30. SoAd_RxIndication

Purpose	Data reception of UDP datagram or TCP stream.	
Synopsis	<pre>void SoAd_RxIndication (TcpIp_SocketIdType SocketId , const TcpIp_SockAddrType * RemoteAddrPtr , uint8 * BufPtr , uint16 Length);</pre>	
Service ID	0x12	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	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
Description	Indicates the reception of a new TCP segment or UDP datagram and provides all data via BufPtr.	



5.6.3.3.31. SoAd_SetRemoteAddr

Purpose	Set the remote address of a socket connection.	
Synopsis	<pre>Std_ReturnType SoAd_SetRemoteAddr (SoAd_SoConIdType SoConId , const TcpIp_SockAddrType * IpAddrPtr);</pre>	
Service ID	0x10	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SoConIds. Non reentrant for the same SoConId.	
Parameters (in)	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
Return Value	Std_ReturnType	
	E_OK	The request was successful.
	E_NOT_OK	The request was not successful.
Description	This API service sets the remote address (IP address and port) of the specified socket connection.	

5.6.3.3.32. SoAd_SetUniqueRemoteAddr

Purpose	Returns the index of a socket connection where the given remote address is set.	
Synopsis	<pre>Std_ReturnType SoAd_SetUniqueRemoteAddr (SoAd_SoConIdType So- ConId , const TcpIp_SockAddrType * RemoteAddrPtr , SoAd_So- ConIdType * AssignedSoConIdPtr);</pre>	
Service ID	0x1e	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SoConGroups. Non reentrant for the same SoConGroup.	
Parameters (in)	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



Parameters (out)	AssignedSoConIdPtr	Pointer to the SoAd_SoConIdType where the index of the socket connection configured with the remote address (IpAddrPtr) shall be stored
Return Value	Std_ReturnType	
	E_OK	The request was accepted.
	E_NOT_OK	The request was rejected, AssignedSoConIdPtr remains unchanged.
Description	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.	

5.6.3.3.33. SoAd_TcpAccepted

Purpose	Indicates an incoming TCP connection on a server socket.	
Synopsis	<pre>Std_ReturnType SoAd_TcpAccepted (TcpIp_SocketIdType SocketId , TcpIp_SocketIdType SocketIdConnected , const TcpIp_SockAddrType * RemoteAddrPtr);</pre>	
Service ID	0x15	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	SocketId	Socket handle identifying the local socket resource
	SocketIdConnected	Socket handle for the established connection
	RemoteAddrPtr	IP address and port of the remote host
Return Value	Result of operation	
	E_OK	SoAd accepts the established connection.
	E_NOT_OK	SoAd refuses the established connection, Tcplp stack shall close the connection.
Description	<p>SoAd_TcpAccepted() 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>	



5.6.3.3.34. SoAd_TcpConnected

Purpose	Indicates a successful connection from client side.	
Synopsis	<code>void SoAd_TcpConnected (TcpIp_SocketIdType SocketId);</code>	
Service ID	0x16	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	SocketId	Socket handle identifying the local socket resource
Description	<p>SoAd_TcpConnected() 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 <code>Tcplp-TcpConnect()</code> and a TCP connection is confirmed by the peer.</p>	

5.6.3.3.35. SoAd_TcplpEvent

Purpose	Indicates an event in the TCP/IP stack.	
Synopsis	<code>void SoAd_TcplpEvent (TcpIp_SocketIdType SocketId , TcpIp_EventType Event);</code>	
Service ID	0x17	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	SocketId	Socket handle identifying the local socket resource
	Event	This parameter contains a description of the event just encountered.
Description	The service gets called if the stack encounters a condition described by the values in <code>TcplpEvent</code> .	

5.6.3.3.36. SoAd_TpCancelReceive

Purpose	Cancel reception.
----------------	-------------------



Synopsis	<code>Std_ReturnType SoAd_TpCancelReceive (PduIdType PduId);</code>	
Service ID	0x06	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PDU IDs. Non reentrant for the same PDU ID.	
Parameters (in)	PduId	Identification of the I-PDU to be cancelled
Return Value	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).
Description	Requests cancellation of the reception via TP for a specific I-PDU.	

5.6.3.3.37. SoAd_TpCancelTransmit

Purpose	Cancel transmission.	
Synopsis	<code>Std_ReturnType SoAd_TpCancelTransmit (PduIdType PduId);</code>	
Service ID	0x05	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PDU IDs. Non reentrant for the same PDU ID.	
Parameters (in)	PduId	Identification of the I-PDU to be cancelled
Return Value	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).
Description	Requests cancellation of the transmission via TP for a specific I-PDU.	

5.6.3.3.38. SoAd_TpChangeParameter

Purpose	Change a parameter of the TCP/IP stack.
Synopsis	<code>Std_ReturnType SoAd_TpChangeParameter (PduIdType Id , TPPParameterType Parameter , uint16 Value);</code>



Service ID	0x11	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SoConlds. Non reentrant for the same SoConld.	
Parameters (in)	Id	Identification of the I-PDU that the parameter change shall affect
	Parameter	Identifier of the parameter to be changed
	Value	New parameter value
Return Value	Std_ReturnType	
	E_OK	The request was accepted.
	E_NOT_OK	The request was not accepted.
Description	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.	

5.6.3.3.39. SoAd_TpTransmit

Purpose	Transfer data.	
Synopsis	<pre>Std_ReturnType SoAd_TpTransmit (PduIdType SoAdSrcPduId , const PduInfoType * SoAdSrcPduInfoPtr);</pre>	
Service ID	0x04	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	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.
Return Value	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).
Description	This service is used to request the transfer of data.	



5.6.3.3.40. SoAd_TxConfirmation

Purpose	Acknowledges transmitted data.	
Synopsis	<code>void SoAd_TxConfirmation (TcpIp_SocketIdType SocketId , uint16 Length);</code>	
Service ID	0x14	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	SocketId	Socket handle identifying the local socket resource
	Length	Number of transmitted data bytes
Description	The TCP/IP stack calls this function after the data was acknowledged by the peer for TCP.	

5.6.3.3.41. SoAd_WriteDhcpHostNameOption

Purpose	Write the DHCP host name.	
Synopsis	<code>Std_ReturnType SoAd_WriteDhcpHostNameOption (SoAd_SoConIdType SoConId , uint8 Length , const uint8 * Data);</code>	
Service ID	0x1B	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SoConIds. Non reentrant for the same SoConId.	
Parameters (in)	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
Return Value	<code>Std_ReturnType</code>	
	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

Description	<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">➤ SoAd_RxIndication()➤ SoAd_TxConfirmation()➤ SoAd_TcpAccepted()➤ SoAd_TcpConnected()➤ SoAd_TcplpEvent()➤ SoAd_LocallpAddrAssignmentChg() <p>List of SoAd main functions:</p> <ul style="list-style-type: none">➤ SoAd_MainFunction()➤ SoAd_MainFunctionTx() <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">➤ EthIfEnableRxInterrupt➤ EthCtrlEnableRxInterrupt➤ EthIfEnableTxInterrupt➤ EthCtrlEnableTxInterrupt
--------------------	--



	<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"> ▶ EthIf_MainFunctionRx() ▶ EthIf_MainFunctionTx() ▶ EthIf_MainFunctionState() ▶ Tcplp_MainFunction() ▶ Tcplp_MainFunctionTx() ▶ EthSM_MainFunction() <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>
Rationale	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

Description	<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_INTREQ_SoAd_0001. List of SoAd upper layer functions:</p> <ul style="list-style-type: none"> ▶ SoAd_SetRemoteAddr() ▶ SoAd_SetUniqueRemoteAddr() ▶ SoAd_ReleaseRemoteAddr() ▶ SoAd_IfRoutingGroupTransmit() ▶ SoAd_IfSpecificRoutingGroupTransmit() ▶ SoAd_GetSoConId() ▶ SoAd_CloseSoCon() ▶ SoAd_OpenSoCon() ▶ SoAd_RequestIpAddrAssignment() ▶ SoAd_ReleaseIpAddrAssignment()
--------------------	--



	<ul style="list-style-type: none"> ▶ SoAd_GetLocalAddr() ▶ SoAd_GetPhysAddr() ▶ SoAd_GetRemoteAddr() ▶ SoAd_EnableRouting() ▶ SoAd_EnableSpecificRouting() ▶ SoAd_DisableRouting() ▶ SoAd_DisableSpecificRouting() ▶ SoAd_TpChangeParameter() ▶ SoAd_ReadDhcpHostNameOption() ▶ SoAd_WriteDhcpHostNameOption() ▶ SoAd_GetSoConMode() ▶ SoAd_ChangeParameter() ▶ SoAd_IsConnectionReady() <p>Exception: The following functions are excluded:</p> <ul style="list-style-type: none"> ▶ SoAd_IfTransmit() ▶ SoAd_TpTransmit() ▶ SoAd_TpCancelTransmit() ▶ SoAd_TpCancelReceive() <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>
Rationale	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

Description	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.
--------------------	---



Rationale	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

Description	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

Description	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

Description	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

Description	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

Description	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

Description	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

Description	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. SomelpTp

5.7.1. Configuration parameters

Containers included		
Container name	Multiplicity	Description
CommonPublishedInformation	1..1	Label: Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.

**Containers included**

PublishedInformation	1..1	Label: EB Published Information Additional published parameters not covered by CommonPublishedInformation container.
SomeIpTpChannel	1..n	This container contains the configuration parameters of the SomeIpTp channel.
SomeIpTpUnSupportedChannel	1..1	
SomeIpTpGeneral	1..1	This container contains the general configuration parameters of the SomeIpTp module.

Parameters included

Parameter name	Multiplicity
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT	
Label	Config Variant	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	VariantLinkTime	
Range	VariantPreCompile	
Configuration class	VariantLinkTime:	VariantLinkTime
	VariantPostBuild:	VariantPostBuild
	VariantPreCompile:	VariantPreCompile

5.7.1.1. CommonPublishedInformation**Parameters included**

Parameter name	Multiplicity
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1



Parameters included

SwPatchVersion	1..1
ModuleId	1..1
VendorId	1..1
Release	1..1

Parameter Name	ArMajorVersion
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	ArMinorVersion
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	3
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ArPatchVersion
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	0
Configuration class	PublishedInformation:



Origin	Elektrobit Automotive GmbH
---------------	----------------------------

Parameter Name	SwMajorVersion
Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMinorVersion
Label	Software Minor Version
Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	0
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwPatchVersion
Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	36
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ModuleId
Label	Numeric Module ID
Description	Module ID of this module from Module List



Multiplicity	1..1
Type	INTEGER_LABEL
Default value	177
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	VendorId
Label	Vendor ID
Description	Vendor ID of the dedicated implementation of this module according to the AU-TOSAR vendor list
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	Release
Label	Release Information
Multiplicity	1..1
Type	STRING_LABEL
Default value	
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

5.7.1.2. PublishedInformation

Parameters included	
Parameter name	Multiplicity
PbcfgMSupport	1..1

Parameter Name	PbcfgMSupport
Label	PbcfgM support
Description	Specifies whether or not the SomelTp can use the PbcfgM module for post-build support.



Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

5.7.1.3. SomelTpChannel

Containers included		
Container name	Multiplicity	Description
SomelTpRxNSdu	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.
SomelTpTxNSdu	0..n	The following parameters needs to be configured for each N-SDU that the SomelTp module transmits via the SomelTpChannel.

Parameters included	
Parameter name	Multiplicity
SomelTpNPduSeparationTime	1..1
SomelTpRxTimeoutTime	1..1
SomelTpTxConfirmationTimeout	1..1

Parameter Name	SomelTpNPduSeparationTime	
Description	Sets the duration of the minimum time in seconds the SomelTp module shall wait between the transmissions of N-PDUs.	
Multiplicity	1..1	
Type	FLOAT	
Configuration class	VariantLinkTime:	VariantLinkTime
	VariantPostBuild:	VariantPostBuild
	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	SomelTpRxTimeoutTime
-----------------------	-----------------------------



Description	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.	
Multiplicity	1..1	
Type	FLOAT	
Configuration class	VariantLinkTime:	VariantLinkTime
	VariantPostBuild:	VariantPostBuild
	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	SomelTpTxConfirmationTimeout	
Description	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. SomelTp_TxConfirmation) which is called only if successful transmission occurs without result argument, otherwise if transmission failed no SomelTp_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	
Multiplicity	1..1	
Type	FLOAT	
Configuration class	VariantLinkTime:	VariantLinkTime
	VariantPostBuild:	VariantPostBuild
	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

5.7.1.4. SomelTpRxNSdu

Containers included		
Container name	Multiplicity	Description
SomelTpRxNPdu	1..1	This container contains the configuration parameters of the NPdu that is received from a lower layer

Parameters included		
Parameter name	Multiplicity	
SomelTpRxSduRef	1..1	
Parameter Name	SomelTpRxSduRef	



Description	Reference to a Pdu in the COM-Stack that represents the assembled RxPdu which is passed via the PduR to the upper layer.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantLinkTime:	VariantLinkTime
	VariantPostBuild:	VariantPostBuild
	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.7.1.5. SomelpTpRxNPdu

Parameters included	
Parameter name	Multiplicity
SomelpTpRxNPduHandleId	1..1
SomelpTpRxNPduRef	1..1

Parameter Name	SomelpTpRxNPduHandleId	
Description	This parameter defines the handle ID that is used by the PduR when calling SomelpTp_RxIndication.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantLinkTime:	VariantLinkTime
	VariantPostBuild:	VariantPostBuild
	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	SomelpTpRxNPduRef	
Description	Reference to a global Pdu that is used to harmonize HandleIDs in the COM-Stack.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantLinkTime:	VariantLinkTime
	VariantPostBuild:	VariantPostBuild



	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

5.7.1.6. SomelpTpTxNSdu

Containers included		
Container name	Multiplicity	Description
SomelpTpTxNPdu	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
SomelpTpTxNSduHandleId	1..1
SomelpTpTxNSduRef	1..1

Parameter Name	SomelpTpTxNSduHandleId	
Description	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.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantLinkTime:	VariantLinkTime
	VariantPostBuild:	VariantPostBuild
	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	SomelpTpTxNSduRef	
Description	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.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantLinkTime:	VariantLinkTime
	VariantPostBuild:	VariantPostBuild
	VariantPreCompile:	VariantPreCompile



Origin	AUTOSAR_ECUC
--------	--------------

5.7.1.7. SomelpTpTxNPdu

Parameters included	
Parameter name	Multiplicity
SomelpTpTxNPduHandleId	1..1
SomelpTpTxNPduRef	1..1

Parameter Name	SomelpTpTxNPduHandleId	
Description	This parameter defines the handle ID that is used by PduR when calling SomelpTp_TriggerTransmit.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantLinkTime:	VariantLinkTime
	VariantPostBuild:	VariantPostBuild
	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	SomelpTpTxNPduRef	
Description	Reference to a global Pdu that is used to harmonize HandleIDs in the COM-Stack.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantLinkTime:	VariantLinkTime
	VariantPostBuild:	VariantPostBuild
	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.7.1.8. SomelpTpUnSupportedChannel

Containers included		
Container name	Multiplicity	Description



Containers included

SomelpTpUnSupportedRxNSdu	1..1	
SomelpTpUnSupportedTxNSdu	1..1	

5.7.1.9. SomelpTpUnSupportedRxNSdu

Containers included

Container name	Multiplicity	Description
SomelpTpUnSupportedRxNPdu	1..1	

5.7.1.10. SomelpTpUnSupportedRxNPdu

5.7.1.11. SomelpTpUnSupportedTxNSdu

Containers included

Container name	Multiplicity	Description
SomelpTpUnSupportedTxNPdu	1..1	

5.7.1.12. SomelpTpUnSupportedTxNPdu

5.7.1.13. SomelpTpGeneral

Parameters included

Parameter name	Multiplicity
SomelpTpDevErrorDetect	1..1
SomelpTpRxMainFunctionPeriod	1..1
SomelpTpTxMainFunctionPeriod	1..1



Parameters included

SomelpTpVersionInfoApi	1..1
--	------

Parameter Name	SomelpTpDevErrorDetect	
Description	Switches the Development Error Detection and Notification ON or OFF.	
Multiplicity	1..1	
Type	BOOLEAN	
Configuration class	VariantLinkTime: VariantPostBuild: VariantPreCompile:	VariantLinkTime VariantPostBuild VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	SomelpTpRxMainFunctionPeriod	
Description	This parameter defines the cycle time in seconds of the periodic call of the SomelpTp_MainFunctionRx.	
Multiplicity	1..1	
Type	FLOAT	
Range	<Infinity >0.0	
Configuration class	VariantLinkTime: VariantPostBuild: VariantPreCompile:	VariantLinkTime VariantPostBuild VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	SomelpTpTxMainFunctionPeriod	
Description	This parameter defines the cycle time in seconds of the periodic call of the SomelpTp_MainFunctionTx.	
Multiplicity	1..1	
Type	FLOAT	
Range	<Infinity >0.0	
Configuration class	VariantLinkTime: VariantPostBuild: VariantPreCompile:	VariantLinkTime VariantPostBuild VariantPreCompile



Origin	AUTOSAR_ECUC						
Parameter Name	SomelpTpVersionInfoApi						
Description	Activates the SomelpTp_GetVersionInfo() API.						
Multiplicity	1..1						
Type	BOOLEAN						
Default value	false						
Configuration class	<table><tr><td>VariantLinkTime:</td><td>VariantLinkTime</td></tr><tr><td>VariantPostBuild:</td><td>VariantPostBuild</td></tr><tr><td>VariantPreCompile:</td><td>VariantPreCompile</td></tr></table>	VariantLinkTime:	VariantLinkTime	VariantPostBuild:	VariantPostBuild	VariantPreCompile:	VariantPreCompile
VariantLinkTime:	VariantLinkTime						
VariantPostBuild:	VariantPostBuild						
VariantPreCompile:	VariantPreCompile						
Origin	AUTOSAR_ECUC						

5.7.2. Application programming interface (API)

5.7.2.1. Macro constants

5.7.2.1.1. SOMEIPTP_DET_REPORT_ERROR

Purpose	
Value	(void) Det_ReportError(SOMEIPTP_MODULE_ID, SOMEIPTP_INSTANCE_ID, (Apild), (ErrorId))

5.7.2.1.2. SOMEIPTP_DET_REPORT_RUNTIME_ERROR

Purpose	
Value	(void) Det_ReportRuntimeError(SOMEIPTP_MODULE_ID, SOMEIPTP_INSTANCE_ID, (Apild), (ErrorId))

5.7.2.1.3. SOMEIPTP_E_ASSEMBLY_INTERRUPT

Purpose	Definition of DET error code SOMEIPTP_E_ASSEMBLY_INTERRUPT.
---------	---



Value	0x08U
--------------	-------

5.7.2.1.4. SOMEIPTP_E_ASSEMBLY_INTERRUPT_SOR

Purpose	Definition of DET error code SOMEIPTP_E_ASSEMBLY_INTERRUPT for start of reception.
Value	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.
Value	0x01U

5.7.2.1.12. SOMEIPTP_E_PARAM

Purpose	Definition of DET error code SOMEIPTP_E_PARAM.
Value	0x03U

5.7.2.1.13. SOMEIPTP_E_PARAM_POINTER

Purpose	Definition of DET error code SOMEIPTP_E_PARAM_POINTER.
Value	0x02U

5.7.2.1.14. SOMEIPTP_INSTANCE_ID

Purpose	Module instance ID.
Value	0U
Description	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

Purpose	Defines API id of function SomelpTp_GetVersionInfo() .
----------------	--



Value	0x01U
--------------	-------

5.7.2.1.16. SOMEIPTP_SID_INIT

Purpose	Defines API id of function SomelpTp_Init() .
Value	0x02U

5.7.2.1.17. SOMEIPTP_SID_MAINFUNCTIONRX

Purpose	Defines API id of function SomelpTp_MainFunctionRx() .
Value	0x04U

5.7.2.1.18. SOMEIPTP_SID_MAINFUNCTIONTX

Purpose	Defines API id of function SomelpTp_MainFunctionTx() .
Value	0x03U

5.7.2.1.19. SOMEIPTP_SID_PDURIFTRANSMIT

Purpose	Defines Dummy API id of function SomelpTp_PduRifTransmit() .
Value	0xFF

5.7.2.1.20. SOMEIPTP_SID_PDURTPCOPYRXDATA

Purpose	Defines Dummy API id of function SomelpTp_PduRTpCopyRxData() .
Value	0xFB

5.7.2.1.21. SOMEIPTP_SID_PDURTPCOPYTXDATA

Purpose	Defines Dummy API id of function SomelpTp_PduRTpCopyTxData() .
Value	0xFD



5.7.2.1.22. SOMEIPTP_SID_PDURTPRXINDICATION

Purpose	Defines Dummy API id of function SomelpTp_PduRTpRxIndication() .
Value	0xFA

5.7.2.1.23. SOMEIPTP_SID_PDURTPSTARTOFRECEPTION

Purpose	Defines Dummy API id of function SomelpTp_PduRTpStartOfReception() .
Value	0xFC

5.7.2.1.24. SOMEIPTP_SID_PDURTPTXCONFIRMATION

Purpose	Defines Dummy API id of function SomelpTp_PduRTpTxConfirmation() .
Value	0xFE

5.7.2.1.25. SOMEIPTP_SID_RXINDICATION

Purpose	Defines API id of function SomelpTp_RxIndication() .
Value	0x42U

5.7.2.1.26. SOMEIPTP_SID_TRANSMIT

Purpose	Defines API id of function SomelpTp_Transmit() .
Value	0x49U

5.7.2.1.27. SOMEIPTP_SID_TRIGGERTRANSMIT

Purpose	Defines API id of function SomelpTp_TriggerTransmit() .
Value	0x41U

5.7.2.1.28. SOMEIPTP_SID_TXCONFIRMATION

Purpose	Defines API id of function SomelpTp_TxConfirmation() .
---------	--



Value	0x40U
-------	-------

5.7.2.2. Functions

5.7.2.2.1. SomeIpTp_GetVersionInfo

Purpose	API to get the version information of SomeIpTp module.	
Synopsis	<code>void SomeIpTp_GetVersionInfo (Std_VersionInfoType * VersionInfo);</code>	
Parameters (out)	VersionInfo	- Pointer to where to store the version information of this module.
Description	This service returns the version information of this module.	

5.7.2.2.2. SomeIpTp_Init

Purpose	Initializes the SomeIpTp module.	
Synopsis	<code>void SomeIpTp_Init (const SomeIpTp_ConfigType * config);</code>	
Parameters (in)	config	- Base pointer to the configuration structure of the SOME/IP TP module.
Description	This service initializes the SomeIpTp module. It shall be the first function of the module to be called.	

5.7.2.2.3. SomeIpTp_MainFunctionRx

Purpose	SomeIpTp module main function for reception.	
Synopsis	<code>void SomeIpTp_MainFunctionRx (void);</code>	
Description	This function performs the processing of the AUTOSAR SOME/IP TP module's reception activities.	

5.7.2.2.4. SomeIpTp_MainFunctionTx

Purpose	SomeIpTp module main function for Transmission.	
---------	---	--



Synopsis	<code>void SomeIpTp_MainFunctionTx (void);</code>
Description	This function performs the processing of the AUTOSAR SOME/IP TP module's transmission activities.

5.7.2.2.5. SomelpTp_PduRIfTransmit

Purpose	Dummy funtion for PduR.
Synopsis	<code>Std_ReturnType SomeIpTp_PduRIfTransmit (PduIdType TxPduId , const PduInfoType * PduInfoPtr);</code>
Service ID	0xFF
Return Value	
Description	This function shall not be called, and if so it shall report developement error.

5.7.2.2.6. SomelpTp_PduRTpCopyRxData

Purpose	Dummy funtion for PduR.
Synopsis	<code>BufReq_ReturnType SomeIpTp_PduRTpCopyRxData (PduIdType PduId , const PduInfoType * PduInfoPointer , PduLengthType * RxBufferSizePtr);</code>
Service ID	0xFB
Return Value	
Description	This function shall not be called, and if so it shall report developement error.

5.7.2.2.7. SomelpTp_PduRTpCopyTxData

Purpose	Dummy funtion for PduR.
Synopsis	<code>BufReq_ReturnType SomeIpTp_PduRTpCopyTxData (PduIdType PduId , PduInfoType * PduInfoPtr , RetryInfoType * RetryInfoPtr , PduLengthType * TxDataCntPtr);</code>
Service ID	0xFD
Return Value	
Description	This function shall not be called, and if so it shall report developement error.



5.7.2.2.8. SomeIpTp_PduRTpRxIndication

Purpose	Dummy funtion for PduR.
Synopsis	<code>void SomeIpTp_PduRTpRxIndication (PduIdType RxPduId , NotifResultType Result);</code>
Service ID	0xFA
Description	This function shall not be called, and if so it shall report developement error.

5.7.2.2.9. SomeIpTp_PduRTpStartOfReception

Purpose	Dummy funtion for PduR.
Synopsis	<code>BufReq_ReturnType SomeIpTp_PduRTpStartOfReception (PduIdType PduId , PduLengthType PduLength , PduLengthType * RxBufferSizePtr);</code>
Service ID	0xFC
Return Value	
Description	This function shall not be called, and if so it shall report developement error.

5.7.2.2.10. SomeIpTp_PduRTpTxConfirmation

Purpose	Dummy funtion for PduR.
Synopsis	<code>void SomeIpTp_PduRTpTxConfirmation (PduIdType PduId , NotifResultType Result);</code>
Service ID	0xFE
Description	This function shall not be called, and if so it shall report developement error.

5.7.2.2.11. SomeIpTp_RxIndication

Purpose	SomeIpTp module reception indication callback function.
Synopsis	<code>void SomeIpTp_RxIndication (PduIdType RxPduId , PduInfoType * PduInfoPtr);</code>



Parameters (in)	RxPduId	- ID received PDU.
	PduInfoPtr	- pointer to a buffer containing the PDU data, and length of the received PDU.
Description	This callback function is called to indicate a received PDU from a lower layer communication interface module	

5.7.2.2.12. SomeIpTp_Transmit

Purpose	SomeIpTp module Transmit request function.	
Synopsis	Std_ReturnType SomeIpTp_Transmit (PduIdType TxPduId , const PduInfoType * PduInfoPtr);	
Service ID	0x49	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Pdulds Non Reentrant for the same Pduld	
Parameters (in)	TxPduId	- Identifier of the PDU to be transmitted.
	PduInfoPtr	- Length of and pointer to the PDU data and pointer to MetaData.
Return Value		
Std_ReturnType	E_OK if transmit request has been accepted. E_NOT_OK if not accepted.	
Description	This service requests transmission of a PDU.	

5.7.2.2.13. SomeIpTp_TriggerTransmit

Purpose	SomeIpTp module trigger transmit callback function.	
Synopsis	Std_ReturnType SomeIpTp_TriggerTransmit (PduIdType TxPduId , PduInfoType * PduInfoPtr);	
Service ID	0x41	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Pdulds Reentrant for the same Pduld	
Parameters (in)	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 SduLengh. On return, the service will indicate the length of the copied SDU data in SduLength.
Parameters (in,out)	inout]	PduInfoPtr - pointer to a buffer to where the SDU data shall be copied, and the available buffer size in SduLengh. On return, the service will indicate the length of the copied SDU data in SduLength.
Return Value		
Std_ReturnType	E_OK if SDU has been copied. E_NOT_OK if not copied.	
Description	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

Purpose	SomeIpTp module Transmission confirmation callback function.	
Synopsis	<code>void SomeIpTp_TxConfirmation (PduIdType TxPduId);</code>	
Parameters (in)	TxPduId	- ID of the PDU that has been transmitted.
Description	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

Protected data structures	All shared data in the Rx path 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.
--------------------------------------	--

5.7.3.1.2. SCHM_SOMEIPTP_EXCLUSIVE_AREA_1

Protected data structures	All shared data in the Tx path 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.

5.7.3.2. 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
VAR_CLEARED_8
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 SomelpTp module.

5.8. Tcplp

5.8.1. Configuration parameters

Containers included		
Container name	Multiplicity	Description
CommonPublishedInformation	1..1	Label: Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
PublishedInformation	1..1	Label: EB Published Information Additional published parameters not covered by CommonPublishedInformation container.
TcplpConfig	1..1	This container contains the configuration parameters and sub containers of the AUTOSAR Tcplp module.
TcplpDefensiveProgramming	1..1	Label: Defensive Programming Options Parameters for defensive programming
TcplpGeneral	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
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT
----------------	-------------------------------



Multiplicity	1..1
Type	ENUMERATION
Default value	VariantPostBuild
Range	VariantPostBuild

5.8.1.1. CommonPublishedInformation

Parameters included	
Parameter name	Multiplicity
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1
SwPatchVersion	1..1
ModuleId	1..1
VendorId	1..1
Release	1..1

Parameter Name	ArMajorVersion
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	ArMinorVersion
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	3
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ArPatchVersion
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	0
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMajorVersion
Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	3
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	SwMinorVersion
Label	Software Minor Version
Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	5
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH



Parameter Name	SwPatchVersion
Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	17
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	ModuleId
Label	Numeric Module ID
Description	Module ID of this module from Module List
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	170
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	VendorId
Label	Vendor ID
Description	Vendor ID of the dedicated implementation of this module according to the AU-TOSAR vendor list
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	Release
Label	Release Information
Multiplicity	1..1
Type	STRING_LABEL
Default value	



Configuration class	PublishedInformation:	
Origin	Elektrobit Automotive GmbH	

5.8.1.2. PublishedInformation

Parameters included	
Parameter name	Multiplicity
PbcfgMSupport	1..1

Parameter Name	PbcfgMSupport
Label	PbcfgM support
Description	Specifies whether or not the Tcplp can use the PbcfgM module for post-build support.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

5.8.1.3. TcplpConfig

Containers included		
Container name	Multiplicity	Description
TcplpCtrl	1..n	Specifies the Ethlf controller used for IP communication and Tcplp errors that shall be reported to DEM.
TcplpDhcpServerConfig	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>
TcplpIpConfig	0..1	Specifies the configuration parameters of the IP (Internet Protocol) sub-module.
TcplpLocalAddr	1..253	Specifies the local IP (Internet Protocol) addresses used for IP communication.



Containers included		
TcplpNvmBlock	0..1	<i>The functionality related to this parameter is not supported by the current implementation.</i> 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).
TcplpPhysAddrConfig	0..1	Specifies the physical address configuration.
TcplpSocketOwnerConfig	1..1	Specifies the upper layer modules of Tcplp using the socket API.
TcplpTcpConfig	0..1	Specifies the configuration parameters of the TCP (Transmission Control Protocol) sub-module.
TcplpUdpConfig	0..1	This container is a subcontainer of TcplpConfig and specifies the configuration parameters of the UDP (User Datagram Protocol) sub-module.
TcplpDuplicateAddressDetectionConfig	0..1	Specifies the DAD callout function
TcplpRxPolicyCheckIngressHandlerConfig	0..1	Specifies the policy check ingress handler function
TcplpRxPolicyCheckRoutingHandlerConfig	0..1	Specifies the policy check routing handler function
TcplpRxPacketPostProcessHandlerConfig	0..1	Specifies the post process handler function
TcplpRxPacketDropHandlerConfig	0..1	Specifies the packet drop handler function
TcplpMemoryConfig	1..1	Specifies the configuration parameters for memory pools.
TcplpIpSecConfig	1..1	Specifies the configuration parameters of the IPsec sub-module.

5.8.1.4. TcplpCtrl

Containers included		
Container name	Multiplicity	Description
TcplpOffloadChecksum	1..1	This container is a subcontainer of TcplpCtrl and specifies the parameters of the software/hardware checksum calculations.

**Containers included**

TcplIpVXCtrl	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
TcplIpFramePrioDefault	0..1
TcplDhcpServerConfigRef	0..1
TcplEthIfCtrlRef	1..1

Parameter Name	TcplIpFramePrioDefault	
Description	Specifies the default value for the frame priority used by all sockets.	
Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Range	<=7 >=0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplDhcpServerConfigRef	
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> Reference to a TcplDhcpServerConfig which shall be used for this controller setting (VLAN).	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplEthIfCtrlRef	
-----------------------	-------------------------	--



Description	Reference to Ethlf controller where the IP address shall be assigned	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.8.1.5. TcplpOffloadChecksum

Parameters included	
Parameter name	Multiplicity
TcplpEnableOffloadChecksumIPv4	1..1
TcplpEnableOffloadChecksumICMP	1..1
TcplpEnableOffloadChecksumTCP	1..1
TcplpEnableOffloadChecksumUDP	1..1

Parameter Name	TcplpEnableOffloadChecksumIPv4	
Description	Enables (FALSE) or disables (TRUE) the IPv4 checksum calculation in software.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpEnableOffloadChecksumICMP	
Description	Enables (FALSE) or disables (TRUE) the ICMP checksum calculation in software.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpEnableOffloadChecksumTCP	



Description	Enables (FALSE) or disables (TRUE) the TCP checksum calculation in software.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpEnableOffloadChecksumUDP	
Description	Enables (FALSE) or disables (TRUE) the UDP checksum calculation in software.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.6. TcpllpVXCtrl

Containers included		
Container name	Multiplicity	Description
TcpllpV4Ctrl	1..1	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Specifies an Internet Protocol version 4 (IPv4) instance.</p>
TcpllpV6Ctrl	1..1	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Specifies an Internet Protocol version 6 (IPv6) instance.</p>

5.8.1.7. TcpllpV4Ctrl

Containers included		
Container name	Multiplicity	Description
TcpllpV4MtuConfig	1..1	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p>

**Containers included**

		This container specifies the Maximum Transmission Unit parameters for this IPv4 instance.
--	--	---

Parameters included

Parameter name	Multiplicity
TcplpArpConfigRef	0..1
TcplpAutolpConfigRef	0..1
TcplpDhcpConfigRef	0..1
TcplpFragmentationConfigRef	0..1

Parameter Name**TcplpArpConfigRef****Description**

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

Reference to ARP configuration for this IPv4 instance.

Multiplicity

0..1

Type

REFERENCE

Configuration class

PostBuild:	VariantPostBuild
-------------------	------------------

Origin

AUTOSAR_ECUC

Parameter Name**TcplpAutolpConfigRef****Description**

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

Reference to Autolp configuration for this IPv4 instance.

Multiplicity

0..1

Type

REFERENCE

Configuration class

PostBuild:	VariantPostBuild
-------------------	------------------

Origin

AUTOSAR_ECUC

Parameter Name**TcplpDhcpConfigRef****Description**

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

Reference to DHCP configuration for this IPv4 instance.

Multiplicity

0..1



Type	REFERENCE	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpFragmentationConfigRef	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Reference to Fragmentation configuration for this IPv4 instance.</p>	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.8.1.8. TcpIpV4MtuConfig

Parameters included	
Parameter name	Multiplicity
TcpIpV4PathMtuEnabled	1..1
TcpIpV4PathMtuTimeout	1..1

Parameter Name	TcpIpV4PathMtuEnabled	
Description	<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>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpV4PathMtuTimeout	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p>	



	If this value is >0 the IpV4 will reset the MTU value stored for each destination after n seconds.
Multiplicity	1..1
Type	FLOAT
Default value	600.0
Range	<=86400.0 >=1.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.8.1.9. TcpllpV6Ctrl

Containers included		
Container name	Multiplicity	Description
TcpllpV6MtuConfig	1..1	This container specifies the Maximum Transmission Unit parameters for this IPv6 instance.

Parameters included	
Parameter name	Multiplicity
TcpllpV6DhcpConfigRef	0..1
TcpllpV6FragmentationConfigRef	0..1
TcpllpV6NdpConfigRef	1..1

Parameter Name	TcpllpV6DhcpConfigRef
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> Reference to DHCPv6 configuration.
Multiplicity	0..1
Type	REFERENCE
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpllpV6FragmentationConfigRef
-----------------------	--------------------------------



Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> Reference to IPv6 Fragmentation Configuration	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpIpV6NdpConfigRef	
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> Reference to Neighbor Discovery Protocol Configuration.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.8.1.10. TcpIpIpV6MtuConfig

Parameters included		
Parameter name		Multiplicity
TcplpIpV6PathMtuEnabled		1..1
TcplpIpV6PathMtuTimeout		1..1

Parameter Name	TcplpIpV6PathMtuEnabled	
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> If enabled the IPv6 processes incoming ICMPv6 "Packet Too Big" messages and stores a MTU value for each destination address.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	AUTOSAR_ECUC
---------------	--------------

Parameter Name	TcpllpV6PathMtuTimeout	
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> If this value is >0 the IPv6 will reset the MTU value stored for each destination after n seconds.	
Multiplicity	1..1	
Type	FLOAT	
Default value	600.0	
Range	<=86400.0 >=1.0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.8.1.11. TcplpDhcpServerConfig

Containers included		
Container name	Multiplicity	Description
TcplpDhcpAddressAssignment	0..n	Defines a Ethernet Switch port based IP address assignment.

Parameters included	
Parameter name	Multiplicity
TcplpDhcpDefaultRouter	0..1
TcplpDhcpNetmask	0..1
TcplpDhcpEthIfSwitchRef	0..1

Parameter Name	TcplpDhcpDefaultRouter	
Description	IP address of default router (gateway).	
Multiplicity	0..1	
Type	STRING	
Configuration class	PostBuild:	VariantPostBuild



Origin	AUTOSAR_ECUC
---------------	--------------

Parameter Name	TcplpDhcpNetmask	
Description	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.	
Multiplicity	0..1	
Type	INTEGER	
Range	<=128 >=0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplpDhcpEthIfSwitchRef	
Description	Reference to EthIfSwitch representation.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.8.1.12. TcplpDhcpAddressAssignment

Parameters included		
Parameter name	Multiplicity	
TcplpDhcpAddressLowerBound	1..1	
TcplpDhcpAddressUpperBound	1..1	
TcplpDhcpSwitchPortRef	0..1	

Parameter Name	TcplpDhcpAddressLowerBound	
Description	The lower bound IP address which shall be assigned.	
Multiplicity	1..1	
Type	STRING	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpDhcpAddressUpperBound	
Description	The upper bound IP address which shall be assigned.	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpDhcpSwitchPortRef	
Description	Reference to Ethernet Switch port.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.8.1.13. TcpllpConfig

Containers included		
Container name	Multiplicity	Description
TcpllpFragmentationConfig	1..1	Specifies the configuration parameters of IPv4 & IPv6 packet fragmentation/reassembly.
TcpllpV4Config	0..1	Specifies the configuration parameters of the IPv4 (Internet Protocol version 4) sub-module.
TcpllpV6Config	0..1	Specifies the configuration parameters of the IPv6 (Internet Protocol version 6) sub-module.

5.8.1.14. TcpllpFragmentationConfig

Parameters included	
Parameter name	Multiplicity



Parameters included

TcpllpFragMemReserved	0..1
TcpllpFragmentationRxEnabled	1..1
TcpllpReassemblyTimeout	1..1
TcpllpReassemblyBufferCount	1..1
TcpllpReassemblyBufferSize	1..1
TcpllpFragmentationTxEnabled	1..1
TcpllpTxFragmentBufferCount	1..1
TcpllpTxFragmentBufferSize	1..1
TcpllpTxFragmentSegmentCount	1..1

Parameter Name	TcpllpFragMemReserved	
Description	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.	
Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpllpFragmentationRxEnabled	
Description	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"> ▶ true: IP Datagram Reassembly enabled ▶ false: IP Datagram Reassembly disabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpllpReassemblyTimeout	



Description	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.
Multiplicity	1..1
Type	FLOAT
Default value	60.0
Configuration class	PostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIplpReassemblyBufferCount
Description	Number of fragmented IP datagrams that can be reassembled in parallel.
Multiplicity	1..1
Type	INTEGER
Default value	2
Configuration class	PostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIplpReassemblyBufferSize
Description	Size of each reassembly buffer.
Multiplicity	1..1
Type	INTEGER
Default value	1500
Configuration class	PostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIplpFragmentationTxEnabled
Description	Enables or disables support for fragmenting outgoing datagrams according to IETF RFC 791 / RFC 2460 Available choices: <ul style="list-style-type: none">▶ 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.
Multiplicity	1..1
Type	ENUMERATION
Default value	OFF
Range	OFF OUTOFORDER INORDER
Configuration class	PreCompile: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpTxFragmentBufferCount
Description	For in-order transmission: TcpIpTxEnabled = INORDER Maximum number of transmit buffers. Number of fragmented IP datagrams that can be sent in parallel.
Multiplicity	1..1
Type	INTEGER
Default value	2
Configuration class	PostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpTxFragmentBufferSize
Description	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.
Multiplicity	1..1
Type	INTEGER
Default value	1500
Configuration class	PostBuild: VariantPostBuild



Origin	Elektrobit Automotive GmbH
Parameter Name	TcpIpTxFragmentSegmentCount
Description	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.
Multiplicity	1..1
Type	INTEGER
Default value	5
Configuration class	PostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.15. TcpllpV4Config

Containers included		
Container name	Multiplicity	Description
TcplpArpConfig	0..1	Specifies the configuration parameters of the ARP (Address Resolution Protocol) sub-module.
TcplpAutoIpConfig	0..1	Specifies the configuration parameters of the Auto-IP (automatic private IP addressing) sub-module.
TcplpDhcpConfig	0..1	Specifies the configuration parameters of the DHCPv4.
TcplpIcmpConfig	0..1	Specifies the configuration parameters of the ICMP (Internet Control Message Protocol) sub-module.
TcpllpV4ArpPacketFilter	0..1	This container is a subcontainer of TcpllpConfig and specifies the configuration parameters for the ARP packet filter.
TcpllpFragmentationConfig	0..n	DISABLED - Tcplp/TcpllpConfig/TcpllpFragmentationConfig is used instead for both IPv4 & IPv6! Specifies the configuration parameters of IPv4 packet fragmentation/reassembly.

Parameters included	
Parameter name	Multiplicity



Parameters included

TcpllpV4StaticAddrDefenseMechanism	1..1
--	------

Parameter Name	TcpllpV4StaticAddrDefenseMechanism
Description	<p>Specifies the performed defense mechanism for TCPIP_STATIC assignment method if an address conflict is detected.</p> <ul style="list-style-type: none"> ▶ TCPIP_DISABLE: Address Conflict Detection is not performed. ▶ TCPIP_ABANDON_ADDR: If an address conflict is detected the address is ceased immediately. ▶ 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. ▶ 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.
Multiplicity	1..1
Type	ENUMERATION
Default value	TCPIP_DISABLE
Range	TCPIP_DISABLE TCPIP_ABANDON_ADDR TCPIP_DEFEND_ADDR TCPIP_DEFEND_INF_ADDR
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.16. TcplpArpConfig

Parameters included

Parameter name	Multiplicity
TcplpArpNumGratuitousARPPonStartup	1..1
TcplpArpPacketQueueEnabled	1..1
TcplpArpRequestTimeout	1..1
TcplpArpTableEntryTimeout	1..1

**Parameters included**

TcplpArpTableSizeMax	1..1
--------------------------------------	------

Parameter Name	TcplpArpNumGratuitousARPonStartup	
Description	Specifies the number of gratuitous ARP replies which shall be sent on assignment of a new IP address.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255	>=0
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplpArpPacketQueueEnabled	
Description	<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.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplpArpRequestTimeout	
Description	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	
Multiplicity	1..1	
Type	FLOAT	
Default value	2	
Range	<=255	>=0
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	Elektrobit Automotive GmbH
---------------	----------------------------

Parameter Name	TcplpArpTableEntryTimeout	
Description	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.	
Multiplicity	1..1	
Type	FLOAT	
Default value	60	
Range	=Infinity <=65535 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplpArpTableSizeMax	
Description	Maximum number of entries in the ARP table.	
Multiplicity	1..1	
Type	INTEGER	
Default value	5	
Range	<=65535 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.8.1.17. TcplpAutolpConfig

Parameters included	
Parameter name	Multiplicity
TcplpAutolpInitTimeout	1..1
TcplpIpV4AutolpAddrDefenseMechanism	1..1
TcplpAutolpv4EntriesMax	1..1
Parameter Name	TcplpAutolpInitTimeout



Description	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.
Multiplicity	1..1
Type	FLOAT
Default value	60
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpIpv4AutolpAddrDefenseMechanism
Description	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"> ▶ TCPIP_DISABLE: Address Conflict Detection is not performed. ▶ TCPIP_ABANDON_ADDR: If an address conflict is detected the address is ceased immediately. ▶ 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. ▶ 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.
Multiplicity	1..1
Type	ENUMERATION
Default value	TCPIP_DEFEND_ADDR
Range	TCPIP_DISABLE TCPIP_ABANDON_ADDR TCPIP_DEFEND_ADDR TCPIP_DEFEND_INF_ADDR
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpAutolpv4EntriesMax
Description	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>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	INTEGER
Default value	1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.18. TcplpDhcpConfig

Containers included		
Container name	Multiplicity	Description
TcplpDhcpConfigurableOptions	0..n	

Parameters included	
Parameter name	Multiplicity
TcplpDhcplpv4EntriesMax	1..1
TcplpDhcpInitDelay	1..1
TcplpIpv4DhcpAddrDefenseMechanism	1..1
TcplpDhcpFQDNOptionEnabled	1..1
TcplpDhcplpv4DomainNameMaxSize	1..1
TcplpDhcpConfigurableOptionsEnabled	1..1
TcplpDhcpConfigurableOptionsEntriesMax	1..1
TcplpDhcpConfigurableOptionsDataSizeMax	1..1
TcplpDhcpArpProbingEnabled	1..1
TcplpDhcpArpProbingType	1..1

Parameter Name	TcplpDhcplpv4EntriesMax
Description	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>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	INTEGER
Default value	1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpDhcpInitDelay
Description	The time in seconds DHCP waits at startup, before beginning with the transmission of the DHCPDISCOVER message.
Multiplicity	1..1
Type	INTEGER
Default value	1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpV4DhcpAddrDefenseMechanism
Description	<p>Specifies the performed defense mechanism for TCPIP_DHCP assignment method if an address conflict is detected.</p> <ul style="list-style-type: none"> ▶ TCPIP_DISABLE: Address Conflict Detection is not performed. ▶ TCPIP_ABANDON_ADDR: If an address conflict is detected the address is ceased immediately. ▶ 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. ▶ 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.
Multiplicity	1..1
Type	ENUMERATION
Default value	TCPIP_DISABLE



Range	TCPIP_DISABLE TCPIP_ABANDON_ADDR TCPIP_DEFEND_ADDR TCPIP_DEFEND_INF_ADDR
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpDhcpFQDNOptionEnabled
Description	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"> ▶ true: Enables the transmission of the hostname option. ▶ false: Disables the transmission of the hostname option.
Optimization Effect:	<ul style="list-style-type: none"> ▶ RAM reduction: Disabling this parameter reduces the RAM consumption of the module. ▶ Execution time reduction (code): Disabling this parameter reduces the execution time of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpDhcpIpv4DomainNameMaxSize
Description	This parameter defines the maximum size of the DHCPv4 Domain Name. This parameter has impact on the size of the PostBuild RAM.
Optimization Effect:	<ul style="list-style-type: none"> ▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	INTEGER



Default value	5
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpDhcpConfigurableOptionsEnabled
Description	Enables (TRUE) or disables (FALSE) Dhcp options Optimization Effect: <ul style="list-style-type: none"> ▶ ROM reduction (config): Disabling this parameter reduces the ROM consumption of the module configuration. ▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code. ▶ RAM reduction: Disabling this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpDhcpConfigurableOptionsEntriesMax
Description	Maximum number of Dhcp option entries in PostBuild configuration. This parameter has impact on the size of the PostBuild RAM. Optimization Effect: <ul style="list-style-type: none"> ▶ ROM reduction (config): Selecting a small value for this parameter reduces the ROM consumption of the module configuration. ▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	INTEGER
Configuration class	PreCompile: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpDhcpConfigurableOptionsDataSizeMax
Description	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>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (config): Selecting a small value for this parameter reduces the ROM consumption of the module configuration. ▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	INTEGER
Configuration class	PreCompile: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpDhcpArpProbingEnabled
Description	Enable DhcpV4 Arp probing (Duplicate Address Detection according to RFC2131 and RFC5227).
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpDhcpArpProbingType
Description	<p>Sets the probing type for Dhcp</p> <p>Available choices:</p> <ul style="list-style-type: none"> ▶ PROBING_DEFAULT: Probing is done according to the default probing specifications. ▶ PROBING_DOIP: Probing is done according to the specifications for TCPIP_LINKLOCAL_DOIP.
Multiplicity	1..1
Type	ENUMERATION
Default value	PROBING_DEFAULT
Range	PROBING_DEFAULT PROBING_DOIP
Configuration class	PreCompile: VariantPostBuild



Origin	Elektrobit Automotive GmbH
--------	----------------------------

5.8.1.19. TcplpDhcpConfigurableOptions

Parameters included	
Parameter name	TcplpDhcpOptionCode
Description	Code of the Dhcp option
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH
Parameter Name	
Parameter Name	TcplpDhcpOptionMaxLength
Description	Memory which shall be reserved for this Dhcp option
Multiplicity	1..1
Type	INTEGER
Range	<=255 >=1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH
Parameter Name	
Parameter Name	TcplpDhcpOptionTransmit
Description	Indicates if the option shall be transmitted in every DHCP message or just stored when it is received from the server. <ul style="list-style-type: none">▶ true: option is transmitted▶ false: option is stored when received
Multiplicity	1..1
Type	BOOLEAN
Default value	true



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.20. TcplplcmpConfig

Containers included		
Container name	Multiplicity	Description
TcplplcmpMsgHandler	0..1	This container is a subcontainer of cplplcmpConfig and specifies the configuration parameters for the ICMP message handler.

Parameters included	
Parameter name	Multiplicity
TcplplcmpEchoReplyEnabled	1..1
TcplplcmpTtl	1..1

Parameter Name	TcplplcmpEchoReplyEnabled
Description	Enables or disables transmission of ICMP echo reply message in case of a ICMP echo reception. <ul style="list-style-type: none"> ▶ true: ICMP echo reply enabled ▶ false: ICMP echo reply disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

Parameter Name	TcplplcmpTtl
Description	Default Time-to-live value of outgoing ICMP packets.
Multiplicity	1..1
Type	INTEGER
Default value	255
Range	<=255 >=1



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.8.1.21. TcplplcmplMsgHandler

Parameters included	
Parameter name	Multiplicity
TcplplcmplMsgHandlerHeaderFileName	1..1
TcplplcmplMsgHandlerName	1..1

Parameter Name	TcplplcmplMsgHandlerHeaderFileName
Description	This parameter specifies the name of the header file containing the definition of the ICMP message handler function..
Multiplicity	1..1
Type	STRING
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

Parameter Name	TcplplcmplMsgHandlerName
Description	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)
Multiplicity	1..1
Type	FUNCTION-NAME
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

5.8.1.22. TcplplpV4ArpPacketFilter

Parameters included	
Parameter name	Multiplicity



Parameters included

TcpllpV4ArpPacketFilterFileName	1..1
TcpllpV4ArpPacketFilterName	1..1

Parameter Name	TcpllpV4ArpPacketFilterFileName
Description	This parameter specifies the name of the header file containing the definition of the ARP packet filter function.
Multiplicity	1..1
Type	STRING
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpllpV4ArpPacketFilterName
Description	<p>This parameter defines the name of the ARP packet filter function. The function must follow the signature:</p> <pre><function>boolean <function_name>(Tcplp_LocalAddrIdType localAddrId, uint8 *dataPtr, uint16 lenByte)</function></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>
Multiplicity	1..1
Type	FUNCTION-NAME
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.23. TcpllpFragmentationConfig

Parameters included

Parameter name	Multiplicity
TcpllpFragmentationRxEnabled	1..1
TcpllpNumFragments	0..1
TcpllpNumReassDgrams	0..1

**Parameters included**

TcpllpReassTimeout	0..1
------------------------------------	------

Parameter Name	TcpllpFragmentationRxEnabled
Description	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"> ▶ true: IP Datagram Reassembly enabled ▶ false: IP Datagram Reassembly disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpllpNumFragments
Description	Maximum number of IP fragments per datagram.
Multiplicity	0..1
Type	INTEGER
Default value	10
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpllpNumReassDgrams
Description	Number of fragmented IP datagrams that can be reassembled in parallel.
Multiplicity	0..1
Type	INTEGER
Default value	10
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpllpReassTimeout
Description	Time after which an incomplete datagram gets discarded. RFC1122 suggests a value between 60 and 120 seconds.



Multiplicity	0..1	
Type	FLOAT	
Default value	60.0	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.8.1.24. TcplIpv6Config

Containers included		
Container name	Multiplicity	Description
TcplpDhcpV6Config	0..1	Specifies the configuration parameters of the DHCPv6.
TcplpIcmpV6Config	1..1	Specifies the configuration parameters of the ICMPv6 (Internet Control Message Protocol for IPv6) sub-module.
TcplpIpV6FragmentationConfig	0..n	<p>DISABLED - Tcplp/TcplpConfig/TcplpIpFragmentationConfig is used instead for both IPv4 & IPv6!</p> <p>Specifies the configuration parameters of IPv6 packet fragmentation/reassembly.</p>
TcplpNdpConfig	0..1	Specifies the configuration parameters of the Neighbor Discovery Protocol for IPv6.
TcplpIpV6ConfigExtHeaderFilter	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

Containers included		
Container name	Multiplicity	Description
TcplpDhcpV6ConfigurableOptions	0..n	

Parameters included		
Parameter name	Multiplicity	
TcplpDhcpV6CnfDelayMax	1..1	
TcplpDhcpV6CnfDelayMin	1..1	



Parameters included

TcplpDhcpV6InfDelayMax	1..1
TcplpDhcpV6InfDelayMin	1..1
TcplpDhcpV6SolDelayMax	1..1
TcplpDhcpV6SolDelayMin	1..1
TcplpDhcplpv6EntriesMax	1..1
TcplpDhcplpv6ServerDuidMaxSize	1..1
TcplpDhcplpv6DomainNameMaxSize	1..1
TcplpDhcpV6FQDNOptionEnabled	1..1
TcplpDhcpV6ConfigurableOptionsEnabled	1..1
TcplpDhcpV6ConfigurableOptionsEntriesMax	1..1
TcplpDhcpV6ConfigurableOptionsDataSizeMax	1..1
TcplpDhcpV6SlaacDadEnabled	1..1

Parameter Name	TcplpDhcpV6CnfDelayMax
Description	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.
Multiplicity	1..1
Type	FLOAT
Default value	1.0
Range	<=100.0 >=0.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcplpDhcpV6CnfDelayMin
Description	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.
Multiplicity	1..1
Type	FLOAT



Default value	0.0
Range	<=100.0 >=0.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpDhcpV6InfDelayMax
Description	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.
Multiplicity	1..1
Type	FLOAT
Default value	1.0
Range	<=100.0 >=0.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpDhcpV6InfDelayMin
Description	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.
Multiplicity	1..1
Type	FLOAT
Default value	0.0
Range	<=100.0 >=0.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC



Parameter Name	TcpIpDhcpV6SolDelayMax
Description	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
Multiplicity	1..1
Type	FLOAT
Default value	1.0
Range	<=100.0 >=0.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpDhcpV6SolDelayMin
Description	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.
Multiplicity	1..1
Type	FLOAT
Default value	0.0
Range	<=100.0 >=0.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpDhcplpv6EntriesMax
Description	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:



	► RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	INTEGER
Default value	1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpDhcIpv6ServerDuidMaxSize
Description	This parameter defines the maximum supported size of the DHCPv6 Server DUID Default value is based on DUID_LLT This parameter has impact on the size of the PostBuild RAM. Optimization Effect:
	► RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	INTEGER
Default value	12
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpDhcIpv6DomainNameMaxSize
Description	This parameter defines the maximum size of the DHCPv6 Domain Name. This parameter has impact on the size of the PostBuild RAM. Optimization Effect:
	► RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	INTEGER
Default value	5
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH



Parameter Name	TcpIpDhcpV6FQDNOptionEnabled	
Description	<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"> ▶ true: Enables the transmission of the hostname option. ▶ false: Disables the transmission of the hostname option. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ RAM reduction: Disabling this parameter reduces the RAM consumption of the module. ▶ Execution time reduction (code): Disabling this parameter reduces the execution time of the module code. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpDhcpV6ConfigurableOptionsEnabled	
Description	Enables (TRUE) or disables (FALSE) Dhcp options	
Optimization Effect:	<ul style="list-style-type: none"> ▶ ROM reduction (config): Disabling this parameter reduces the ROM consumption of the module configuration. ▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code. ▶ RAM reduction: Disabling this parameter reduces the RAM consumption of the module. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	
Parameter Name	TcpIpDhcpV6ConfigurableOptionsEntriesMax	



Description	Maximum number of DhcpV6 option entries in PostBuild configuration. This parameter has impact on the size of the PostBuild RAM.
	Optimization Effect: <ul style="list-style-type: none"> ▶ ROM reduction (config): Selecting a small value for this parameter reduces the ROM consumption of the module configuration. ▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	INTEGER
Configuration class	PreCompile: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpDhcpV6ConfigurableOptionsDataSizeMax	
Description	Maximum data size of all DhcpV6 option entries in PostBuild configuration. This parameter has impact on the size of the PostBuild RAM.	
	Optimization Effect: <ul style="list-style-type: none"> ▶ ROM reduction (config): Selecting a small value for this parameter reduces the ROM consumption of the module configuration. ▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module. 	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	PreCompile:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpDhcpV6SlaacDadEnabled	
Description	Enable DhcpV6 Duplicate Address Detection (DAD) according to RFC4862 and RFC3315.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	Elektrobit Automotive GmbH
---------------	----------------------------

5.8.1.26. TcplpDhcpV6ConfigurableOptions

Parameters included	
Parameter name	Multiplicity
TcplpDhcpV6OptionCode	1..1
TcplpDhcpV6OptionMaxLength	1..1
TcplpDhcpV6OptionTransmit	1..1

Parameter Name	TcplpDhcpV6OptionCode	
Description	Code of the DhcpV6 option	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpDhcpV6OptionMaxLength	
Description	Memory which shall be reserved for this DhcpV6 option	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65535 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpDhcpV6OptionTransmit	
Description	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	
Multiplicity	1..1	
Type	BOOLEAN	



Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.27. TcplplcmpV6Config

Containers included		
Container name	Multiplicity	Description
TcplplcmpV6MsgHandler	0..1	This container is a subcontainer of TcplplcmpConfig and specifies the configuration parameters for the ICMPv6 message handler.

Parameters included	
Parameter name	Multiplicity
TcplplcmpV6EchoReplyEnabled	1..1
TcplplcmpV6EchoReplyToMulticastEnabled	1..1
TcplplcmpV6HopLimit	1..1
TcplplcmpV6MsgDestinationUnreachableEnabled	1..1
TcplplcmpV6MsgParameterProblemEnabled	1..1
TcplplcmpV6EchoReplyAvoidFragmentation	1..1

Parameter Name	TcplplcmpV6EchoReplyEnabled
Description	If enabled, the stack will respond to incoming ICMPv6 Echo Requests (Pings).
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcplplcmpV6EchoReplyToMulticastEnabled
Description	If enabled, the stack will respond to incoming ICMPv6 Echo Requests (Pings) addressed to multicast address.
Multiplicity	1..1
Type	BOOLEAN



Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIplcmpV6HopLimit	
Description	Default Hop-Limit value of outgoing ICMPv6 packets.	
Multiplicity	1..1	
Type	INTEGER	
Default value	255	
Range	<=255	>=0
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIplcmpV6MsgDestinationUnreachableEnabled	
Description	Dis/Enables transmission of Destination Unreachable Messages	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIplcmpV6MsgParameterProblemEnabled	
Description	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.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIplcmpV6EchoReplyAvoidFragmentation	
Description	<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.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.8.1.28. TcplIcmpV6MsgHandler

Parameters included	
Parameter name	Multiplicity
TcplIcmpV6MsgHandlerHeaderFileName	1..1
TcplIcmpV6MsgHandlerName	1..1

Parameter Name	TcplIcmpV6MsgHandlerHeaderFileName
Description	This parameter specifies the name of the header file containing the definition of the ICMPv6 message handler function.
Multiplicity	1..1
Type	STRING
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcplIcmpV6MsgHandlerName
Description	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)
Multiplicity	1..1
Type	FUNCTION-NAME
Configuration class	VariantPostBuild: VariantPostBuild



Origin	AUTOSAR_ECUC
--------	--------------

5.8.1.29. TcpllpV6FragmentationConfig

Parameters included	
Parameter name	Multiplicity
TcpllpV6ReassemblyBufferCount	1..1
TcpllpV6ReassemblyBufferSize	0..1
TcpllpV6ReassemblySegmentCount	0..1
TcpllpV6ReassemblyTimeout	0..1
TcpllpV6TxFragmentBufferCount	1..1
TcpllpV6TxFragmentBufferSize	0..1

Parameter Name	TcpllpV6ReassemblyBufferCount	
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	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpllpV6ReassemblyBufferSize	
Description	[RFC2460 5. Packet Size Issues]	
Multiplicity	0..1	
Type	INTEGER	
Default value	1500	
Range	<=65535 >=1500	
Configuration class	PreCompile:	VariantPostBuild



Origin	AUTOSAR_ECUC
---------------	--------------

Parameter Name	TcpIplpV6ReassemblySegmentCount	
Description	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.	
Multiplicity	0..1	
Type	INTEGER	
Default value	5	
Range	<=255 >=1	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIplpV6ReassemblyTimeout	
Description	[RFC2460 4.5 Fragment Header]	
Multiplicity	0..1	
Type	FLOAT	
Default value	60.0	
Range	<=100.0 >=0.001	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIplpV6TxFragmentBufferCount	
Description	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.	
Multiplicity	1..1	
Type	INTEGER	
Default value	2	
Range	<=1000 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	AUTOSAR_ECUC
Parameter Name	TcpIpV6TxFragmentBufferSize
Description	Size of each fragment tx buffer in bytes
Multiplicity	0..1
Type	INTEGER
Default value	1500
Range	<=65535 >=1500
Configuration class	PreCompile: VariantPostBuild
Origin	AUTOSAR_ECUC

5.8.1.30. TcpIpNdpConfig

Containers included		
Container name	Multiplicity	Description
TcIpNdpArNudConfig	0..1	Specifies the configuration parameters for NDP Address Resolution and Neighbor Unreachability Detection.
TcIpNdpPrefixRouterDiscoveryConfig	0..1	Specifies the configuration parameters for NDP Prefix and Router Discovery.
TcIpNdpSlaacConfig	0..1	Specifies the configuration parameters for StateLess Address AutoConfiguration.

5.8.1.31. TcIpNdpArNudConfig

Parameters included	
Parameter name	Multiplicity
TcIpNdpDefaultReachableTime	1..1
TcIpNdpDefaultRetransTimer	1..1
TcIpNdpDelayFirstProbeTime	1..1
TcIpNdpMaxNeighborCacheSize	1..1
TcIpNdpMaxRandomFactor	1..1
TcIpNdpMinRandomFactor	1..1



Parameters included

TcplpNdpNeighborUnreachabilityDetectionEnabled	1..1
TcplpNdpNumMulticastSolicitations	1..1
TcplpNdpNumUnicastSolicitations	1..1
TcplpNdpPacketQueueEnabled	1..1
TcplpNdpRandomReachableTimeEnabled	1..1
TcplpNdpDefensiveProcessing	1..1

Parameter Name	TcplpNdpDefaultReachableTime	
Description	Configuration of the ReachableTime (s) specified in [RFC4861 6.3.2. Host Variables].	
Multiplicity	1..1	
Type	FLOAT	
Default value	30.0	
Range	<=120.0 >=0.0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplpNdpDefaultRetransTimer	
Description	Configures the default value (s) for the RetransTimer variable specified in [RFC4861 6.3.2. Host Variables].	
Multiplicity	1..1	
Type	FLOAT	
Default value	1.0	
Range	<=60.0 >=0.0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplpNdpDelayFirstProbeTime	
Description	Delay before sending the first NUD probe in (s).	
Multiplicity	1..1	
Type	FLOAT	



Default value	5.0
Range	≤ 60.0 ≥ 0.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpNdpMaxNeighborCacheSize
Description	Maximum number of entries in the neighbor cache.
Multiplicity	1..1
Type	INTEGER
Default value	5
Range	≤ 254 ≥ 1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpNdpMaxRandomFactor
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i>
	Maximum random factor used for randomization
Multiplicity	1..1
Type	INTEGER
Default value	15
Range	≤ 100 ≥ 0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpNdpMinRandomFactor
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i>
	Minimum random factor used for randomization"
Multiplicity	1..1



Type	INTEGER
Default value	5
Range	<=100 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpNdpNeighborUnreachabilityDetectionEnabled
Description	<p>Note: Neighbor Unreachability Detection is always turned on per default and cannot be turned off.</p> <p>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.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpNdpNumMulticastSolicitations
Description	Maximum number of multicast solicitations that will be sent when performing address resolution.
Multiplicity	1..1
Type	INTEGER
Default value	3
Range	<=255 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpNdpNumUnicastSolicitations
Description	Maximum number of unicast solicitations that will be sent when performing Neighbor Unreachability Detection.
Multiplicity	1..1
Type	INTEGER



Default value	3
Range	<=255 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpNdpPacketQueueEnabled
Description	<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.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpNdpRandomReachableTimeEnabled
Description	<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
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpNdpDefensiveProcessing
Description	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.



Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.8.1.32. TcplpNdpPrefixRouterDiscoveryConfig

Containers included		
Container name	Multiplicity	Description
TcplpNdpPrefixList	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>

Parameters included	
Parameter name	Multiplicity
TcplpNdpDefaultRouterListSize	1..1
TcplpNdpDestinationCacheSize	1..1
TcplpNdpDestinationCacheEnabled	1..1
TcplpNdpDynamicHopLimitEnabled	1..1
TcplpNdpDynamicMtuEnabled	1..1
TcplpNdpDynamicReachableTimeEnabled	1..1
TcplpNdpDynamicRetransTimeEnabled	1..1
TcplpNdpMaxRtrSolicitationDelay	1..1
TcplpNdpMaxRtrSolicitations	1..1
TcplpNdpPrefixListSize	1..1
TcplpNdpRndRtrSolicitationDelayEnabled	1..1
TcplpNdpRtrSolicitationInterval	1..1

Parameter Name	TcplpNdpDefaultRouterListSize
Description	value="Maximum number of default router entries."
Multiplicity	1..1



Type	INTEGER	
Default value	2	
Range	<=254 >=2	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpNdpDestinationCacheSize	
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	TcpIpNdpDestinationCacheEnabled	
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	TcpIpNdpDynamicHopLimitEnabled	
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	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	TcpIpNdpDynamicMtuEnabled	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Allow dynamic reconfiguration of link MTU via Router Advertisements.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	TcpIpNdpDynamicReachableTimeEnabled	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>If enabled the default Reachable Time value may be reconfigured based on received Router Advertisements.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	TcpIpNdpDynamicRetransTimeEnabled	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>If enabled the default Retransmit Timer value may be reconfigured based on received Router Advertisements.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	



Parameter Name	TcpIpNdpMaxRtrSolicitationDelay	
Description	Maximum delay before the first Router Solicitation will be sent after interface initialization in (s).	
Multiplicity	1..1	
Type	FLOAT	
Default value	1.0	
Range	<=60.0 >=0.001	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpNdpMaxRtrSolicitations	
Description	Maximum number of Router Solicitations that will be sent before the first Router Advertisement has been received.	
Multiplicity	1..1	
Type	INTEGER	
Default value	3	
Range	<=255 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpNdpPrefixListSize	
Description	Maximum number of entries in the on-link prefix list.	
Multiplicity	1..1	
Type	INTEGER	
Default value	5	
Range	<=254 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpNdpRndRtrSolicitationDelayEnabled	
-----------------------	---	--



Description	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.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpNdpRtrSolicitationInterval	
Description	Interval between consecutive Router Solicitations in (s).	
Multiplicity	1..1	
Type	FLOAT	
Default value	4.0	
Range	<=60.0 >=0.001	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.8.1.33. TcpIpNdpPrefixList

Containers included		
Container name	Multiplicity	Description
TcplpNdpPrefixListEntry	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

Parameters included	
Parameter name	Multiplicity
TcplpNdpPrefixListEntryPrefixAddress	1..1



Parameters included

TcplpNdpPrefixListEntryPrefixLength	1..1
---	------

Parameter Name	TcplpNdpPrefixListEntryPrefixAddress
Description	<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
Multiplicity	1..1
Type	STRING
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcplpNdpPrefixListEntryPrefixLength
Description	<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
Multiplicity	1..1
Type	INTEGER
Range	<=128 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.8.1.35. TcplpNdpSlaacConfig

Parameters included	
Parameter name	Multiplicity
TcplpNdpSlaacDadNumberOfTransmissions	1..1
TcplpNdpSlaacDadRetransmissionDelay	1..1
TcplpNdpSlaacDelayEnabled	1..1
TcplpNdpSlaacOptimisticDadEnabled	1..1

Parameter Name	TcplpNdpSlaacDadNumberOfTransmissions
----------------	---------------------------------------



Description	"Number of Neighbor Solicitations that have to be unanswered in order to set an autoconfigurated address to PREFERRED (usable) state.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=254 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	TcpIpNdpSlaacDadRetransmissionDelay	
Description	Sets the maximum value for the address configuration delay (s)	
Multiplicity	1..1	
Type	FLOAT	
Default value	1.0	
Range	<=10.0 >=0.0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	TcpIpNdpSlaacDelayEnabled	
Description	If enabled transmission of the first DAD Neighbor Solicitation will be delayed by a random value from [0...MAX_DAD_DELAY].	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	TcpIpNdpSlaacOptimisticDadEnabled	
Description	Enable Optimistic Duplicate Address Detection (DAD) according to RFC4429.	
Multiplicity	1..1	
Type	BOOLEAN	



Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.8.1.36. TcplIpv6ConfigExtHeaderFilter

Parameters included	
Parameter name	Multiplicity
TcplIpv6ConfigExtHeaderFilterEntry	1..n
Parameter Name	
Description	IPv6 Extension Header type allowed by this filter
Multiplicity	1..n
Type	INTEGER
Range	<=255 >=0
Configuration class	PreCompile: VariantPostBuild
Origin	AUTOSAR_ECUC

5.8.1.37. TcplpLocalAddr

Containers included		
Container name	Multiplicity	Description
TcplpAddrAssignment	1..n	This container is a subcontainer of TcplpLocalAddr and specifies the assignment policy for the IP address.
TcplpStaticIpAddressConfig	0..1	This container is a subcontainer of TcplpLocalAddr and specifies a static IP address including directly related parameters.

Parameters included	
Parameter name	Multiplicity
TcplpAddrId	1..1
TcplpAddressType	1..1
TcplpDomainType	1..1



Parameters included

TcplpCtrlRef	1..1
TcplpLocalAddrIpv6ExtHeaderFilterRef	0..1

Parameter Name	TcplpAddrId	
Description	IP address table identifier assigned by TCP/IP stack.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=65535 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplpAddressType	
Description	Address type <ul style="list-style-type: none"> ▶ TCPIP_UNICAST: Unicast address ▶ TCPIP_MULTICAST: Multicast address ▶ TCPIP_ANYCAST: Anycast address 	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	TCPIP_UNICAST	
Range	TCPIP_MULTICAST TCPIP_UNICAST TCPIP_ANYCAST	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplpDomainType	
Description	Address family <ul style="list-style-type: none"> ▶ TCPIP_AF_INET: IPv4 address ▶ TCPIP_AF_INET6: IPv6 address 	



Multiplicity	1..1
Type	ENUMERATION
Default value	TCPIP_AF_INET
Range	TCPIP_AF_INET TCPIP_AF_INET6
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpCtrlRef
Description	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.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpLocalAddrIPv6ExtHeaderFilterRef
Description	Reference to a set of IPv6 Extension Headers which are allowed for this local IPv6 address
Multiplicity	0..1
Type	REFERENCE
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.8.1.38. TcpIpAddrAssignment

Parameters included	
Parameter name	Multiplicity
TcIpAssignmentLifetime	0..1
TcIpAssignmentMethod	1..1
TcIpAssignmentPriority	1..1
TcIpAssignmentTrigger	1..1

**Parameters included**

TcplpUseSimpleDhcpClient	1..1
--	------

Parameter Name	TcplpAssignmentLifetime
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> Defines the lifetime of a dynamically fetched IP address.
Multiplicity	0..1
Type	ENUMERATION
Default value	TCPIP_FORGET
Range	TCPIP_FORGET TCPIP_STORE
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECU

Parameter Name	TcplpAssignmentMethod
Description	Address type <ul style="list-style-type: none"> ▶ TCPIP_STATIC: <ul style="list-style-type: none"> ▶ TCPIP_AF_INET: Assignment of static IPv4 address ▶ TCPIP_AF_INET6: Assignment of static IPv6 address ▶ TCPIP_LINKLOCAL: <ul style="list-style-type: none"> ▶ TCPIP_AF_INET: Assignment of IPv4 Link Local address; timing according to IETF RFC 3927 - Dynamic Configuration of IPv4 Link-Local Addresses ▶ TCPIP_AF_INET6: Assignment of IPv6 Link Local address ▶ TCPIP_LINKLOCAL_DOIP: <ul style="list-style-type: none"> ▶ TCPIP_AF_INET: Assignment of IPv4 Link Local address; timing according to ISO 13400 (DoIP) ▶ TCPIP_AF_INET6: Not applicable ▶ TCPIP_DHCP: <ul style="list-style-type: none"> ▶ TCPIP_AF_INET: Assignment of IPv4 address through a DHCPv4 server ▶ TCPIP_AF_INET6: Assignment of IPv6 address through a DHCPv6 server



	<ul style="list-style-type: none"> ▶ TCPIP_IPV6_ROUTER: <ul style="list-style-type: none"> ▶ TCPIP_AF_INET: Not applicable ▶ TCPIP_AF_INET6: Assignment of IPv6 address through Router Advertisements
Multiplicity	1..1
Type	ENUMERATION
Default value	TCPIP_STATIC
Range	TCPIP_DHCP TCPIP_IPV6_ROUTER TCPIP_LINKLOCAL TCPIP_LINKLOCAL_DOIP TCPIP_STATIC
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpAssignmentPriority
Description	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.
Multiplicity	1..1
Type	INTEGER
Default value	1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpAssignmentTrigger
Description	Trigger of address assignment.
Multiplicity	1..1
Type	ENUMERATION
Default value	TCPIP_AUTOMATIC
Range	TCPIP_AUTOMATIC TCPIP_MANUAL
Configuration class	VariantPostBuild: VariantPostBuild



Origin	AUTOSAR_ECUC
Parameter Name	TcplpUseSimpleDhcpClient
Description	<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>
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.39. TcplpStaticIpConfig

Parameters included	
Parameter name	Multiplicity
TcplpDefaultRouter	0..1
TcplpNetmask	0..1
TcplpStaticIpAddress	1..1

Parameter Name	TcplpDefaultRouter
Description	IP address of default router (gateway) with following notation dependent on parameter TcplpDomainType: <ul style="list-style-type: none"> ▶ TCPIP_AF_INET: Four octets separated by character "." in decimal format ▶ TCPIP_AF_INET6: 8 block to 16bit separated by character ":" in hexadecimal format
Multiplicity	0..1
Type	STRING
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcplpNetmask
Description	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.
Multiplicity	0..1
Type	INTEGER
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcplpStaticIpAddress	
Description	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	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.8.1.40. TcplpNvmBlock

Parameters included		
Parameter name		Multiplicity
TcplpNvmBlockDescriptorRef		1..1
TcplpNvmBlockSize		1..1

Parameter Name	TcplpNvmBlockDescriptorRef	
Description	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.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	



Parameter Name	TcplpNvmBlockSize	
Description	This parameter defines the size of internal Tcplp cache dedicated to interaction with NvM [in units of four bytes].	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.41. TcplpPhysAddrConfig

Containers included		
Container name	Multiplicity	Description
TcplpPhysAddrChgHandler	0..1	This container is a subcontainer of TcplpPhysAddrConfig

5.8.1.42. TcplpPhysAddrChgHandler

Parameters included	
Parameter name	Multiplicity
TcplpPhysAddrChgHandlerHeaderFileName	1..1
TcplpPhysAddrChgHandlerName	1..1

Parameter Name	TcplpPhysAddrChgHandlerHeaderFileName	
Description	This parameter specifies the name of the header file containing the definition of the physical address change handler function.	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplpPhysAddrChgHandlerName	
Description	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)
Multiplicity	1..1
Type	FUNCTION-NAME
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.8.1.43. TcplpSocketOwnerConfig

Containers included		
Container name	Multiplicity	Description
TcplpSocketOwner	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
TcplpSocketOwnerHeaderFileName	0..1
TcplpSocketOwnerCopyTxDataName	0..1
TcplpSocketOwnerLocallpAddrAssignmentChgName	0..1
TcplpSocketOwnerRxIndicationName	0..1
TcplpSocketOwnerTcpAcceptedName	0..1
TcplpSocketOwnerTcpConnectedName	0..1
TcplpSocketOwnerTxConfirmationName	0..1
TcplpSocketOwnerTcplpEventName	0..1
TcplpSocketOwnerDtlsConnectionStateChgName	0..1
TcplpSocketOwnerUpperLayerType	1..1

Parameter Name	TcplpSocketOwnerHeaderFileName
Description	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.



Multiplicity	0..1
Type	STRING
Configuration class	Link: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpSocketOwnerCopyTxDataName
Description	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.
Multiplicity	0..1
Type	STRING
Configuration class	Link: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpSocketOwnerLocallpAddrAssignmentChgName
Description	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.
Multiplicity	0..1
Type	STRING
Configuration class	Link: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpSocketOwnerRxIndicationName
Description	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.
Multiplicity	0..1
Type	STRING
Configuration class	Link: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpSocketOwnerTcpAcceptedName
Description	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.



Multiplicity	0..1
Type	STRING
Configuration class	Link: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpSocketOwnerTcpConnectedName
Description	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.
Multiplicity	0..1
Type	STRING
Configuration class	Link: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpSocketOwnerTxConfirmationName
Description	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.
Multiplicity	0..1
Type	STRING
Configuration class	Link: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpSocketOwnerTcplpEventName
Description	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.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	Link: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpSocketOwnerDtlsConnectionStateChgName
Description	This parameter defines the name of the <Up_DtlsConnectionStateChg> function of the TcpIpSocketOwner module. The function name shall only be configurable if TcpIpSocketOwnerUpperLayerType is set to CDD.



Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	Link: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplpSocketOwnerUpperLayerType	
Description	This parameter specifies the type of the upper layer module.	
Multiplicity	1..1	
Type	ENUMERATION	
Range	CDD SOAD	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.8.1.45. TcplpTcpConfig

Containers included		
Container name	Multiplicity	Description
TcplpTcpConfigOptionFilter	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
TcplpTcpUnpredictableSequenceNumbers	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.
TcplpTcpSynCookies	0..1	Container for SYN cookies parameters

Parameters included	
Parameter name	Multiplicity
TcplpDelayedAckTimeout	1..1
TcplpTcpCongestionAvoidanceEnabled	1..1
TcplpTcpFastRecoveryEnabled	1..1
TcplpTcpFastRetransmitEnabled	1..1



Parameters included

TcplpTcpFinWait2Timeout	1..1
TcplpTcpKeepAliveEnabled	1..1
TcplpTcpKeepAliveDefault	1..1
TcplpTcpKeepAliveInterval	1..1
TcplpTcpKeepAliveProbesMax	1..1
TcplpTcpKeepAliveTime	1..1
TcplpTcpMaxRtx	0..1
TcplpTcpMsl	1..1
TcplpTcpNagleEnabled	1..1
TcplpTcpReceiveWindowMax	1..1
TcplpTcpRetransmissionTimeout	0..1
TcplpTcpSlowStartEnabled	1..1
TcplpTcpSynMaxRtx	1..1
TcplpTcpSynReceivedTimeout	1..1
TcplpTcpTtl	1..1
TcplpTcpOptionFilterEnabled	1..1
TcplpTcpDupAckTransmitTimeoutSeqNum	1..1
TcplpTcpDupAckTransmitTimeoutAckNum	1..1
TcplpTcpOutOfOrderReceptionEnabled	1..1
TcplpTcpOutOfOrderReceptionBufferCount	1..1
TcplpTcpOutOfOrderReceptionBufferSize	1..1
TcplpTcpOutOfOrderReceptionHoleListSize	1..1
TcplpTcpCopyWindowCheckEnabled	1..1
TcplpTcpTimeWaitRstIgnoreReceptionEnabled	1..1

Parameter Name	TcplpDelayedAckTimeout
Description	The maximal time an acknowledgment is delayed for transmission in seconds.
Multiplicity	1..1
Type	FLOAT
Default value	0.1
Range	>=0 <=0.5



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpTcpCongestionAvoidanceEnabled
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Enables or disables support of TCP congestion avoidance algorithm according to IETF RFC 5681.</p> <ul style="list-style-type: none"> ▶ true: Congestion avoidance algorithm enabled ▶ false: Congestion avoidance algorithm disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpTcpFastRecoveryEnabled
Description	<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"> ▶ true: Fast recovery algorithm enabled ▶ false: Fast recovery algorithm disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpTcpFastRetransmitEnabled
Description	<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"> ▶ true: Fast retransmissions algorithm enabled ▶ false: Fast retransmissions algorithm disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpTcpFinWait2Timeout
Description	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.
Multiplicity	1..1
Type	FLOAT
Default value	20.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpTcpKeepAliveEnabled
Description	Enables or disables TCP Keep Alive Probes according to IETF RFC 1122 chapter 4.2.3.6 <ul style="list-style-type: none"> ▶ true: Keep alive probes enabled ▶ false: Keep alive probes disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpTcpKeepAliveDefault
Description	Enables or disables TCP Keep Alive Probes for all sockets by default.
Multiplicity	1..1
Type	BOOLEAN
Default value	false



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpTcpKeepAliveInterval	
Description	Specifies the interval in [s] between subsequent keepalive probes.	
Multiplicity	1..1	
Type	FLOAT	
Default value	7200	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpTcpKeepAliveProbesMax	
Description	Maximum number of times that a TCP Keep Alive is retransmitted before the connection is closed.	
Multiplicity	1..1	
Type	INTEGER	
Default value	10	
Range	>=1 <=255	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpTcpKeepAliveTime	
Description	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.	
Multiplicity	1..1	
Type	FLOAT	
Default value	7200	
Range	>=0.1 <=65535	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	



Parameter Name	TcpIpTcpMaxRtx	
Description	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.	
Multiplicity	0..1	
Type	INTEGER	
Default value	12	
Range	<=255	>=0
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	
Parameter Name	TcpIpTcpMsl	
Description	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.)	
Multiplicity	1..1	
Type	FLOAT	
Default value	120.0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	
Parameter Name	TcpIpTcpNagleEnabled	
Description	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"> ▶ true: Nagle's algorithm enabled ▶ false: Nagle's algorithm disabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	
Parameter Name	TcpIpTcpReceiveWindowMax	



Description	Default value of maximum receive window in bytes.	
Multiplicity	1..1	
Type	INTEGER	
Default value	4380	
Range	>=1 <=65535	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpTcpRetransmissionTimeout	
Description	Timeout in [s] before an unacknowledged TCP segment is sent again.	
Multiplicity	0..1	
Type	FLOAT	
Default value	3.5	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpTcpSlowStartEnabled	
Description	<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"> ▶ true: Slow start algorithm enabled ▶ false: Slow start algorithm disabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpTcpSynMaxRtx	
Description	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.
Multiplicity	1..1
Type	INTEGER
Default value	4
Range	<=255 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcplpTcpSynReceivedTimeout
Description	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.
Multiplicity	1..1
Type	FLOAT
Default value	0.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcplpTcpTtl
Description	Default Time-to-live value of outgoing TCP packets
Multiplicity	1..1
Type	INTEGER
Default value	255
Range	>=1 <=255
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcplpTcpOptionFilterEnabled
Description	Enables (TRUE) or disables (FALSE) filtering of TCP options
Multiplicity	1..1



Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpTcpDupAckTransmitTimeoutSeqNum
Description	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.
Multiplicity	1..1
Type	FLOAT
Default value	0
Range	<=1.0 >=0.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpTcpDupAckTransmitTimeoutAckNum
Description	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.
Multiplicity	1..1
Type	FLOAT
Default value	0
Range	<=1.0 >=0.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpTcpOutOfOrderReceptionEnabled
Description	Enables (TRUE) or disables (FALSE) support for out-of-order reception of TCP segments.
Multiplicity	1..1



Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpTcpOutOfOrderReceptionBufferCount	
Description	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.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	≥ 1 ≤ 32764	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpTcpOutOfOrderReceptionBufferSize	
Description	Sets the size in octets per buffer that can be used for the reception of out-of-order TCP segments.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1024	
Range	≥ 1 ≤ 65535	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpTcpOutOfOrderReceptionHoleListSize	
Description	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.	
Multiplicity	1..1	
Type	INTEGER	



Default value	8
Range	>=1 <=254
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplpTcpCopyWindowCheckEnabled
Description	Determines the behavior of Tcplp_TcpTransmit when it is called with the parameter forceRetrieve set to FALSE. When copying data from the upper layer to be transmitted over TCP, Tcplp selects the amount of data to be copied based on: <ul style="list-style-type: none"> ▶ true: the minimum of the window size of the remote host and the available buffer signaled by the upper layer. ▶ false: the available buffer signaled by the upper layer.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplpTcpTimeWaitRstIgnoreReceptionEnabled
Description	Enables (TRUE) or disables (FALSE) support for ignoring the reception of TCP RST segments.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.46. TcplpTcpConfigOptionFilter

Parameters included	
Parameter name	Multiplicity
TcplpTcpConfigOptionFilterEntry	1..n



Parameters included

[TcplpTcpConfigOptionFilterId](#)

1..1

Parameter Name	TcplpTcpConfigOptionFilterEntry	
Description	TCP option kind allowed by this filter	
Multiplicity	1..n	
Type	INTEGER	
Range	>=0 <=255	
Configuration class	PostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplpTcpConfigOptionFilterId	
Description	Identification of the TCP option filter	
Multiplicity	1..1	
Type	INTEGER	
Range	>=0 <=255	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.8.1.47. TcplpTcpUnpredictableSeqNumbers

Parameters included

Parameter name

Multiplicity

[TcplpTcpUnpredictableSeqNumbersKeyGenerateJobId](#)

1..1

[TcplpTcpUnpredictableSeqNumbersKeyResetTime](#)

1..1

Parameter Name	TcplpTcpUnpredictableSeqNumbersKeyGenerateJobId	
Description	Reference to Csm_MacGenerate job which is used to generate unpredictable value for initial TCP sequence number secret.	
Multiplicity	1..1	
Type	REFERENCE	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpTcpUnpredictableSeqNumbersKeyResetTime	
Description	Time period for resetting the key used for generating unpredictable TCP initial sequence numbers in seconds.	
Multiplicity	1..1	
Type	FLOAT	
Range	<=1000 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.48. TcpIpTcpSynCookies

Parameters included	
Parameter name	Multiplicity
TcplpTcpSynCookiesKey1GenerateJobId	1..1
TcplpTcpSynCookiesKey2GenerateJobId	1..1
TcplpTcpSynCookiesTimeResetKeys	1..1
TcplpTcpSynCookiesAcceptAckOverflowTime	1..1
TcplpTcpSynCookiesTimebaseRef	1..1

Parameter Name	TcplpTcpSynCookiesKey1GenerateJobId
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	VariantPostBuild:
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplpTcpSynCookiesKey2GenerateJobId
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	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpTcpSynCookiesTimeResetKeys	
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	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpTcpSynCookiesAcceptAckOverflowTime	
Description	Time (in seconds [s]) after SYN reception overflow (no free entry for creating new SYN-RECEIVED socket) after which ACK can be accepted.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=128 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpTcpSynCookiesTimebaseRef	
Description	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.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	Elektrobit Automotive GmbH
---------------	----------------------------

5.8.1.49. TcplpUdpConfig

Parameters included	
Parameter name	Multiplicity
TcplpUdpTtl	1..1
TcplpUdpMayReFragment	1..1

Parameter Name	TcplpUdpTtl	
Description	Default Time-to-live value of outgoing UDP packets.	
Multiplicity	1..1	
Type	INTEGER	
Default value	255	
Range	<=255 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplpUdpMayReFragment	
Description	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 bandwidth.	
	<ul style="list-style-type: none"> ▶ true: UDP Datagram may be fragmented ▶ false: UDP Datagram may not be fragmented 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	



Configuration class	PostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.50. TcplpDuplicateAddressDetectionConfig

Parameters included	
Parameter name	Multiplicity
TcplpDuplicateAddressDetectionCalloutName	1..1
TcplpDuplicateAddressDetectionHeaderFileName	1..1

Parameter Name	TcplpDuplicateAddressDetectionCalloutName
Description	This parameter defines the name of the DAD callout function Up_DADAddress-Conflict.
Multiplicity	1..1
Type	FUNCTION-NAME
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECU

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:
Origin	AUTOSAR_ECU

5.8.1.51. TcplpRxPolicyCheckIngressHandlerConfig

Parameters included	
Parameter name	Multiplicity
TcplpRxPolicyCheckIngressHandlerName	1..1
TcplpRxPolicyCheckIngressHandlerHeaderFileName	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	TcpIpRxPolicyCheckIngressHandlerHeaderFileName	
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. TcpIpRxPolicyCheckRoutingHandlerConfig

Parameters included		
Parameter name	Multiplicity	
TcIpRxPolicyCheckRoutingHandlerName	1..1	
TcIpRxPolicyCheckRoutingHandlerHeaderFileName	1..1	

Parameter Name	TcpIpRxPolicyCheckRoutingHandlerName	
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	TcpIpRxPolicyCheckRoutingHandlerHeaderFileName	
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
TcplpRxPacketPostProcessHandlerName	1..1
TcplpRxPacketPostProcessHandlerHeaderFileName	1..1

Parameter Name	TcplpRxPacketPostProcessHandlerName	
Description	This parameter defines the name of the RxPacketPostProcessHandler callout function.	
Multiplicity	1..1	
Type	FUNCTION-NAME	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpRxPacketPostProcessHandlerHeaderFileName	
Description	This parameter specifies the header file name containing the definition of RxPacketPostProcessHandler callout function.	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.54. TcplpRxPacketDropHandlerConfig

Parameters included	
Parameter name	Multiplicity
TcplpRxPacketDropHandlerName	1..1
TcplpRxPacketDropHandlerHeaderFileName	1..1



Parameter Name	TcplpRxPacketDropHandlerName	
Description	This parameter defines the name of the RxPacketDropHandler callout function.	
Multiplicity	1..1	
Type	FUNCTION-NAME	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpRxPacketDropHandlerHeaderFileName	
Description	This parameter specifies the header file name containing the definition of Rx-PacketDropHandler callout function.	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.55. TcplpMemoryConfig

Containers included		
Container name	Multiplicity	Description
TcplpMemoryPool	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
TcplpMemoryBlockSize	1..1
TcplpMemoryBlockCount	1..1

Parameter Name	TcplpMemoryBlockSize
Description	This parameter defines the size of a memory block in bytes.



Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpMemoryBlockCount	
Description	This parameter defines the number of available memory blocks with the size specified by TcpIpMemoryBlockSize.	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65535 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.57. TcpllpSecConfig

Containers included		
Container name	Multiplicity	Description
TcpllpSecGeneral	1..1	General pre-compile configuration parameters for IPsec.
TcpllpIkeGeneral	1..1	General pre-compile configuration parameters for IKEv2.
TcpllpSecConnections	1..255	This container contains the list of secured connections to the remote hosts. If TcplpSecurityMode is not equal to NO_SECURITY: Transmissions and Receptions of packets will be blocked for hosts which are not listed.
TcpllpSecSecurityAssociationCommonCfg	1..n	This container contains the configuration of the IPsec security associations.
TcpllpIkeSecurityAssociationCommonCfg	0..n	This container contains the configuration of the IKEv2 security associations.
TcpllpIkeCertificate	0..255	This container contains the list of all configured certificates.
TcpllpIkeIdentifiers	0..n	This container contains the list of ID_DER ASN1 DN identifiers used to identify the remote hosts.
TcpllpSecReportErrorHandler	0..1	This container contains the configuration of the TcpllpSecReportErrorHandler function.



5.8.1.58. TcpllpSecGeneral

Parameters included	
Parameter name	Multiplicity
TcpllpSecAntiReplayMaxWindowSize	1..1
TcpllpSecGmacEnable	1..1
TcplpMaxNumIpsecConnections	1..1
TcplpMaxNumIpsecSecurityAssociation	1..1
TcplpMaxNumIpsecSecurityAssociationConfigurations	1..1
TcplpIcvSizeMax	1..1
TcpllpSecRemotePhysAddrCheckEnable	1..1
TcpllpSecRandomGenerateJobRef	1..1

Parameter Name	TcpllpSecAntiReplayMaxWindowSize	
Description	Size of the anti-replay window	
Multiplicity	1..1	
Type	INTEGER	
Default value	64	
Range	<=65534 >=32	
Configuration class	PreCompile:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpllpSecGmacEnable	
Description	Enables/Disables AES GMAC. ▶ true: AES GMAC is enabled. ▶ false: AES GMAC is disabled.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpMaxNumIpsecConnections	



Description	Total number of IPSec connections (number of entries in the list TcplpIpSecConnections).	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpMaxNumIpSecSecurityAssociation	
Description	Total number of available IPSec security associations (sum of entries in TcplpIpSecSecurityAssociationCfg for all entries in TcplpIpSecSecurityAssociationCommonCfg where each proposal counts only once).	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	PreCompile:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpMaxNumIpSecSecurityAssociationConfigurations	
Description	Maximal number of IPSec security association configurations (number of entries in the list TcplpIpSecSecurityAssociationCommonCfg).	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	PreCompile:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpIcvSizeMax	
Description	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.	
	<ul style="list-style-type: none"> ▶ AES_CMAC_96 = 12 bytes ▶ HMAC_SHA2_256_128 = 16 bytes ▶ AES_GMAC_128 = 16 bytes ▶ AES_GMAC_256 = 16 bytes 	
Multiplicity	1..1	
Type	INTEGER	
Default value	16	



Range	12
	16
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIpSecRemotePhysAddrCheckEnable
Description	Enables (FALSE) or disables (TRUE) filtering by remote MAC address.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIpSecRandomGenerateJobRef
Description	Reference to Csm_RandomGenerate job which is used to generate random numbers (e.g. for nonces).
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.59. TcplpIkeGeneral

Containers included		
Container name	Multiplicity	Description
TcplpIkeSignatureCfg	1..1	Settings related to IKEv2 optional usage of certificates on local/remote host and the hash algorithm used for detecting re-transmitted IKEv2 requests.
TcplpIkeFragmentationCfg	1..1	Settings related to IKEv2 message fragmentation as defined in RFC 7383.

Parameters included	
Parameter name	Multiplicity
TcplpIkeDeadPeerDetection	1..1



Parameters included

TcplIkeDeadPeerDetectionDelay	1..1
TcplIkeAeadSupportEnabled	1..1
TcplIkeEncryptSupportEnabled	1..1
TcplpMaxNumIkeSecurityAssociation	1..1
TcplpMaxNumIkeConfigurations	1..1
TcplpMaxNumIkeConnections	1..1
TcplpMaxCombinedNonceSize	1..1
TcplpMaxMacSize	1..1
TcplpMaxGeneratedKeySize	1..1
TcplpMaxDiffieHellmanKeySize	1..1
TcplIkeRxBufferSize	1..1
TcplIkeRxBufferSizeMaxMsgSize	1..1

Parameter Name	TcplIkeDeadPeerDetection
Description	Enables/Disables dead peer detection <ul style="list-style-type: none"> ▶ true: Dead peer detectione is enabled. ▶ false: Dead peer detectione is disabled.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplIkeDeadPeerDetectionDelay
Description	Specifies the time delay (in seconds) after which a dead-peer-detection request is transmitted if no traffic is received.
Multiplicity	1..1
Type	FLOAT
Default value	15.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplIkeAeadSupportEnabled



Description	Enable or disable support of Csm_AEADEncrypt and Csm_AEADDencrypt API. This API is required for AES_GCM encryption algorithm.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpIkeEncryptSupportEnabled	
Description	Enable or disable support of Csm_Encrypt and Csm_Dencrypt API. This API is required for AES_CBC encryption algorithm.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpMaxNumIkeSecurityAssociation	
Description	Total number of available IKEv2 security associations (sum of entries in TcplpIkeSecurityAssociationCfg for all entries in TcplpIkeSecurityAssociationCommonCfg where each proposal counts only once).	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	PostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpMaxNumIkeConfigurations	
Description	Maximal number of IKEv2 security association configurations (number of entries in the list TcplpIkeSecurityAssociationCommonCfg).	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	PostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpMaxNumIkeConnections	
-----------------------	----------------------------------	--



Description	Total number of IKEv2 connections (number of entries in the list TcplpIpSecConnections that have TcplpIkeConnectionCfg enabled).	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	PostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpMaxCombinedNonceSize	
Description	Indicates the maximal size of the combination of local nonce and remote nonce. It should be equal to the largest key size of all locally and remotely proposed pseudo random functions.	
Multiplicity	1..1	
Type	INTEGER	
Default value	64	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpMaxMacSize	
Description	Indicates the maximal size of a generated MAC in bytes. It should be equal to the output size of the largest pseudorandom function in use. <ul style="list-style-type: none"> ▶ PRF_HMAC_SHA2_384_NO6 = 48 bytes ▶ PRF_HMAC_SHA2_256_NO5 = 32 bytes 	
Multiplicity	1..1	
Type	INTEGER	
Default value	48	
Range	32 48	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpMaxGeneratedKeySize	
Description	Indicates the maximal size of all generated keys.	
Multiplicity	1..1	
Type	INTEGER	



Default value	48
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpMaxDiffieHellmanKeySize	
Description	Indicates the maximal Diffie Hellman key size. It should be equal to the key size configured for TcplpIkeDiffieHellmanExchangeKeyRef.	
	<ul style="list-style-type: none"> ▶ DH_256_BIT_RANDOM_ECP_GROUP_NO19 = 32 bytes ▶ DH_384_BIT_RANDOM_ECP_GROUP_NO20 = 48 bytes 	
Multiplicity	1..1	
Type	INTEGER	
Default value	48	
Range	32	48
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpIkeRxBufferSize	
Description	Number of IKEv2 messages that can be stored in the IKEv2 RX-Buffer. Each fragment of a fragmented IKEv2 messages counts as a separate message.	
Multiplicity	1..1	
Type	INTEGER	
Default value	8	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpIkeRxBufferSizeMaxMsgSize	
Description	Maximum size of an IKEv2 message that can be stored in the IKEv2 RX-Buffer. If TcplpIkeFragmentationRxEnabled is set to FALSE the value of TcplpIkeRxBufferSizeMaxMsgSize shall be set to the MTU of the connection path (1500 in case of ethernet).	
Multiplicity	1..1	
Type	INTEGER	
Default value	1500	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.60. TcplpIkeSignatureCfg

Containers included		
Container name	Multiplicity	Description
TcplpIkeSignatureVerify	1..n	This container contains a list of IKEv2 signature verify algorithms.

Parameters included	
Parameter name	Multiplicity
TcplpIkeLocalCertEnabled	1..1
TcplpIkeRemoteCertEnabled	1..1
TcplpMaxDigitalSignatureSize	1..1
TcplpIkeHashAlgorithm	1..1
TcplpIkeHashJobRef	1..1
TcplpIkeMaxCertChainSize	1..1
TcplpIkeMaxCertSubjectSize	1..1
TcplpIkeIdentifierItemTotalLength	1..1
TcplpIkeMaxExtractedObjects	1..1
TcplpIkeTotalExtractedObjects	1..1

Parameter Name	TcplpIkeLocalCertEnabled
Description	Enables (TRUE) or disables (FALSE) IKEv2 usage of certificates on local host.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild:
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplpIkeRemoteCertEnabled
Description	Enables (TRUE) or disables (FALSE) IKEv2 usage of certificates on remote host.



Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpMaxDigitalSignatureSize
Description	Indicates the maximal size of the digital signature in bytes. It should be greater or equal to the biggest configured TcplpIkeSignatureGenerateAlgorithm. <ul style="list-style-type: none"> ▶ ECDSA256 = 64 bytes ▶ ECDSA384 = 96 bytes ▶ ECDSA521 = 132 bytes
Multiplicity	1..1
Type	INTEGER
Default value	64
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplpIkeHashAlgorithm
Description	Indicates the algorithm used for hashing <ul style="list-style-type: none"> ▶ true: SHA1 hash will be used for message comparison and Certificate request will be calculated in runtime. ▶ false: SHA256 hash will be used for message comparison Certificate request will be pre-configured.
Multiplicity	1..1
Type	ENUMERATION
Range	SHA1 SHA256
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplpIkeHashJobRef
Description	Reference to Csm_Hash job defined by TcplpIkeHashAlgorithm
Multiplicity	1..1



Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	
Parameter Name	TcpIpIkeMaxCertChainSize	
Description	Indicates the maximal size of certificate chain that can be received	
Multiplicity	1..1	
Type	INTEGER	
Default value	3	
Range	<=4 >=1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	
Parameter Name	TcpIpIkeMaxCertSubjectSize	
Description	Indicates the maximal size of the certificate subject used for ID verification.	
Multiplicity	1..1	
Type	INTEGER	
Default value	300	
Range	<=65535 >1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	
Parameter Name	TcpIpIkeIdentifierItemTotalLength	
Description	Indicates the total length of all extracted identifiers	
Multiplicity	1..1	
Type	INTEGER	
Range	<=65535 >0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	
Parameter Name	TcpIpIkeMaxExtractedObjects	



Description	Indicates the maximum number of extracted remote ID items for one remote identifier	
Multiplicity	1..1	
Type	INTEGER	
Range	<=255 >0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpIkeTotalExtractedObjects	
Description	Indicates the total number of extracted remote ID items	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.61. TcpIpIkeSignatureVerify

Parameters included		
Parameter name		Multiplicity
TcpIpIkeSignatureVerifyAlgorithm		1..1
TcpIpIkeSignatureVerifyJobRef		1..1

Parameter Name	TcpIpIkeSignatureVerifyAlgorithm	
Description	The algorithm which is used to verify the signature. MUST match the hash algorithm defined in the CSM primitive of TcplIkeSignatureVerifyJobRef.	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	ECDSA_WITH_SHA256	
Range	ECDSA_WITH_SHA256 ECDSA_WITH_SHA384 ECDSA_WITH_SHA512	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	Elektrobit Automotive GmbH
---------------	----------------------------

Parameter Name	TcpIpIkeSignatureVerifyJobRef	
Description	Reference to Csm_SignatureVerify job which is used to verify signatures for authentication.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.62. TcpIpIkeFragmentationCfg

Parameters included		
Parameter name	Multiplicity	
TcpIpIkeFragmentationEnabled	1..1	
TcpIpIkeTotalFragmentsMaxRx	1..1	
TcpIpIkeTotalFragmentsMaxTx	1..1	
TcpIpIkeDatagramMaxSizeIPv4	1..1	
TcpIpIkeDatagramMaxSizeIPv6	1..1	

Parameter Name	TcpIpIkeFragmentationEnabled	
Description	Enables (TRUE) or disables (FALSE) IKEv2 message fragmentation.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpIkeTotalFragmentsMaxRx	
Description	Specifies the maximum number of fragments that can be buffered during a fragmented reception, i.e. the highest supported "Total Fragments" field value while receiving.	
Multiplicity	1..1	
Type	INTEGER	



Default value	8
Range	<=255 >=1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIkeTotalFragmentsMaxTx
Description	Specifies the maximum number of fragments that can be buffered during a fragmented transmission, i.e. the highest supported "Total Fragments" field value while transmitting.
Multiplicity	1..1
Type	INTEGER
Default value	8
Range	<=255 >=1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIkeDatagramMaxSizeIpv4
Description	Specifies the maximum size of a transmitted IPv4 datagram containing an IKEv2 message. Therefore, the maximum IKEv2 message size is reduced by the length of the IPv4 and UDP header length. Messages resulting in datagrams larger than this limit are going to be fragmented. Consecutively, datagrams containing IKEv2 fragments will have this size, except for the last one.
Multiplicity	1..1
Type	INTEGER
Default value	576
Range	<=1500 >=68
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIkeDatagramMaxSizeIpv6
Description	Specifies the maximum size of a transmitted IPv6 datagram containing an IKEv2 message. Therefore, the maximum IKEv2 message size is reduced by the



	length of the IPv6 and UDP header length. Messages resulting in datagrams larger than this limit are going to be fragmented. Consecutively, datagrams containing IKEv2 fragments will have this size, except for the last one.
Multiplicity	1..1
Type	INTEGER
Default value	1280
Range	<=1500 >=1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.63. TcplIpSecConnections

Containers included		
Container name	Multiplicity	Description
TcplIpSecRemoteAddrConfig	1..1	This parameter is responsible to indicate the remote address configuration for the IPsec connection
TcplIpSecRemotePhysAddr	0..1	Specifies the range of destination MAC Address for filtering.
TcplIpSecSecurityPolicies	1..255	This container describes the security policies between the local and remote host. Communication can either be BY-PASSED or SECURED. The current implementation only supports one inbound and one outbound policy with TcplIpSecSecurityPolicyMechanism equal to SECURED. Transmissions and receptions which do not match any policy are dropped.
TcplIkeConnectionCfg	0..1	This container contains the configuration of the IKEv2 connection. This container needs to be configured only if any SECURED policy uses a TcplIpSecSecurityAssociationCommonCfgRef with TcplIpSecSecurityAssociationKeyExchangeMethod set to DYNAMIC.

Parameters included	
Parameter name	Multiplicity
TcplIpSecConId	1..1
TcplIpSecDomainType	1..1
TcplIpSecLocalAddrRef	1..1



Parameters included

TcpllpSecRemoteAddrType	1..1
---	------

Parameter Name	TcpllpSecConId
Description	IPsec connection identifier to be called as first parameter of Tcplp_RequestIpSecMode.
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<=255 >=0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpllpSecDomainType
Description	Address family ▶ TCPIP_AF_INET: IPv4 address ▶ TCPIP_AF_INET6: IPv6 address
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"> ▶ TCPIP_UNICAST: Unicast address ▶ TCPIP_MULTICAST: Multicast address
Multiplicity	1..1
Type	ENUMERATION
Default value	TCPIP_UNICAST
Range	TCPIP_MULTICAST TCPIP_UNICAST
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.64. TcpllpSecRemoteAddrConfig

Containers included		
Container name	Multiplicity	Description
TcpllpSecRemoteSingleAddress	1..1	Configuration for a remote host using a single IP address.
TcpllpSecRemoteAddress-Range	1..1	Configuration for a remote host using a an IP address range.

5.8.1.65. TcpllpSecRemoteSingleAddress

Parameters included	
Parameter name	Multiplicity
TcpllpSecRemoteAddr	1..1

Parameter Name	TcpllpSecRemoteAddr
Description	Remote IP Address with following notation dependent on parameter TcpllpSecDomainType: <ul style="list-style-type: none"> ▶ TCPIP_AF_INET: Four octets separated by character "." in decimal format ▶ TCPIP_AF_INET6: 8 block to 16bit separated by character ":" in hexadecimal format



Multiplicity	1..1
Type	STRING
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.66. TcpllpSecRemoteAddressRange

Parameters included	
Parameter name	Multiplicity
TcpllpSecStartRemoteAddr	1..1
TcpllpSecEndRemoteAddr	1..1

Parameter Name	TcpllpSecStartRemoteAddr
Description	Remote IP Address with following notation dependent on parameter TcpllpSec-DomainType: <ul style="list-style-type: none">▶ TCPIP_AF_INET: Four octets separated by character "." in decimal format▶ TCPIP_AF_INET6: 8 block to 16bit separated by character ":" in hexadecimal format
Multiplicity	1..1
Type	STRING
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpllpSecEndRemoteAddr
Description	Remote IP Address with following notation dependent on parameter TcpllpSec-DomainType: <ul style="list-style-type: none">▶ TCPIP_AF_INET: Four octets separated by character "." in decimal format▶ TCPIP_AF_INET6: 8 block to 16bit separated by character ":" in hexadecimal format
Multiplicity	1..1
Type	STRING
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH



5.8.1.67. TcpllpSecRemotePhysAddr

Parameters included	
Parameter name	Multiplicity
TcpllpSecStartRemotePhysAddr	1..1
TcpllpSecEndRemotePhysAddr	1..1

Parameter Name	TcpllpSecStartRemotePhysAddr	
Description	Start range of destination MAC Address	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpllpSecEndRemotePhysAddr	
Description	End range of destination MAC Address	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.68. TcpllpSecSecurityPolicies

Containers included		
Container name	Multiplicity	Description
TcpllpSecSecurityRule	2..255	This container describes the ports and protocols for which the policy applies.

Parameters included	
Parameter name	Multiplicity
TcpllpSecSecurityPolicyMechanism	1..1
TcpllpSecSecurityPolicyDirection	1..1
TcpllpSecSecurityAssociationCommonCfgRef	1..1

Parameter Name	TcpllpSecSecurityPolicyMechanism
----------------	---



Description	Indicates if packets shall be bypassed or secured <ul style="list-style-type: none"> ▶ BYPASSED: Packets that match this policy bypass the security protection. ▶ SECURED: Packets that match this policy are checked and protected by IPsec 	
Multiplicity	1..1	
Type	ENUMERATION	
Range	SECURED BYPASSED	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIplpSecSecurityPolicyDirection	
Description	Security policy direction. <ul style="list-style-type: none"> ▶ INBOUND: Describes the handling for incoming messages. ▶ OUTBOUND: Describes the handling of outgoing messages. 	
Multiplicity	1..1	
Type	ENUMERATION	
Range	INBOUND OUTBOUND	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIplpSecSecurityAssociationCommonCfgRef	
Description	Reference to the (common) IPsec security association configuration used for this policy. The IPsec security associations configured on this (common) IPsec security association configuration are shared among all IPsec connections that use the same reference. Note: Only needed when TcpIplpSecSecurityPolicyMechanism is SECURED. Note: Must be the same reference for inbound and outbound SECURED policy.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	PostBuild:	VariantPostBuild



Origin	Elektrobit Automotive GmbH
---------------	----------------------------

5.8.1.69. TcpllpSecSecurityRule

Parameters included	
Parameter name	Multiplicity
TcpllpSecSecurityRuleLocation	1..1
TcpllpSecSecurityRuleUpperLayer	1..1
TcpllpSecSecurityRuleStartPort	1..1
TcpllpSecSecurityRuleEndPort	1..1

Parameter Name	TcpllpSecSecurityRuleLocation	
Description	Location of the rule	
	<ul style="list-style-type: none"> ▶ LOCAL: Port and protocol range refer to local sockets ▶ REMOTE: Port and protocol range refer to remote sockets 	
Multiplicity	1..1	
Type	ENUMERATION	
Range	LOCAL REMOTE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpllpSecSecurityRuleUpperLayer	
Description	Upper layer protocol	
	<ul style="list-style-type: none"> ▶ ICMP: Internet Control Message Protocol for the Internet Protocol Version 4 and 6. ▶ UDP: User Datagram Protocol. ▶ TCP: Transport Control Protocol. ▶ ANY: An arbitrary protocol. 	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	ANY	



Range	UDP
	TCP
	ICMP
	ANY
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIpSecSecurityRuleStartPort
Description	Start of the port range for which the policy is valid
Multiplicity	1..1
Type	INTEGER
Configuration class	PostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIpSecSecurityRuleEndPort
Description	End of the port range for which the policy is valid
Multiplicity	1..1
Type	INTEGER
Configuration class	PostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.70. TcplpIkeConnectionCfg

Containers included		
Container name	Multiplicity	Description
TcplpIkeLocalAuthentication-Method	1..1	This parameter is responsible to indicate the local authentication method in the IKE_AUTH phase
TcplpIkeRemoteAuthenticationMethod	1..1	This parameter is responsible to indicate the local authentication method in the IKE_AUTH phase

Parameters included	
Parameter name	Multiplicity
TcplpIkesInitiator	1..1
TcplpIkeSecurityAssociationCommonCfgRef	1..1



Parameter Name	TcpIpIkeIsInitiator	
Description	Indicates if the local host shall act as initiator of new IKEv2 SAs. This includes starting new IKEv2 SAs automatically during REAUTHENTICATION if no IKEv2 SA is currently established.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpIkeSecurityAssociationCommonCfgRef	
Description	Reference to the (common) IKEv2 security association configuration used for this connection. The IKEv2 security associations configured on this (common) IKEv2 security association configuration are shared among all IKEv2 connections that use the same reference.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.71. TcpIpIkeLocalAuthenticationMethod

Containers included		
Container name	Multiplicity	Description
TcpIpIkeLocalPreshared-KeyAuthenticationMethod	1..1	Specifies the preshared key authentication method for the remote host.
TcpIpIkeLocalCertificateAuthenticationMethod	1..1	Specifies the digital signature authentication method with certificates for the local host.

5.8.1.72. TcpIpIkeLocalPresharedKeyAuthenticationMethod

Parameters included	
Parameter name	Multiplicity



Parameters included

TcplikeLocalPresharedKeyRef	1..1
---	------

Parameter Name	TcplikeLocalPresharedKeyRef
Description	Reference to the local preshared key.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.73. TcplikeLocalCertificateAuthenticationMethod

Parameters included

Parameter name	Multiplicity
TcplikeLocalCertificateRef	1..1
TcplikeSignatureGenerateAlgorithm	1..1
TcplikeSignatureHashAlgorithm	1..1
TcplikeSignatureGenerateJobRef	1..1

Parameter Name	TcplikeLocalCertificateRef
Description	Reference to the certificate of the local host.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplikeSignatureGenerateAlgorithm
Description	The signature algorithm which is used to generate the signature. It has to match the signature algorithm (public key type) in the local certificate.
Multiplicity	1..1
Type	ENUMERATION
Default value	ECDSA256
Range	ECDSA256 ECDSA384



	ECDSA521
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIplkeSignatureHashAlgorithm
Description	The hash algorithm which is used to generate the signature. MUST match the hash algorithm defined in the CSM primitive of TcplikeSignatureGenerateJobRef.
Multiplicity	1..1
Type	ENUMERATION
Default value	SHA256
Range	SHA256 SHA384 SHA512
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIplkeSignatureGenerateJobRef
Description	Reference to the private key. Reference to Csm_SignatureGenerate job which is used to generate signatures for authentication.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.74. TcplikeRemoteAuthenticationMethod

Containers included		
Container name	Multiplicity	Description
TcplikeRemotePreshared-KeyAuthenticationMethod	1..1	Specifies the preshared key authentication method for the remote host.
TcplikeRemoteCertificateAuthenticationMethod	1..1	Specifies the digital signature authentication method with certificates for the remote host.



5.8.1.75. TcplIkeRemotePresharedKeyAuthenticationMethod

Parameters included	
Parameter name	Multiplicity
TcplIkeRemotePresharedKeyRef	1..1

Parameter Name	TcplIkeRemotePresharedKeyRef	
Description	Reference to the remote preshared key.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.76. TcplIkeRemoteCertificateAuthenticationMethod

Parameters included	
Parameter name	Multiplicity
TcplIkeRemotIdentificationRef	1..1
TcplIkelssuerCertificateRef	1..1
TcplIkeVerifyCertificateRef	1..1

Parameter Name	TcplIkeRemotIdentificationRef	
Description	Reference to Identification of the remote host.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplIkelssuerCertificateRef	
Description	Reference to a placeholder for a certificate. This reference is required to verify certificates received in the authentication message. The TcplIkelssuerCertificate is used to verify the signature of the TcplIkeVerifyCertificate.	
Multiplicity	1..1	
Type	REFERENCE	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpIkeVerifyCertificateRef	
Description	Reference to a placeholder for a certificate. This reference is required to verify certificates received in the authentication message. The signature of TcpIpIkeVerifyCertificate is verified with TcpIpIkeIssuerCertificate.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.77. TcpIpIpSecSecurityAssociationCommonCfg

Containers included		
Container name	Multiplicity	Description
TcIpIpSecSecurityProposal	1..n	This container contains a list of IPsec SA proposals used during the IKE_AUTH exchange of the IKEv2 negotiation.

Parameters included	
Parameter name	Multiplicity
TcIpIpSecSecurityAssociationKeyExchangeMethod	1..1
TcIpIpSecSecurityAssociationInboundSpi	1..1
TcIpIpSecSecurityAssociationOutboundSpi	1..1
TcIpIpSecSecurityAssociationProtocol	1..1

Parameter Name	TcIpIpSecSecurityAssociationKeyExchangeMethod
Description	Indicates if the IPsec security association is either configured manually or dynamically through IKEv2. ▶ MANUAL: Manual static configuration. ▶ DYNAMIC: Dynamic configuration through IKEv2.
Multiplicity	1..1
Type	ENUMERATION
Default value	MANUAL



Range	MANUAL
	DYNAMIC
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIpSecSecurityAssociationInboundSpi
Description	Note: This parameter can only be configured if TcpIpIpSecSecurityAssociation-KeyExchangeMethod equals MANUAL. Indicates the Security Policy Index for the inbound IPSec security association.
Multiplicity	1..1
Type	INTEGER
Configuration class	PostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIpSecSecurityAssociationOutboundSpi
Description	Note: This parameter can only be configured if TcpIpIpSecSecurityAssociation-KeyExchangeMethod equals MANUAL. Indicates the Security Policy Index for the outbound IPSec security association.
Multiplicity	1..1
Type	INTEGER
Configuration class	PostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIpSecSecurityAssociationProtocol
Description	<i>The current implementation supports AH only.</i> Defines the used security protocol. ▶ AH: Authentication Header ▶ ESP: Encapsulating Security Payload
Multiplicity	1..1
Type	ENUMERATION
Default value	AH
Range	AH



	ESP
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.78. TcpllpSecSecurityProposal

Containers included		
Container name	Multiplicity	Description
TcpllpSecSecurityAssociationCfg	1..n	<p>This container contains one entry for each IPsec security association that can be used on this (common) IPsec security association configuration.</p> <p>Note: This list of IPsec security associations is shared among all IPsec connections that reference this (common) IPsec security association configuration in the SECURED policy.</p>

Parameters included	
Parameter name	Multiplicity
TcpllpSecSecurityAssociationIntegrityAlgorithm	1..1
TcpllpSecSecurityExtendedSequenceNumberEnabled	1..1
TcpllpSecProposalPriority	1..1

Parameter Name	TcpllpSecSecurityAssociationIntegrityAlgorithm
Description	<p>Supported algorithms for Integrity:</p> <ul style="list-style-type: none"> ▶ AES_CMAC_96 ▶ HMAC_SHA2_256_128 ▶ AES_GMAC_128 ▶ AES_GMAC_256 <p>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.</p>
Multiplicity	1..1
Type	ENUMERATION



Range	AES_CMAC_96 HMAC_SHA2_256_128 AES_GMAC_128 AES_GMAC_256
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIpSecSecurityExtendedSequenceNumberEnabled
Description	Indicates the support for Extended Sequence Number (ESN) handling. Note: This parameter cannot be configured. Extended Sequence Number is always disabled if TcpIpIpSecSecurityAssociationKeyExchangeMethod is equal to MANUAL and it is always enabled if TcpIpIpSecSecurityAssociationKeyExchangeMethod is equal to DYNAMIC.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIpSecProposalPriority
Description	Priority of the IPsec security association proposal. If multiple IPSec proposals are configured the proposal with lower priority value will be preferred during IKEv2 negotiation. If TcpIpIpSecSecurityAssociationKeyExchangeMethod equals MANUAL only one proposal is allowed.
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.79. TcpIpIpSecSecurityAssociationCfg

Parameters included	
Parameter name	Multiplicity



Parameters included

TcplIpSecSecurityAssociationInboundIntegrityJobRef	1..1
TcplIpSecSecurityAssociationOutboundIntegrityJobRef	1..1

Parameter Name	TcplIpSecSecurityAssociationInboundIntegrityJobRef
Description	Reference to a job with key which is used for: Verification of MACs.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplIpSecSecurityAssociationOutboundIntegrityJobRef
Description	Reference to a job with key which is used for: Generation of MACs.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.80. TcplIkeSecurityAssociationCommonCfg

Containers included		
Container name	Multiplicity	Description
TcplIkeSecurityProposal	1..n	This container contains a list of IKEv2 SA proposals used during the IKE_SA_INIT exchange of the IKEv2 negotiation.

Parameters included	
Parameter name	Multiplicity
TcplIkeInitialRetransmissionTimeout	1..1
TcplIkeResponderInitTimeout	1..1
TcplIkeMaxNumRetransmissions	1..1
TcplIkeRetransmissionsIncreaseBase	1..1



Parameters included

TcplIkeSALifeTime	1..1
TcplIkeSAReauthenticationTime	1..1

Parameter Name	TcplIkeInitialRetransmissionTimeout
Description	Time until the first retransmission of an unanswered request IKEv2 message occurs.
Multiplicity	1..1
Type	FLOAT
Default value	1.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplIkeResponderInitTimeout
Description	Maximum time the local host waits for the IKE_AUTH request message of the other side before deleting the IKEv2 SA (only relevant when responder).
Multiplicity	1..1
Type	FLOAT
Default value	1.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplIkeMaxNumRetransmissions
Description	Number of retransmissions before the IKEv2 SA is deleted.
Multiplicity	1..1
Type	INTEGER
Default value	5
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplIkeRetransmissionsIncreaseBase
Description	The retransmission interval increases according to: RetransmissionTime = InitialRetransmissionTimeout*(RetransmissionsIncreaseBase**NumRetransmissions)
Multiplicity	1..1
Type	FLOAT



Default value	2.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIkeSALifeTime
Description	Specifies the time until REAUTHENTICATION starts or DELETION of the IKEv2 SA starts. REAUTHENTICATION starts if either the local host is the initiator or the local host is the responder with only one IKEv2 SA on this connection. Otherwise DELETION starts.
Multiplicity	1..1
Type	FLOAT
Default value	600.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIkeSAReauthenticationTime
Description	Maximal time until REAUTHENTICATION starts or DELETION must finish before the IKEv2 SA is finally deleted.
Multiplicity	1..1
Type	FLOAT
Default value	10.0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.1.81. TcpIpIkeSecurityProposal

Containers included		
Container name	Multiplicity	Description
TcpIpIkeSecurityAssociation-Cfg	1..n	<p>This container contains one entry for each IKEv2 security association that can be used on this (common) IKEv2 security association configuration.</p> <p>Note: This list of IKEv2 security associations is shared among all IPsec connections that reference this (common) IKEv2 security association configuration in TcIpIkeSecurityAssociationCommonCfgRef.</p>



Parameters included

Parameter name	Multiplicity
TcplikeTransformIdAuth	1..1
TcplikeTransformIdEnc	1..1
TcplikeTransformIdPrf	1..1
TcplikePseudoRandomVerifyJobRef	1..1
TcplikePseudoRandomGenerateJobRef	1..1
TcplikeTransformIdDhgroup	1..1
TcplikeProposalPriority	1..1

Parameter Name	TcplikeTransformIdAuth	
Description	Authentication algorithm which shall be used for IKEv2 SA.	
	<ul style="list-style-type: none"> ▶ 12: AUTH_HMAC_SHA2_256_128 ▶ 13: AUTH_HMAC_SHA2_384_192 	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	AUTH_HMAC_SHA2_384_192_NO13	
Range	AUTH_HMAC_SHA2_256_128_NO12 AUTH_HMAC_SHA2_384_192_NO13 NONE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplikeTransformIdEnc
Description	Encryption algorithm which shall be used for IKEv2 SA.
	<ul style="list-style-type: none"> ▶ 12: ENCR_AES_CBC_128 ▶ 12: ENCR_AES_CBC_256 ▶ 19: ENCR_AES_GCM_256_ICV_12 ▶ 20: ENCR_AES_GCM_256_ICV_16 <p>NOTE: When using CBC it has to be ensured that the used Crypto Driver performs removal of padding bytes according to PKCS #7 (https://tools.ietf.org/html/rfc2315#section-10.3)</p>
Multiplicity	1..1



Type	ENUMERATION	
Default value	ENCR_AES_CBC_256_NO12	
Range	ENCR_AES_CBC_128_NO12	
	ENCR_AES_CBC_256_NO12	
	ENCR_AES_GCM_256_ICV_12_NO19	
	ENCR_AES_GCM_256_ICV_16_NO20	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplplkeTransformIdPrf	
Description	Pseudo-random function which shall be used for IKEv2 SA. ▶ 5: PRF_HMAC_SHA2_256 ▶ 6: PRF_HMAC_SHA2_384	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	PRF_HMAC_SHA2_384_NO6	
Range	PRF_HMAC_SHA2_256_NO5	
	PRF_HMAC_SHA2_384_NO6	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplplkePseudoRandomVerifyJobRef	
Description	Reference to Csm_MacVerify job which is used to validate pseudo random values with algorithm from TcplplkeTransformIdPrf.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplplkePseudoRandomGenerateJobRef	
Description	Reference to Csm_MacGenerate job which is used to generate pseudo random values with algorithm from TcplplkeTransformIdPrf.	
Multiplicity	1..1	



Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpIkeTransformIdDhgroup	
Description	Diffie-Hellman group which shall be used for IKEv2 SA. ▶ 19: 256-bit Random ECP Group ▶ 20: 384-bit Random ECP Group	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	DH_384_BIT_RANDOM_ECP_GROUP_NO20	
Range	DH_256_BIT_RANDOM_ECP_GROUP_NO19 DH_384_BIT_RANDOM_ECP_GROUP_NO20	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpIkeProposalPriority	
Description	Priority of the IKEv2 security association proposal. If multiple IKEv2 proposals are configured the proposal with lower priority value will be prefered during IKEv2 negotiation.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.82. TcpIpIkeSecurityAssociationCfg

Parameters included	
Parameter name	Multiplicity
TcpIpIkeDiffieHellmanExchangeKeyRef	1..1
TcpIpIkeAuthenticationGenerateJobRef	1..1
TcpIpIkeAuthenticationVerifyJobRef	1..1



Parameters included

TcplIkeEncryptionJobRef	1..1
TcplIkeDecryptionJobRef	1..1
TcplIkeSKPseudoFunctionVerifyKeyRef	1..1
TcplIkeSKPseudoFunctionGenerateKeyRef	1..1
TcplIkeSKDeriveKeyRef	1..1

Parameter Name	TcplIkeDiffieHellmanExchangeKeyRef	
Description	g^ir - Diffie Hellman key exchange reference for the SKEYSEED calculation	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplIkeAuthenticationGenerateJobRef	
Description	Reference to SK_a key - authentication, a.k.a. integrity protection. (Initiator key or responder key depending on the role) Reference to Csm_MacGenerate job which is used to generate MACs with algorithm from TcplIkeTransformIdAuth.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplIkeAuthenticationVerifyJobRef	
Description	Reference to SK_a key - authentication, a.k.a. integrity protection. (Initiator key or responder key depending on the role) Reference to Csm_MacVerify job which is used to verify integrity with algorithm from TcplIkeTransformIdAuth.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplIkeEncryptionJobRef	



Description	Reference to SK_e key - encryption secret keyref (Initiator key or responder key depending on the role)
	Reference to Csm_Encrypt job which is used to encrypt data which algorithm from TcplIkeTransformIdEnc.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplIkeDecryptionJobRef
Description	Reference to SK_e key - encryption secret keyref (Initiator key or responder key depending on the role)
	Reference to Csm_Decrypt job which is used to decrypt data which algorithm from TcplIkeTransformIdEnc.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplIkeSKPseudoFunctionVerifyKeyRef
Description	Reference to the pseudo random function key of the remote host (SK_pr if local host is initiator, SK_pi if local host is responder)
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplIkeSKPseudoFunctionGenerateKeyRef
Description	Reference to the pseudo random function key of the local host (SK_pi if local host is initiator, SK_pr if local host is responder)
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH



Parameter Name	TcpIplkeSKDeriveKeyRef	
Description	SK_d secret keyref from the SKEYSEED (used for derivation of further keying material for Child SAs)	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.83. TcpIplkeCertificate

Parameters included	
Parameter name	Multiplicity
TcpIplkeCertificateIsCA	1..1
TcpIplkeCertificateKeyRef	1..1
TcpIplkeCertificateSize	1..1
TcpIplpSecPublicKeySha1	1..1
TcpIplkeIssuerCertificateRef	0..1

Parameter Name	TcpIplkeCertificateIsCA	
Description	Indicates if the certificate is a CA.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIplkeCertificateKeyRef	
Description	Reference to a key where the CA certificate is stored.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	
Parameter Name	TcpIplkeCertificateSize	



Description	Max size of the certificate in bytes. Required for transmitting certificates in authentication message to know how much space needs to be reserved for the certificate. The value should be as close as possible or greater to the real certificate size.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIplpSecPublicKeySha1	
Description	SHA1 hash of public key The value is parsed as comma-separated byte values given in hexadecimal representation (uint8 array). E.g. 0x12, 0xab, 0xff ... would be a valid input.	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIplkIssuerCertificateRef	
Description	Reference to the issuer certificate to indicate the certificate path.	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.84. TcpIplkIdentifiers

Containers included		
Container name	Multiplicity	Description
TcplpkIdentifierItems	0..n	This container contains the list of ID_DER ASN1 DN identifier items used to identify the remote host.

Parameters included	
Parameter name	Multiplicity
TcplpkIdentifierCertificateRef	1..1



Parameter Name	TcplpkeldentifierCertificateRef	
Description	Reference to the certificate from which remote identifier items will be extracted from.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.85. TcplpkeldentifierItems

Containers included		
Container name	Multiplicity	Description
TcplpkeldentifierItemId	1..1	Configuration parameter to select if the ID of the present identifier item is a preconfigured or a custom value.
TcplpkeldentifierSrc	1..1	Configuration parameter to select if the content of this identifier item is configured explicitly or extracted from a reference certificate.

5.8.1.86. TcplpkeldentifierItemId

Containers included		
Container name	Multiplicity	Description
TcplpkeldentifierItemId	1..1	This container contains the ID of an identifier item from a preconfigured list.
TcplpkeldentifierItemId	1..1	This container contains the ID of an identifier item with a custom ID.

5.8.1.87. TcplpkeldentifierItemId

Parameters included		
Parameter name	Multiplicity	
TcplpkeldentifierItemId	1..1	
Parameter Name	TcplpkeldentifierItemId	



Description	Object identifier of the identifier item.	
Multiplicity	1..1	
Type	ENUMERATION	
Range	Common_Name	
	Organisation_Name	
	Organizational_Unit_name	
	Locality	
	State_or_province_name	
	Device_Serial_Number	
	Title	
Configuration class	VariantPostBuild:	VariantPostBuild
	Origin	
Elektrobit Automotive GmbH		

5.8.1.88. TcplplkeCustomIdentifierItemId

Parameters included		
Parameter name		Multiplicity
TcplplkeldentifierItemId		1..1
Parameter Name		TcplplkeldentifierItemId
Description		Object identifier of the identifier item.
Multiplicity		1..1
Type		INTEGER
Configuration class	VariantPostBuild:	VariantPostBuild
	Origin	
Elektrobit Automotive GmbH		

5.8.1.89. TcplplkeldentifierSrc

Containers included		
Container name	Multiplicity	Description
TcplplkeConfiguredIdentifier	1..1	This container contains the configuration of the identifier item content.



Containers included

TcplIkeExtractedIdentifier	1..1	This container contains values needed for extracting this identifier item from the reference certificate given in TcplIkeIdentifierCertificateRef.
--	------	--

5.8.1.90. TcplIkeConfiguredIdentifier

Parameters included

Parameter name	Multiplicity
TcplIkeIdentifierItemFormat	1..1
TcplIkeIdentifierItemContent	1..1

Parameter Name [TcplIkeIdentifierItemFormat](#)

Description Format of the identifier item.

Multiplicity 1..1

Type ENUMERATION

Range UTF8_String

Printable_String

Configuration class VariantPostBuild: VariantPostBuild

Origin Elektrobit Automotive GmbH

Parameter Name [TcplIkeIdentifierItemContent](#)

Description Content of the identifier item.

Multiplicity 1..1

Type STRING

Configuration class VariantPostBuild: VariantPostBuild

Origin Elektrobit Automotive GmbH

5.8.1.91. TcplIkeExtractedIdentifier

Parameters included

Parameter name	Multiplicity
TcplIkeIdentifierItemMaxLength	1..1



Parameter Name	TcpIpLkIdentifierItemMaxLength	
Description	Max size of the extracted identifier item.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.92. TcpIpIpSecReportErrorHandler

Parameters included	
Parameter name	Multiplicity
TcplpIpSecReportErrorHandlerFileName	1..1
TcplpIpSecReportErrorHandlerName	1..1

Parameter Name	TcplpIpSecReportErrorHandlerFileName
Description	This parameter specifies the name of the header file containing the declaration of the report error handler function: extern FUNC(void, TCPIP_CODE) TcplpIpSecReportErrorHandlerName (uint8 ctrlIdx, P2CONST(Tcplp_SockAddrType, AUTOMATIC, TCPIP_APPL_DATA) localSockAddPtr, P2CONST(Tcplp_SockAddrType, AUTOMATIC, TCPIP_APPL_DATA) remoteSockAddPtr, uint32 numberOfValidPolicies, uint32 numberOfInvalidPolicies, uint8 errorType);
Multiplicity	1..1
Type	STRING
Configuration class	VariantPostBuild:
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplpIpSecReportErrorHandlerName
Description	This parameter defines the name of the report error handler function. 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 - An error in the IKEv2 negotiation occurred.
Multiplicity	1..1
Type	FUNCTION-NAME



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.93. TcplpDefensiveProgramming

Parameters included	
Parameter name	Multiplicity
TcplpDefProgEnabled	1..1
TcplpPrecondAssertEnabled	1..1
TcplpPostcondAssertEnabled	1..1
TcplpStaticAssertEnabled	1..1
TcplpUnreachAssertEnabled	1..1
TcplpInvariantAssertEnabled	1..1

Parameter Name	TcplpDefProgEnabled
Label	Enable Defensive Programming
Description	<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"> 1. Enable development error detection 2. Enable defensive programming 3. Enable assertions as required
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild:
Origin	Elektrobit Automotive GmbH

Parameter Name	TcplpPrecondAssertEnabled
Label	Enable Precondition Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>TcpIpDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>TcpIpDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpPostcondAssertEnabled
Label	Enable Postcondition Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>TcpIpDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>TcpIpDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpStaticAssertEnabled
Label	Enable Static Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>TcpIpDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>TcpIpDefProgEnabled</code>): must be enabled
Multiplicity	1..1



Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpUnreachAssertEnabled	
Label	Enable Unreachable Code Assertions	
Description	<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"> ▶ Enable Development Error Detection (<code>TcpIpDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>TcpIpDefProgEnabled</code>): must be enabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpInvariantAssertEnabled	
Label	Enable Invariant Assertions	
Description	<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"> ▶ Enable Development Error Detection (<code>TcpIpDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>TcpIpDefProgEnabled</code>): must be enabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	Elektrobit Automotive GmbH
--------	----------------------------

5.8.1.94. TcplpGeneral

Containers included		
Container name	Multiplicity	Description
TcplpSecurityEventRefs	1..1	Container for the references to IdsMEvent elements representing the security events that the Tcplp module shall report to the IdsM in case the corresponding security related event occurs (and if TcplpEnableSecurityEventReporting is set to "true").
TcplpV4General	1..1	This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack for IPv4.
TcplpV6General	1..1	This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack for IPv6.

Parameters included	
Parameter name	Multiplicity
TcplpBufferMemory	1..1
TcplpNumMemoryBlocks	1..1
TcplpDevErrorDetect	1..1
TcplpDhcpServerEnabled	1..1
TcplpMainFunctionPeriod	1..1
TcplpResetIpAssignmentApi	1..1
TcplpScalabilityClass	1..1
TcplpTcpEnabled	1..1
TcplpTcpSocketMax	1..1
TcplpTcpV6SocketMax	1..1
TcplpUdpEnabled	1..1
TcplpUdpMaxMulticastRxInstances	1..1
TcplpUdpSocketMax	1..1
TcplpUdpV6SocketMax	1..1
TcplpUdpAllowImplicitBroadcastReception	1..1



Parameters included

TcplpVersionInfoApi	1..1
TcplpCtrlMax	1..1
TcplpRelocatablePbcfgEnable	1..1
TcplpTransmitRetriesMax	1..1
TcplpGetAndResetMeasurementDataApi	1..1
TcplpEnableMainFunctionTx	1..1
TcplpDhcpGetStatusApi	1..1
TcplpIcmpTransmitErrorApi	1..1
TcplpSecurityMode	1..1
TcplpIpSecGetStatusApiEnabled	1..1
TcplpCustomCsmInterfaceHeaderFile	0..1
TcplpTlsInterfaceEnabled	1..1
TcplpEnableSecurityEventReporting	1..1

Parameter Name	TcplpBufferMemory	
Description	<p>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</p> <p>Memory size in bytes reserved for TCP/IP Tx buffers.</p> <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module. 	
Multiplicity	1..1	
Type	INTEGER	
Default value	8320	
Range	<p><=4294967295</p> <p>>=1</p>	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	
Parameter Name	TcplpNumMemoryBlocks	



Description	The functionality related to this parameter is not supported anymore. See parameter <code>TcplpBufferMemory</code> for more information.	
	Defines the number of memory blocks that <code>TcplpBufferMemory</code> is divided by.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcplpDevErrorDetect	
Description	If true then TCP/IP will enable the error-reporting to the Development Error Tracker (DET).	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplpDhcpServerEnabled	
Description	<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.	
Multiplicity	1..1	
Type	BOOLEAN	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcplpMainFunctionPeriod	
Description	Allow to configure the time for the MainFunction (in seconds). This configuration value shall be equal to the value in the ScheduleManager module. Note: 100ms should be dividable by the Mainfunction period without rest, otherwise it would lead to incorrect timeout calculations	
Multiplicity	1..1	



Type	FLOAT
Default value	0.1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpResetIpAssignmentApi
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> Enables/disables the API TcpIp_ResetIpAssignment of a DHCP-client.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpScalabilityClass
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> In order to customize the TcpIp Stack to the specific needs of the user it can be scaled according to the scalability classes.
Multiplicity	1..1
Type	ENUMERATION
Default value	SC1
Range	SC1 SC2 SC3
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpTcpEnabled
Description	Enable or disable the TCP support. ▶ true: Enables protocol TCP. ▶ false: Disables protocol TCP.



	<p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (config): Disabling this parameter reduces the ROM consumption of the module configuration. ▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code. ▶ RAM reduction: Disabling this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpTcpSocketMax
Description	Maximum number of TCP IPv4 sockets.
	<p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	INTEGER
Default value	1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpTcpV6SocketMax
Description	Maximum number of TCP IPv6 sockets.
	<p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	INTEGER
Default value	1
Configuration class	VariantPostBuild: VariantPostBuild



Origin	Elektrobit Automotive GmbH
---------------	----------------------------

Parameter Name	TcpIpUdpEnabled	
Description	Enables or disabled support of UDP (User Datagram Protocol) <ul style="list-style-type: none"> ▶ true: UDP enabled ▶ false: UDP disabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpUdpMaxMulticastRxInstances	
Description	For Udp broadcast or multicast incoming messages may match multiple sockets This configuration parameter specifies the maximum number of sockets elected for reception.	
Multiplicity	1..1	
Type	INTEGER	
Default value	2	
Range	>=1 <=255	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpUdpSocketMax	
Description	Maximum number of UDP IPv4 sockets. Optimization Effect: <ul style="list-style-type: none"> ▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module. 	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	AUTOSAR_ECUC
---------------	--------------

Parameter Name	TcpIpUdpV6SocketMax			
Description	Maximum number of UDP IPv6 sockets.			
Optimization Effect:				
▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module.				
Multiplicity	1..1			
Type	INTEGER			
Default value	1			
Configuration class	VariantPostBuild:	VariantPostBuild		
Origin	Elektrobit Automotive GmbH			

Parameter Name	TcpIpUdpAllowImplicitBroadcastReception			
Description	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.			
Multiplicity				
1..1				
Type	BOOLEAN			
Default value	false			
Configuration class	PreCompile:	VariantPostBuild		
Origin	Elektrobit Automotive GmbH			

Parameter Name	TcpIpVersionInfoApi			
Description	Switches the API service <code>TcpIp_GetVersionInfo()</code> on or off.			
▶ true: <code>TcpIp_GetVersionInfo()</code> is implemented as function.				
▶ false: <code>TcpIp_GetVersionInfo()</code> is implemented as preprocessor macro.				
Optimization Effect:				
▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code.				

Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpCtrlMax	
Description	<p>This parameter defines the maximum index a EthIf controller referenced via parameter TcpIpEthIfCtrlRef is allowed to have.</p> <p>This parameter has impact on the size of the PostBuild RAM.</p> <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module. 	
Multiplicity	1..1	
Type	INTEGER	
Default value	4	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpRelocatablePbcfgEnable	
Description	<p>Enables/disable support for relocatable postbuild configuration.</p> <ul style="list-style-type: none"> ▶ True: Postbuild configuration relocatable in memory. ▶ False: Postbuild configuration not relocatable in memory. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpTransmitRetriesMax	
Description	The maximal number of retries to transmit a packet before a socket connection is reseted.	
Multiplicity	1..1	



Type	INTEGER
Default value	3
Range	>=0 <=255
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpGetAndResetMeasurementDataApi
Description	Enables / Disables the Get and Reset Measurement Data API
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpEnableMainFunctionTx
Description	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. ▶ true: TcpIp_MainFunctionTx is available. ▶ false: TcpIp_MainFunctionTx is disabled.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpDhcpGetStatusApi
Description	Enables/Disables the API TcpIp_DhcpGetStatus () This function is used to get the status of the dhcp address assignment ▶ true: TcpIp_DhcpGetStatus is available. ▶ false: TcpIp_DhcpGetStatus is disabled.
Multiplicity	1..1



Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIcmpTransmitErrorApi
Description	<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"> ▶ true: <code>TcpIp_IcmpTransmitError</code> is available. ▶ false: <code>TcpIp_IcmpTransmitError</code> is disabled.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpSecurityMode
Description	<p>Sets the security mode for TcpIp communication</p> <p>Available choices:</p> <ul style="list-style-type: none"> ▶ NO_SECURITY: No security applied to TcpIp communication. ▶ FIREWALL: Enables the availability to define bypass/discard policies. ▶ STATIC_IPSEC: Enables the availability to define bypass/secured/discard policies and manual security associations.
Multiplicity	1..1
Type	ENUMERATION
Default value	NO_SECURITY
Range	NO_SECURITY FIREWALL STATIC_IPSEC IPSEC_IKE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIpSecGetStatusApiEnabled	
Description	Enables/disables the TcpIp_IpSecGetStatus() API.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpCustomCsmInterfaceHeaderFile	
Description	Allows to modify the interface to the Csm Module. Specify a c header file which contains the definition of custom Csm functions.	
Multiplicity	0..1	
Type	STRING	
Configuration class	PreCompile:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpTlsInterfaceEnabled	
Description	<!-- Warning shall be removed when all changes are implemented and RFM. --> WARNING: The functionality related to this parameter is not supported by the current implementation. Enables/Disables the interface for Transport Layer Security (TLS). ▶ true: TLS interface is enabled. ▶ false: TLS interface is disabled.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	TcpIpEnableSecurityEventReporting	
Description	Enables (TRUE) or disables (FALSE) the reporting of security events to the IdsM.	
Multiplicity	1..1	



Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.8.1.95. TcplpSecurityEventRefs

Parameters included	
Parameter name	Multiplicity
TCPIP_SEV_ARP_IP_ADDR_CONFLICT	0..1
TCPIP_SEV_DROP_INV_PORT_TCP	0..1
TCPIP_SEV_DROP_INV_PORT_UDP	0..1
TCPIP_SEV_DROP_INV_IPV4_ADDR	0..1
TCPIP_SEV_DROP_INV_IPV6_ADDR	0..1

Parameter Name	TCPIP_SEV_ARP_IP_ADDR_CONFLICT	
Description	Reference to the IdsMEvent TCPIP_SEV_ARP_IP_ADDR_CONFLICT: Received local IP address in ARP reply for different MAC.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TCPIP_SEV_DROP_INV_PORT_TCP	
Description	Reference to the IdsMEvent TCPIP_SEV_DROP_INV_PORT_TCP: Dropped TCP packet because of invalid destination TCP-Port.	
Multiplicity	0..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TCPIP_SEV_DROP_INV_PORT_UDP	
Description	Reference to the IdsMEvent TCPIP_SEV_DROP_INV_PORT_UDP: Dropped UDP packet because of invalid destination UDP-Port.	



Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TCPIP_SEV_DROP_INV_IPV4_ADDR
Description	Reference to the IdsMEvent TCPIP_SEV_DROP_INV_IPV4_ADDR: Dropped datagram because of invalid IPV4 address.
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TCPIP_SEV_DROP_INV_IPV6_ADDR
Description	Reference to the IdsMEvent TCPIP_SEV_DROP_INV_IPV6_ADDR: Dropped datagram because of invalid IPV6 address.
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.8.1.96. TcplpIpV4General

Parameters included	
Parameter name	Multiplicity
TcplpArpEnabled	1..1
TcplpAutolpEnabled	1..1
TcplpDhcpClientEnabled	1..1
TcplpDhcpSimpleClientEnabled	1..1
TcplpIcmpEnabled	1..1
TcplpIpV4Enabled	1..1
TcplpLocalAddrIpv4EntriesMax	1..1
TcplpPathMtuDiscoveryEnabled	1..1



Parameter Name	TcpIpArpEnabled	
Description	Enables or disabled support of ARP (Address Resolution Protocol). <ul style="list-style-type: none"> ▶ true: ARP enabled ▶ false: ARP disabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpAutolpEnabled	
Description	Enable or disable the Auto-IP support. <ul style="list-style-type: none"> ▶ true: Enables Auto-IP. ▶ false: Disables Auto-IP. 	
	Optimization Effect:	
	<ul style="list-style-type: none"> ▶ ROM reduction (config): Disabling this parameter reduces the ROM consumption of the module configuration. ▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code. ▶ RAM reduction: Disabling this parameter reduces the RAM consumption of the module. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpDhcpClientEnabled	
Description	Enable or disable the DHCP client support. <ul style="list-style-type: none"> ▶ true: Enables DHCP client. ▶ false: Disables DHCP client. 	
	Optimization Effect:	



	<ul style="list-style-type: none"> ▶ ROM reduction (config): Disabling this parameter reduces the ROM consumption of the module configuration. ▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code. ▶ RAM reduction: Disabling this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	TcpIpDhcpSimpleClientEnabled: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpDhcpSimpleClientEnabled
Description	<p>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.</p> <ul style="list-style-type: none"> ▶ true: Enables DHCP client. ▶ false: Disables DHCP client. <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (config): Disabling this parameter reduces the ROM consumption of the module configuration. ▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	TcpIpLcmEnabled: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpLcmEnabled
Description	Enables or disabled support of ICMP (Internet Control Message Protocol).

- ▶ true: ICMP enabled



	▶ false: ICMP disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpIplpV4Enabled
Description	Enables (TRUE) or disables (FALSE) support of IPv4 (Internet Protocol version 4).
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpLocalAddrIpv4EntriesMax
Description	Maximum number of IPv4 address table entries in PostBuild configuration. This parameter has impact on the size of the PostBuild RAM.
	Optimization Effect:
	<ul style="list-style-type: none"> ▶ ROM reduction (config): Selecting a small value for this parameter reduces the ROM consumption of the module configuration. ▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	INTEGER
Default value	10
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpPathMtuDiscoveryEnabled
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i>



	Enables (TRUE) or disables (FALSE) the discovery of the maximum transmission unit on a path according to IETF RfC 1191.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.8.1.97. TcplpIpV6General

Parameters included	
Parameter name	Multiplicity
TcplpDhcpV6ClientEnabled	1..1
TcplpIpV6ExtensionHeaderFilterEnabled	1..1
TcplpIpV6Enabled	1..1
TcplpIpV6PathMtuDiscoveryEnabled	1..1
Tcplp_LocalAddrIpv6EntriesMax	1..1
TcplpNdpAddressResolutionUnreachabilityDetectionEnabled	1..1
TcplpNdpPrefixAndRouterDiscoveryEnabled	1..1
TcplpSlaacLinkLocalEnabled	1..1
TcplpSlaacLinkLocalEntriesMax	1..1
TcplpIpV6TrafficClassFilterEnabled	1..1
TcplpIpV6FlowLabelFilterEnabled	1..1

Parameter Name	TcplpDhcpV6ClientEnabled
Description	Enables (TRUE) or disables (FALSE) the DHCPv6 (Dynamic Host Configuration Protocol for IPv6) Client.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcplpIpV6ExtensionHeaderFilterEnabled
-----------------------	--



Description	Enables (TRUE) or disables (FALSE) filtering of IPv6 extension headers
	<ul style="list-style-type: none"> ▶ ROM reduction (config): Disabling this parameter reduces the ROM consumption of the module configuration. ▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code. ▶ RAM reduction: Disabling this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIpv6Enabled
Description	Enables (TRUE) or disables (FALSE) support of IPv6 (Internet Protocol version 6).
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpIpv6PathMtuDiscoveryEnabled
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i>
	Enables (TRUE) or disables (FALSE) Path MTU Discovery support for IPv6 according to IETF RFC 1981.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	TcpIpLocalAddrIpv6EntriesMax
-----------------------	-------------------------------------



Description	Maximum number of LocalAddr table entries for IPv6	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpNdpAddressResolutionUnreachabilityDetectionEnabled	
Description	<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>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpNdpPrefixAndRouterDiscoveryEnabled	
Description	Enables (TRUE) or disables (FALSE) support of Prefix and Router Discovery via NDP.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	TcpIpSlaacLinkLocalEnabled	
Description	<p>Enable or disable the Stateless Address Auto Configuration of IPv6 Link Local Addresses.</p> <ul style="list-style-type: none"> ▶ true: Enables Local Slaac. ▶ false: Disables Local Slaac. 	



	<p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ ROM reduction (config): Disabling this parameter reduces the ROM consumption of the module configuration. ▶ ROM reduction (code): Disabling this parameter reduces the ROM consumption of the module code. ▶ RAM reduction: Disabling this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpSlaacLinkLocalEntriesMax
Description	<p>This parameter defines the maximum number of interface which are allowed to have an entry in TcpIpLocalAddr assigned with TcpIpAssignmentMethod set to TCPIP_LINKLOCAL (Ipv6).</p> <p>This parameter has impact on the size of the PostBuild RAM.</p> <p>Optimization Effect:</p> <ul style="list-style-type: none"> ▶ RAM reduction: Selecting a small value for this parameter reduces the RAM consumption of the module.
Multiplicity	1..1
Type	INTEGER
Default value	1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpIpV6TrafficClassFilterEnabled
Description	<p>Enables Traffic Class filter for received IPv6 packets. When enabled TcpIp will discard received packets with Traffic Class field value different than zero.</p> <ul style="list-style-type: none"> ▶ true: value filtering enabled ▶ false: value filtering disabled
Multiplicity	1..1



Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	TcpIpV6FlowLabelFilterEnabled
Description	Enables Flow Label filter for received IPv6 packets. When enabled TcpIp will discard received packets with Flow Label field value different than zero. <ul style="list-style-type: none"> ▶ true: value filtering enabled ▶ false: value filtering disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.8.2. Recommended configurations

5.8.2.1. TcpIpRecConfiguration

Containers included	
Container name	Container definition
TcIpConfig	TcIpConfig

Parameters included

Parameter name	Value
----------------	-------

5.8.2.1.1. TcIpConfig

Containers included	
Container name	Container definition

**Containers included**

TcplpSocketOwnerConfig	TcplpConfig TcplpSocketOwnerConfig
--	--

TcplpIpConfig	TcplpConfig TcplpIpConfig
-------------------------------	---

Parameters included

Parameter name	Value
----------------	-------

5.8.2.1.2. TcplpSocketOwnerConfig**Containers included**

Container name	Container definition
----------------	----------------------

TcplpSocketOwner	TcplpSocketOwner
----------------------------------	----------------------------------

Parameters included

Parameter name	Value
----------------	-------

5.8.2.1.3. TcplpSocketOwner**Parameters included**

Parameter name	Value
TcplpSocketOwnerHeaderFileName	(DISABLED)
TcplpSocketOwnerCopyTxDataName	(DISABLED)
TcplpSocketOwnerLocallpAddrAssignmentChg- Name	(DISABLED)
TcplpSocketOwnerRxIndicationName	(DISABLED)
TcplpSocketOwnerTcpAcceptedName	(DISABLED)
TcplpSocketOwnerTcpConnectedName	(DISABLED)
TcplpSocketOwnerTxConfirmationName	(DISABLED)
TcplpSocketOwnerUpperLayerType	SOAD

5.8.2.1.4. TcplpIpConfig**Containers included**

Container name	Container definition
----------------	----------------------



Containers included

TcpllpV4Config	TcpllpConfig TcpllpV4Config
--------------------------------	---

Parameters included

Parameter name	Value
----------------	-------

5.8.2.1.5. TcpllpV4Config

Containers included

Container name	Container definition
TcplpArpConfig	TcplpArpConfig

Parameters included

Parameter name	Value
----------------	-------

5.8.2.1.6. TcplpArpConfig

Parameters included

Parameter name	Value
TcplpArpPacketQueueEnabled	false
TcplpArpTableSizeMax	5
TcplpArpNumGratuitousARPonStartup	0
TcplpArpTableEntryTimeout	16.0

5.8.3. Application programming interface (API)

5.8.3.1. Type definitions

5.8.3.1.1. Tcpip_ArpCacheEntryType

Purpose	Specifies an entry in the ARP cache.
---------	--------------------------------------



Type	struct	
Members	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

Purpose	Socket state machine events signalized to upper layer.
Type	uint8

5.8.3.1.3. Tcplp_IPsecStateType

Purpose	Specifies the IPsec state for a specific IPsec connection.
Type	uint8

5.8.3.1.4. Tcplp_IpAddrAssignmentType

Purpose	Specification of IP address assignment policy.
Type	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_IpSecChildSaStatusType

Purpose		
Type	struct	
Members	uint32 numOfPacketsReceived	Packets in - last 32 bits of the incoming packet counter.



	<code>uint32 numOfPacketsTransmitted</code>	Packets out - last 32 bits of the incoming packet counter.
	<code>uint16 lastPacketReceivedTime</code>	Last packet in (s) - seconds since last packet received on SA. If value > 0xffff, set to 0xffff.
	<code>uint16 lastPacketTransmitted-Time</code>	Last packet out (s) - seconds since last packet sent on SA. If value > 0xffff, set to 0xffff.
	<code>uint8 inboundSpi</code>	Child SA Inbound SPI last two bytes.
	<code>uint8 outboundSpi</code>	Child SA Outbound SPI last two bytes.
	<code>uint8 protocol</code>	Protocol of encryption (2: AH, 3: ESP).

5.8.3.1.7. `TcpIp_IpSecIkeSaStatusType`

Purpose		
Type	<code>struct</code>	
Members	<code>TcpIp_IpSecChildSaStatusType * childSaStatusPtr</code>	Pointer to memory where the list of Child SAs shall be stored. Memory should be at least of size <code>numberOfChildSaElements</code> .
	<code>uint8 * numberOfChildSaElements</code>	In: Maximum number of entries that can be stored in output <code>childSaStatusPtr</code> . Out: Number of entries written to output <code>childSaStatusPtr</code> (Number of all entries in the cache if input value is 0).
	<code>uint8 initiatorSpi</code>	IKEv2 Initiator SPI last two bytes.
	<code>uint8 responderSpi</code>	IKEv2 Responder SPI last two bytes.
	<code>uint8 ikeState</code>	status of the ike security association
	<code>boolean isIkeInitiator</code>	Flag determining if the local peer is the IKE initiator (TRUE) or the responder (FALSE) of the IKE SA.

5.8.3.1.8. `TcpIp_Ip_HandleIdType`

Purpose	Handle identifier.
Type	<code>uint16</code>
Description	Type definition of the handle identifier used in the <code>CopyData</code> functor interface.



5.8.3.1.9. `Tcplp_Ip_RxReturnType`

Purpose	Return type used for processing of received datagrams inside <code>Tcplp_RxIndication</code> .
Type	<code>uint8</code>

5.8.3.1.10. `Tcplp_NdpCacheEntryType`

Purpose	Specifies an entry in the NDP cache.		
Type	<code>struct</code>		
Members	<code>uint32 Inet6Addr</code>	<code>IPv6 address in network byte order.</code>	
	<code>uint8 PhysAddr</code>	<code>physical address in network byte order</code>	
	<code>uint8 State</code>	<code>state of the address entry</code>	

5.8.3.1.11. `Tcplp_ParamIdType`

Purpose	Type for the specification of all supported Parameter IDs.
Type	<code>uint8</code>

5.8.3.1.12. `Tcplp_ProtocolType`

Purpose	Protocol.
Type	<code>uint8</code>
Description	Type definition of the used protocol

5.8.3.1.13. `Tcplp_ReturnType`

Purpose	Tcplp specific return type.
Type	<code>uint8</code>

5.8.3.1.14. `Tcplp_SockAddrPtrType`

Purpose	Pointer to structure of type <code>Tcplp_SockAddrType</code> .
----------------	--



Type	Tcplp_SockAddrType *
------	--------------------------------------

5.8.3.1.15. 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.	

5.8.3.1.16. Tcplp_SocketIdType

Purpose	Socket identifier.
Type	TcpIp_Ip_HandleIdType
Description	Type definition of socket identifier

5.8.3.1.17. Tcplp_StateType

Purpose	Specifies the Tcplp state for a specific Ethlf 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 Tcplp_DhcpReadOption() and Tcplp_DhcpWriteOption() .
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 Tcplp_DhcpReadOption() and Tcplp_DhcpWriteOption() .
Value	81U
Description	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_DTLS_STATE_CLOSED

Purpose	Dtls is in CLOSED state on a Udp socket.
Value	0U

5.8.3.2.7. TCPIP_DTLS_STATE_CONNECTED

Purpose	Dtls is in CONNECTED state on a Udp socket.
----------------	---



Value	2U
--------------	----

5.8.3.2.8. TCPIP_DTLS_STATE_CONNECTING

Purpose	Dtls is in CONNECTING state on a Udp socket.
Value	1U

5.8.3.2.9. TCPIP_E_ADDRINUSE

Purpose	Det error Id TCPIP_E_ADDRINUSE.
Value	0x09U

5.8.3.2.10. TCPIP_E_ADDRNOTASSIGNED

Purpose	operation failed, no local address assigned
Value	6U

5.8.3.2.11. TCPIP_E_ADDRNOTAVAIL

Purpose	Det error Id TCPIP_E_ADDRNOTAVAIL.
Value	0x0aU

5.8.3.2.12. TCPIP_E_AFNOSUPPORT

Purpose	Det error Id TCPIP_E_AFNOSUPPORT.
Value	0x0eU

5.8.3.2.13. TCPIP_E_DESTADDRREQ

Purpose	Det error Id TCPIP_E_DESTADDRREQ.
----------------	-----------------------------------



Value	0x06U
--------------	-------

5.8.3.2.14. TCPIP_E_DROP

Purpose	operation failed, ignore the packet
Value	3U

5.8.3.2.15. TCPIP_E_INV_ARG

Purpose	Det error Id TCPIP_E_INV_ARG.
Value	0x03U

5.8.3.2.16. TCPIP_E_INV_SOCKADDR

Purpose	Det error Id TCPIP_E_INV_SOCKADDR.
Value	0x05U

5.8.3.2.17. TCPIP_E_ISCONN

Purpose	Det error Id TCPIP_E_ISCONN.
Value	0x0bU

5.8.3.2.18. TCPIP_E_MEMORY

Purpose	operation failed, out of memory
Value	5U

5.8.3.2.19. TCPIP_E_MSGSIZE

Purpose	Det error Id TCPIP_E_MSGSIZE.
----------------	-------------------------------



Value	0x07U
--------------	-------

5.8.3.2.20. TCPIP_E_NOBUFS

Purpose	Det error Id TCPIP_E_NOBUFS.
Value	0x04U

5.8.3.2.21. TCPIP_E_NOPROTOOPT

Purpose	Det error Id TCPIP_E_NOPROTOOPT.
Value	0x0dU

5.8.3.2.22. TCPIP_E_NOTCONN

Purpose	Det error Id TCPIP_E_NOTCONN.
Value	0x0cU

5.8.3.2.23. TCPIP_E_NOTINIT

Purpose	Det error Id TCPIP_E_NOTINIT.
Value	0x01U

5.8.3.2.24. TCPIP_E_NOT_OK

Purpose	operation failed
Value	1U

5.8.3.2.25. TCPIP_E_NOT_PERMITTED

Purpose	operation failed, illegal request
Value	4U



5.8.3.2.26. TCPIP_E_PARAM_POINTER

Purpose	Det error Id TCPIP_E_PARAM_POINTER.
Value	0x02U

5.8.3.2.27. TCPIP_E_PENDING

Purpose	operation failed temporarily, dependent operation not yet concluded
Value	7U

5.8.3.2.28. TCPIP_E_PHYS_ADDR_MISS

Purpose	operation failed because of an ARP cache miss
Value	2U

5.8.3.2.29. TCPIP_E_PROTOTYPE

Purpose	Det error Id TCPIP_E_PROTOTYPE.
Value	0x08U

5.8.3.2.30. TCPIP_IPADDR_ASSIGNMENT_ALL

Purpose	all configured methods with manual trigger
Value	5U

5.8.3.2.31. TCPIP_IPADDR_ASSIGNMENT_DHCP

Purpose	address obtained via DHCP
Value	3U

5.8.3.2.32. TCPIP_IPADDR_ASSIGNMENT_IPV6_ROUTER

Purpose	address obtained via IPV6 ROUTER
----------------	----------------------------------



Value	4U
--------------	----

5.8.3.2.33. TCPIP_IPADDR_ASSIGNMENT_LINKLOCAL

Purpose	address obtained via LINK-LOCAL
Value	1U

5.8.3.2.34. TCPIP_IPADDR_ASSIGNMENT_LINKLOCAL_DOIP

Purpose	address obtained via LINK-LOCAL or DO-IP
Value	2U

5.8.3.2.35. TCPIP_IPADDR_ASSIGNMENT_STATIC

Purpose	Static IPv4 address assignment.
Value	0U

5.8.3.2.36. TCPIP_IPADDR_STATE_ASSIGNED

Purpose	IP address assignment in use.
Value	0U

5.8.3.2.37. TCPIP_IPADDR_STATE_ONHOLD

Purpose	IP address assignment in use, but link lost.
Value	1U

5.8.3.2.38. TCPIP_IPADDR_STATE_UNASSIGNED

Purpose	IP address assignment unused.
Value	2U



5.8.3.2.39. TCPIP IPPROTO_TCP

Purpose	Protocol TCP.
Value	0x06U

5.8.3.2.40. TCPIP IPPROTO_UDP

Purpose	Protocol UDP.
Value	0x11U

5.8.3.2.41. TCPIP_IPSEC_ACTIVE

Purpose	IPsec is activated for this connection.
Value	0U

5.8.3.2.42. TCPIP_IPSEC_INACTIVE

Purpose	IPsec is deactivated for this connection.
Value	1U

5.8.3.2.43. TCPIP_IPSEC_INVALID_ICV

Purpose	
Value	2U

5.8.3.2.44. TCPIP_IPSEC_SECURE

Purpose	IPsec is activated, IKE remains inactive for this connection.
Value	2U

5.8.3.2.45. TCPIP_IP_RX_DEST_UNREACHABLE

Purpose	Destination address not known to host.
----------------	--



Value	0x10U
--------------	-------

5.8.3.2.46. TCPIP_IP_RX_EXT_HDR_OK

Purpose	extension header successfully processed
Value	0x05U

5.8.3.2.47. TCPIP_IP_RX_FRAGMENT_HEADER_SHORT

Purpose	Fragment header too short.
Value	0x08U

5.8.3.2.48. TCPIP_IP_RX_FRAGMENT_LENGTH

Purpose	Length problem during fragment reassembly.
Value	0x0AU

5.8.3.2.49. TCPIP_IP_RX_FRAGMENT_OFFSET

Purpose	Offset problem during fragment reassembly.
Value	0x0BU

5.8.3.2.50. TCPIP_IP_RX_FRAGMENT_OUT_OF_BUFFERS

Purpose	data was not delivered to transport protocol - no more buffers to handle
Value	0x07U

5.8.3.2.51. TCPIP_IP_RX_FRAGMENT_OVERLAP

Purpose	Overlap problem during fragment reassembly.
Value	0x0CU



5.8.3.2.52. TCPIP_IP_RX_FRAGMENT_RESERVED

Purpose	Reserved flag usage problem during fragment reassembly.
Value	0x0DU

5.8.3.2.53. TCPIP_IP_RX_FRAGMENT_TIMEOUT

Purpose	Timeout during fragment reassembly.
Value	0x09U

5.8.3.2.54. TCPIP_IP_RX_INVALID_HEADER

Purpose	Basic header check failed.
Value	0x0FU

5.8.3.2.55. TCPIP_IP_RX_NOT_OK

Purpose	data was not accepted
Value	0x01U

5.8.3.2.56. TCPIP_IP_RX_OK

Purpose	data was accepted
Value	0x00U

5.8.3.2.57. TCPIP_IP_RX_PORT_UNREACHABLE

Purpose	data was not accepted by transport protocol - port unknown
Value	0x03U

5.8.3.2.58. TCPIP_IP_RX_PROTOCOL_UNREACHABLE

Purpose	data was not delivered to transport protocol - protocol unknown
---------	---



Value	0x02U
--------------	-------

5.8.3.2.59. TCPIP_IP_RX_PROTOCOL_UNREACHABLE_FRAG

Purpose	data was not delivered to transport protocol - protocol unknown
Value	0x04U

5.8.3.2.60. TCPIP_IP_RX_RES_POLICYCHECK

Purpose	Reserved values for policy check handlers start here.
Value	0x60U

5.8.3.2.61. TCPIP_IP_RX_SRCADDR_INVALID

Purpose	Data not accepted because of invalid source address.
Value	0x11U

5.8.3.2.62. TCPIP_IP_RX_UNRECOGNIZED_OPTION

Purpose	data was not accepted because of unrecognized ipv6 option code
Value	0x06U

5.8.3.2.63. TCPIP_IP_RX_UNRECOGNIZED_ROUTING_TYPE

Purpose	Reserved flag Segments Left is non-zero with unrecognized Routing Type.
Value	0x0EU

5.8.3.2.64. TCPIP_LOCALADDRID_ANY

Purpose	Any local IP address.
Value	254U



Description	This macro defines a value to select any local IP address (instead of a specific one) used for API service Tcplp_Bind() .
--------------------	---

5.8.3.2.65. TCPIP_OK

Purpose	operation completed successfully
Value	0U

5.8.3.2.66. TCPIP_PARAMID_DSCP

Purpose	The 6-bit Differentiated Service Code Point according to IETF RFC 2474.
Value	0x0bU

5.8.3.2.67. TCPIP_PARAMID_FLOWLABEL

Purpose	The 20-bit Flow Label according to IETF RFC 6437.
Value	0x0aU

5.8.3.2.68. TCPIP_PARAMID_FRAMEPRIO

Purpose	Specifies the frame priority for outgoing frames on the socket.
Value	0x01U

5.8.3.2.69. TCPIP_PARAMID_PATHMTU_ENABLE

Purpose	Specifies if the Path MTU Discovery shall be performed on the related socket.
Value	0x09U

5.8.3.2.70. TCPIP_PARAMID_TCP_KEEPALIVE

Purpose	Specifies if TCP Keep Alive Probes are sent on the socket connection.
Value	0x03U



5.8.3.2.71. TCPIP_PARAMID_TCP_KEEPALIVE_INTERVAL

Purpose	Specifies the interval between subsequent keepalive probes.
Value	0x07U

5.8.3.2.72. TCPIP_PARAMID_TCP_KEEPALIVE_PROBES_MAX

Purpose	Specifies the maximum number of times that a keepalive probe is retransmitted.
Value	0x06U

5.8.3.2.73. TCPIP_PARAMID_TCP_KEEPALIVE_TIME

Purpose	Specifies the time between the last data packet sent and the first keepalive probe.
Value	0x05U

5.8.3.2.74. TCPIP_PARAMID_TCP_NAGLE

Purpose	Specifies if the Nagle Algorithm according to IETF RFC 896 is enabled or not.
Value	0x02U

5.8.3.2.75. TCPIP_PARAMID_TCP_OPTIONFILTER

Purpose	Specifies which TCP option filter shall be applied on the related socket.
Value	0x08U

5.8.3.2.76. TCPIP_PARAMID_TCP_RXWND_MAX

Purpose	Specifies the maximum TCP receive window for the socket.
Value	0x00U

5.8.3.2.77. TCPIP_PARAMID_TCP_TLS_RXWND_MAX

Purpose	Specifies the maximum TCP receive window for the socket with active TLS.
----------------	--



Value	0x82U
--------------	-------

5.8.3.2.78. TCPIP_PARAMID_TLS_CONNECTION_ASSIGNMENT

Purpose	Specifies if TLS interface shall be enabled (TRUE) or disabled (FALSE) on the related socket.
Value	0x0dU

5.8.3.2.79. TCPIP_PARAMID_TTL

Purpose	Specifies the time to live value for outgoing frames on the socket.
Value	0x04U

5.8.3.2.80. TCPIP_PARAMID_UDP_CHECKSUM

Purpose	Specifies if UDP checksum handling shall be enabled (TRUE) or skipped (FALSE) on the related socket.
Value	0x0cU

5.8.3.2.81. TCPIP_PARAMID_UDP_UNSPECIFIED_IP

Purpose	Specifies if unspecified ip address shall be used for transmission as source.
Value	0x81U

5.8.3.2.82. TCPIP_PHYS_ADDR_ENTRY_STATE_DELAY

Purpose	
Value	0x10U

5.8.3.2.83. TCPIP_PHYS_ADDR_ENTRY_STATE_FREE

Purpose	
----------------	--



Value	0x40U
--------------	-------

5.8.3.2.84. TCPIP_PHYS_ADDR_ENTRY_STATE_INCOMPLETE

Purpose	
Value	0x02U

5.8.3.2.85. TCPIP_PHYS_ADDR_ENTRY_STATE_PROBE

Purpose	
Value	0x20U

5.8.3.2.86. TCPIP_PHYS_ADDR_ENTRY_STATE_REACHABLE

Purpose	
Value	0x04U

5.8.3.2.87. TCPIP_PHYS_ADDR_ENTRY_STATE_STALE

Purpose	
Value	0x08U

5.8.3.2.88. TCPIP_PHYS_ADDR_ENTRY_STATE_STATIC

Purpose	
Value	0x01U

5.8.3.2.89. 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 Tcplp_SockAddrType .



5.8.3.2.90. TCPIP_SOCKETID_INVALID

Purpose	Represents the invalid value for function arguments of type Tcplp_SocketIdType.
Value	0xFFFFFU

5.8.3.2.91. TCPIP_STATE_OFFLINE

Purpose	TCP/IP stack state for a specific EthIf controller is OFFLINE, i.e. no communication is possible.
Value	2U

5.8.3.2.92. 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

5.8.3.2.93. 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.
Value	0U

5.8.3.2.94. TCPIP_STATE_SHUTDOWN

Purpose	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.
Value	4U

5.8.3.2.95. TCPIP_STATE_STARTUP

Purpose	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.
----------------	---



Value	3U
--------------	----

5.8.3.2.96. TCPIP_TCP_CLOSED

Purpose	TCP connection reached state 'CLOSE', all resources released.
Value	0x01U

5.8.3.2.97. TCPIP_TCP_FIN_RECEIVED

Purpose	TCP segment with set FIN flag received.
Value	0x02U

5.8.3.2.98. TCPIP_TCP_RESET

Purpose	TCP connection reseted, all resources released.
Value	0x00U

5.8.3.2.99. TCPIP_TLS_ALERT

Purpose	Alert has occurred and Tls connection was closed.
Value	0x06U

5.8.3.2.100. TCPIP_TLS_CLOSE_NOTIFY_RECEIVED

Purpose	Close notify was received.
Value	0x05U

5.8.3.2.101. TCPIP_TLS_HANDSHAKE_SUCCEEDED

Purpose	TLS handshake successfully established, TLS connection available.
----------------	---



Value	0x04U
--------------	-------

5.8.3.2.102. TCPIP_TLS_WARNING

Purpose	Warning has occurred and Tls connection continued.
Value	0x07U

5.8.3.2.103. TCPIP_UDP_CLOSED

Purpose	UDP socket and all related resources have been released.
Value	0x03U

5.8.3.3. Functions

5.8.3.3.1. Tcplp_Bind

Purpose	Binds a TCP/UDP socket to a local address/port pair.	
Synopsis	<pre>Std_ReturnType TcpIp_Bind (TcpIp_SocketIdType SocketId , TcpIp_LocalAddrIdType LocalAddrId , uint16 * PortPtr);</pre>	
Service ID	0x05	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SocketId	
Parameters (in)	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 Tcplp 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) IpAd-



		dr entry has to be configured in the Tcplp module configuration. The IP address given by parameter localAddrPtr has not effect then.
Parameters (in,out)	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.
Return Value	Result of operation	
	E_OK	The request has been accepted
	E_NOT_OK	The request has not been accepted (e.g. address in use)
Description	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 IpAddrId) 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 Tcplp_TcpConnect() or Tcplp_Transmit() .	

5.8.3.3.2. Tcplp_ChangeParameter

Purpose	Changes socket configuration.	
Synopsis	<code>Std_ReturnType TcpIp_ChangeParameter (TcpIp_SocketIdType SocketId , TcpIp_ParamIdType ParameterId , const uint8 * ParameterValue);</code>	
Service ID	0x0f	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SocketId	
Parameters (in)	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.
Return Value	Result of operation	



	E_OK	The request has been accepted
	E_NOT_OK	The request has not been accepted
Description	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

Purpose	Closes a socket.	
Synopsis	Std_ReturnType TcpIp_Close (TcpIp_SocketIdType SocketId , boolean Abort);	
Service ID	0x04	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different SocketId	
Parameters (in)	SocketId	identifying the local socket resource.
	Abort	Terminate immediately?
Return Value	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).
Description	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

Purpose	Read DHCP status.
Synopsis	Std_ReturnType TcpIp_DhcpGetStatus (TcpIp_LocalAddrIdType LocalAddressId , uint8 * StatusPtr);



Parameters (in)	LocalAddressId	Address Id of the DHCP address assignment
	StatusPtr	Pointer to a status variable. Contains the state information.
Return Value	Std_ReturnType	
	E_OK	Status was successfully retrieved
	E_NOT_OK	No Dhcp address assignment was found
Description	<p>This functions 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 lease extension from any server 8 / REASSIGNING (DhcpV6 only) Requested specific lease in response to NoBinding status code 9</p>	

5.8.3.3.5. Tcplp_DhcpReadOption

Purpose	Retrieves value of a DHCP option.	
Synopsis	<pre>Std_ReturnType TcpIp_DhcpReadOption (TcpIp_LocalAddrIdType LocalIpAddrId , uint8 Option , uint8 * DataLength , uint8 * DataPtr);</pre>	
Service ID	0x0D	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	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
Parameters (in,out)	DataLength	in: contains the length of the provided data buffer. out: length of the actual data.
Return Value	Result of operation	



	E_OK	requested data retrieved successfully.
	E_NOT_OK	requested data could not be retrieved
Description	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

Purpose	Retrieves value of a DHCPv6 option.	
Synopsis	<pre>Std_ReturnType TcpIp_DhcpV6ReadOption (TcpIp_LocalAddrIdType LocalIpAddrId , uint16 Option , uint16 * DataLength , uint8 * DataPtr);</pre>	
Service ID	0x19	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	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
Parameters (in,out)	DataLength	in: contains the length of the provided data buffer. out: length of the actual data.
Return Value	Result of operation	
	E_OK	requested data retrieved successfully.
	E_NOT_OK	requested data could not be retrieved
Description	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

Purpose	Write value of a DHCPv6 option.
Synopsis	<pre>Std_ReturnType TcpIp_DhcpV6WriteOption (TcpIp_LocalAddrIdType LocalIpAddrId , uint16 Option , uint16 DataLength , const uint8 * DataPtr);</pre>



Service ID	0x1a	
Sync/Async	Synchronous	
Reentrancy	None Reentrant	
Parameters (in)	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
Return Value	Result of operation	
	E_OK	no error occurred.
	E_NOT_OK	DHCP option data could not be written.
Description	By this API service the TCP/IP stack writes the DHCPv6 option data identified by parameter option.	

5.8.3.3.8. TcpIp_DhcpWriteOption

Purpose	Write value of a DHCP option.	
Synopsis	<pre>Std_ReturnType TcpIp_DhcpWriteOption (TcpIp_LocalAddrIdType LocalIpAddrId , uint8 Option , uint8 DataLength , const uint8 * DataPtr);</pre>	
Service ID	0x0E	
Sync/Async	Synchronous	
Reentrancy	None Reentrant	
Parameters (in)	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
Return Value	Result of operation	
	E_OK	no error occurred.



	E_NOT_OK	DHCP option data could not be written.
Description	By this API service the TCP/IP stack writes the DHCP option data identified by parameter option.	

5.8.3.3.9. Tcplp_DtlsClose

Purpose	Closure of UDP socket on Tls side.	
Synopsis	<code>void TcpIp_DtlsClose (TcpIp_SocketIdType SocketId , const TcpIp_SockAddrType * RemoteAddrPtr);</code>	
Service ID	0xFD	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	SocketId	Socket identifier of the related local socket resource.
	RemoteAddrPtr	IP address and port of the remote host
Description	Indicates the closure of UDP socket on Tls side, combination of the local socket Id and the remote address	

5.8.3.3.10. Tcplp_DtlsConnectionStateChg

Purpose	Forwards the call to the appropriate upper layer of Tcplp to notify that Tls changed dTLS connection state.	
Synopsis	<code>void TcpIp_DtlsConnectionStateChg (TcpIp_SocketIdType SocketId , const TcpIp_SockAddrType * RemoteAddrPtr , uint8 State);</code>	
Service ID	0xEF	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	SocketId	Socket identifier of the related local socket resource.
	RemoteAddrPtr	IP address and port of the remote host.
	State	New dTLS connection state.
Description	This API shall be used as a passthrough between upper layer and Tls to notify that Tls changed dTLS connection state and will directly call appropriate upper layer function.	



5.8.3.3.11. TcpIp_GetAndResetMeasurementData

Purpose	Reads and resets measurement data for diagnostic purposes.	
Synopsis	<pre>Std_ReturnType TcpIp_GetAndResetMeasurementData (TcpIp_MeasurementIdxType MeasurementIdx , boolean MeasurementResetNeeded , uint32 * MeasurementDataPtr);</pre>	
Parameters (in)	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.
Parameters (out)	MeasurementDataPtr	Pointer to data buffer, where to copy measurement data.
Return Value	Std_ReturnType	
	E_OK	The function has been successfully executed.
	E_NOT_OK	The function could not be successfully executed.
Description	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.

5.8.3.3.12. Tcplp_GetArpCacheEntries

Purpose	Retrieve all valid physical addresses from ARP cache.					
Synopsis	<pre>Std_ReturnType TcpIp_GetArpCacheEntries (uint8 ctrlIdx , uint32 * numberOfElements , TcpIp_ArpCacheEntryType * entryListPtr);</pre>					
Service ID	0x1d					
Sync/Async	Synchronous					
Reentrancy	Non-Reentrant					
Parameters (in)	ctrlIdx	EthIf controller index to identify the related ARP table.				
Parameters (in,out)	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).				
Parameters (out)	entryListPtr	Pointer to memory where the list of cache entries shall be stored.				
Return Value	Result of operation <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">E_OK</td> <td style="padding: 2px;">physical address cache could be read.</td> </tr> <tr> <td style="padding: 2px;">E_NOT_OK</td> <td style="padding: 2px;">physical address cache could not be read (i.e. no IPv4 instance active on this controller)</td> </tr> </table>		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)
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)					
Description	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.					

5.8.3.3.13. Tcplp_GetCtrlIdx

Purpose	Obtain controller index.
----------------	--------------------------



Synopsis	<code>Std_ReturnType TcpIp_GetCtrlIdx (TcpIp_LocalAddrIdType LocalAddrId , uint8 * CtrlIdxPtr);</code>	
Service ID	0x17	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	LocalAddrId	Local address identifier implicitly specifying the EthIf controller that shall be returned
Parameters (out)	CtrlIdxPtr	Pointer to the memory where the index of the controller related to LocalAddrId is stored
Return Value	Result of operation	
	E_OK	the request was successful
	E_NOT_OK	the request was not successful.
Description	Obtains the index of the controller related to LocalAddrId.	

5.8.3.3.14. TcpIp_GetIpAddr

Purpose	Obtain local IP address.	
Synopsis	<code>Std_ReturnType TcpIp_GetIpAddr (TcpIp_LocalAddrIdType LocalAddrId , TcpIp_SockAddrType * IpAddrPtr , uint8 * NetmaskPtr , TcpIp_SockAddrType * DefaultRouterPtr);</code>	
Service ID	0x10	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	LocalAddrId	Local address identifier referring to the local IP address which shall be obtained.
Parameters (out)	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.
Return Value	Result of operation	
	E_OK	The request was successful
	E_NOT_OK	The request was not successful
Description	Obtains the local IP address actually used by LocalAddrId.	

5.8.3.3.15. Tcplp_GetNdpCacheEntries

Purpose	Retrieve all valid physical addresses from NDP cache.	
Synopsis	<pre>Std_ReturnType TcpIp_GetNdpCacheEntries (uint8 ctrlIdx , uint32 * numberOfElements , TcpIp_NdpCacheEntryType * entryListPtr);</pre>	
Service ID	0x1d	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	ctrlIdx	EthIf controller index to identify the related NDP table.
Parameters (in,out)	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).
Parameters (out)	entryListPtr	Pointer to memory where the list of cache entries shall be stored.
Return Value	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)
Description	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 num-	



berOfElements 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.

5.8.3.3.16. TcpIp_GetPhysAddr

Purpose	Obtain local physical address.	
Synopsis	Std_ReturnType TcpIp_GetPhysAddr (TcpIp_LocalAddrIdType Local- AddrId , uint8 * PhysAddrPtr);	
Service ID	0x11	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	LocalAddrId	Local address identifier implicitly specifying the EthIf controller for which the physical address shall be obtained.
Parameters (out)	PhysAddrPtr	Pointer to the memory where the physical source address (MAC address) in network byte order is stored
Return Value	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.
Description	Obtains the physical source address used by the EthIf controller implicitly specified via LocalAddrId.	

5.8.3.3.17. TcpIp_GetRemotePhysAddr

Purpose	Retrieve physical address from remote host.
Synopsis	TcpIp_ReturnType TcpIp_GetRemotePhysAddr (uint8 CtrlIdx , const TcpIp_SockAddrType * IpAddrPtr , uint8 * PhysAddrPtr , boolean initRes);
Service ID	0x16
Sync/Async	Synchronous
Reentrancy	Non-Reentrant



Parameters (in)	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.
Parameters (out)	PhysAddrPtr	Pointer to the memory where the physical address (MAC address)
Return Value	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)
Description	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.	

5.8.3.3.18. Tcplp_GetVersionInfo

Purpose	Get version information of the Tcplp module.	
Synopsis	<pre>void Tcplp_GetVersionInfo (Std_VersionInfoType * versioninfo);</pre>	
Service ID	0x02	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	versioninfo	Pointer to where to store the version information of this module.
Description	<p>This service returns the version information of this module. The version information includes:</p> <ul style="list-style-type: none"> ▶ Module Id ▶ Vendor Id 	



▶ Vendor specific version numbers

5.8.3.3.19. `TcpIp_IcmpTransmit`

Purpose	Requests to transmit an ICMP message.	
Synopsis	<pre>Std_ReturnType TcpIp_IcmpTransmit (TcpIp_LocalAddrIdType LocalIpAddrId , const TcpIp_SockAddrType * RemoteAddrPtr , uint8 Ttl , uint8 Type , uint8 Code , uint16 DataLength , const uint8 * DataPtr);</pre>	
Service ID	0x0C	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	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.
Return Value	Result of operation	
	E_OK	UDP message transmission was successful.
	E_NOT_OK	UDP message transmission failed (E.g. caused by ARP cache miss).
Description	This service transmits an ICMP message	

5.8.3.3.20. `TcpIp_IcmpTransmitError`

Purpose	Requests to transmit an ICMP message based on a (EthIf) CtrlIdx.	
Synopsis	<pre>Std_ReturnType TcpIp_IcmpTransmitError (uint8 CtrlIdx , const TcpIp_SockAddrType * RemoteAddrPtr , uint8 Ttl , uint8 Type ,</pre>	



	<code>uint8 Code , uint16 DataLength , const uint8 * DataPtr , const uint32 * SpecificValue);</code>	
Service ID	0xF2	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	CtrlIdx	Index of the ethernet controller (EthIfCtr-Idx).
	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.
Return Value	Result of operation	
	E_OK	Transmission request has been successfully performed.
	E_NOT_OK	Transmission request failed.
Description	This function constructs the ICMP header and payload and invokes the transmission of the Icmp frame over EthIf.	

5.8.3.3.21. Tcplp_Init

Purpose	Initializes the Tcplp stack.	
Synopsis	<code>void TcpIp_Init (const TcpIp_ConfigType * ConfigPtr);</code>	
Service ID	0x01	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	ConfigPtr	Address of the post-build time configuration data of the Tcplp module.
Description	This service initializes the TCP/IP Stack. The call of this service is mandatory before using the Tcplp instance for further processing.	



5.8.3.3.22. TcpIp_IpSecGetStatus

Purpose	Retrieve current IPsec status.	
Synopsis	<pre>Std_ReturnType TcpIp_IpSecGetStatus (uint8 ctrlIdx , TcpIp_- SockAddrPtrType * sockAddPtr , TcpIp_IpSecIkeSaStatusType * ikeSaStatusPtr , uint8 * numberOfIkeSaElementsPtr);</pre>	
Parameters (in)	ctrlIdx	EthIf controller index to identify the related SAs.
Parameters (in,out)	numberOfIkeSaElementsPtr	Entry: Maximum number of SA statuses and IP addresses that can be stored in the output. Exit: Number of SA statuses and IP addresses that were stored. Special Case: If *numberOfIkeSaElementsPtr is zero on input the total number of installed IKE SAs is returned as output in *numberOfIkeSaElementsPtr
Parameters (out)	sockAddPtr	Pointer to memory where the remote IP addresses shall be stored, can be either a list of IPv4 or IPv6 addresses. Memory should be at least of size *numberOfIkeSaElementsPtr.
	ikeSaStatusPtr	Pointer to memory where the list of IKE SAs and Child SAs shall be stored. Memory should be at least of size *numberOfIkeSaElementsPtr.
Return Value	Result of operation	
	E_OK	IPsec status could be read.
	E_NOT_OK	IPsec status could not be read (e.g. invalid controller, controller is offline)
Description	<p>This service returns the currently installed IKE SAs and Child SAs. If numberOfIkeSaElementsPtr points to zero value on entry, the function returns only the total number of IKE SAs for the given controller in *numberOfIkeSaElementsPtr on exit. If on exit ikeSaStatusPtr->ikeState == TCPIP_IKEV2_SA_INIT_STATE the value of ikeSaStatusPtr->responderSpi is not yet available and should not be used. If on exit ikeSaStatusPtr->ikeState == TCPIP_IKEV2_AUTH_STATE the value of ikeSaStatusPtr->childSaStatusPtr and ikeSaStatusPtr->numberOfChildSaElements is not yet available and should not be used.</p>	



5.8.3.3.23. Tcplp_IsConnectionReady

Purpose	Checks if physical address is known and IpSec SA is established. If physical address is not known address resolution shall be triggered.	
Synopsis	<pre>TcpIp_ReturnType TcpIp_IsConnectionReady (TcpIp_SocketIdType SocketId , const TcpIp_SockAddrType * RemoteAddrPtr);</pre>	
Service ID	0xF5	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SocketId	
Parameters (in)	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.
Return Value	Indicates if data can be transmitted.	
	TCPIP_OK	Physical address known and IpSec SA exists, data can be sent
	TCPIP_E_PENDING	Physical address not known or IpSec SA does not exist
	TCPIP_E_NOT_OK	<ul style="list-style-type: none"> - Transmission not allowed according to policy table OR <ul style="list-style-type: none"> ▶ API called with invalid parameters if development error detection is enabled OR ▶ An error occurs during processing of the request.
Description	<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>	



5.8.3.3.24. `Tcplp_IsValidConfig`

Purpose	Checks compatibility of the post-build-time configuration.	
Synopsis	Std_ReturnType TcpIp_IsValidConfig (const void * voidConfigPtr);	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	voidConfigPtr	Pointer to the configuration data of the Tcplp module.
Return Value	Result of compatibility check	
	E_OK	Provided configuration is compatible.
	E_NOT_OK	Provided configuration is not compatible.
Description	This service checks the compatibility of the post-build-time configuration against the source code.	

5.8.3.3.25. `Tcplp_MainFunction`

Purpose	Tcplp main function.
Synopsis	void TcpIp_MainFunction (void);
Service ID	0x15
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Description	This function performs the periodic actions of the Tcplp module (e.g. timer handling, state machine triggering).

5.8.3.3.26. `Tcplp_MainFunctionTx`

Purpose	MainFunction for socket transmission.
Synopsis	void TcpIp_MainFunctionTx (void);
Description	This function performs transmission tasks. e.g. transmission of TCP segments.



5.8.3.3.27. Tcplp_ReleaseIpAddrAssignment

Purpose	Release an IP address assignment.	
Synopsis	<code>Std_ReturnType TcpIp_ReleaseIpAddrAssignment (TcpIp_LocalAddrIdType LocalAddrId);</code>	
Service ID	0x0b	
Sync/Async	Asynchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	LocalAddrId	IP address index specifying the IP address for which an assignment shall be released.
Return Value	Result of operation	
	E_OK	The request has been accepted
	E_NOT_OK	The request has not been accepted
Description	By this API service the local IP address assignment for the IP address specified by LocalAddrId shall be released.	

5.8.3.3.28. Tcplp_RequestComMode

Purpose	Request to change state of communication network.	
Synopsis	<code>Std_ReturnType TcpIp_RequestComMode (uint8 CtrlIdx , TcpIp_StateType State);</code>	
Service ID	0x09	
Sync/Async	Asynchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	CtrlIdx	EthIf controller index to identify the communication network where the Tcplp state is requested.
	State	Requested Tcplp state.
Return Value	Result of operation	
	E_OK	Service accepted
	E_NOT_OK	Service denied
Description	By this API service the TCP/IP stack is requested to change the Tcplp state of the communication network identified by EthIf controller index.	



5.8.3.3.29. Tcplp_RequestIpAddrAssignment

Purpose	Request an IP address assignment.	
Synopsis	<pre>Std_ReturnType TcpIp_RequestIpAddrAssignment (TcpIp_LocalAddrIdType LocalAddrId , TcpIp_IpAddrAssignmentType Type , const TcpIp_SockAddrType * LocalIpAddrPtr);</pre>	
Service ID	0x0a	
Sync/Async	Asynchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	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.
Return Value	Result of operation	
	E_OK	The request has been accepted
	E_NOT_OK	The request has not been accepted
Description	By this API service the local IP address assignment type for the IP address specified by LocalAddrId shall be requested.	

5.8.3.3.30. Tcplp_RequestIpSecMode

Purpose	Request to change state of communication security.
Synopsis	<pre>Std_ReturnType TcpIp_RequestIpSecMode (uint8 conId , TcpIp_IPsecStateType reqState);</pre>
Service ID	0xF2
Sync/Async	Asynchronous



Reentrancy	Non-Reentrant	
Parameters (in)	conId	IpSec connection index to identify the communication where the Tcplp state is requested.
	reqState	Requested Tcplp state.
Return Value	Result of operation	
	E_OK	Service accepted
	E_NOT_OK	Service denied
Description	By this API service the TCP/IP stack is requested to change the IpSec state of the communication network identified by IpSec connection index.	

5.8.3.3.31. Tcplp_RxIndication

Purpose	Tcplp receive interface.	
Synopsis	<pre>void TcpIp_RxIndication (uint8 CtrlIdx , Eth_FrameType FrameType , boolean IsBroadcast , uint8 * PhysAddrPtr , uint8 * DataPtr , uint16 LenByte);</pre>	
Service ID	0x14	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	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.
Description	This is the receive interface of the Tcplp stack. All received data must be passed to the Tcplp module using this API function.	



5.8.3.3.32. Tcplp_SecurityMainFunction

Purpose	Main function for IKEv2 and IpSec functionality.
Synopsis	<code>void Tcplp_SecurityMainFunction (void);</code>
Service ID	0xF6
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Description	This function performs the periodic actions of the IpSec/IkeV2 unit of the Tcplp module. This includes processing of received IKE messages stored in the Tcplp_IkeV2_rxBuffer. Building and transmitting IKE messages. Handling (ikeSaLifetime/retransmission/deadPeerDetection)-timeouts of the IkeV2 protocol. Deleting and creating IpSec and IkeV2 Security associations. Because many of these functionalities require heavy crypto-operations the execution time of Tcplp_SecurityMainFunction might be quite large. For this reason Tcplp_SecurityMainFunction is exposed to the outside to run in a separate low priority task.

5.8.3.3.33. Tcplp_SetRemotePhysAddr

Purpose	Set physical address of remote host.	
Synopsis	<code>TcpIp_ReturnType Tcplp_SetRemotePhysAddr (uint8 CtrlIdx , const TcpIp_SockAddrType * IpAddrPtr , const uint8 * PhysAddrPtr , uint8 State);</code>	
Service ID	0xF0	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	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)
Return Value	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)
Description	This service adds, updates or removes a physical address from a remote host in the ARP/NDP table	

5.8.3.3.34. Tcplp_Socket_getSocket

Purpose	Allocates resources for a new TCP/UDP socket.	
Synopsis	Std_ReturnType TcpIp_Socket_getSocket (TcpIp_DomainType Domain , TcpIp_ProtocolType Protocol , TcpIp_SocketIdType * SocketIdPtr , uint8 sockOwnerId);	
Service ID	0x03	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	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)
Parameters (out)	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.
Return Value	Result of operation	
	E_OK	The request has been accepted
	E_NOT_OK	The request has not been accepted: no free socket resources found.
Description	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.	

5.8.3.3.35. Tcplp_TcpConnect

Purpose	Trigger the TCP connection establishment to a remote host.	
Synopsis	Std_ReturnType Tcplp_TcpConnect (TcpIp_SocketIdType SocketId , const TcpIp_SockAddrType * RemoteAddrPtr);	
Service ID	0x06	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
Parameters (in)	SocketId	Socket identifier of the related local socket resource.
	RemoteAddrPtr	IP address and port of the remote host to connect to.
Return Value	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.
Description	By this API service the TCP/IP stack is requested to establish a TCP connection to the configured peer.	

5.8.3.3.36. Tcplp_TcpListen

Purpose	Start to listen for incoming TCP connection requests.
----------------	---



Synopsis	<code>Std_ReturnType TcpIp_TcpListen (TcpIp_SocketIdType SocketId , uint16 MaxChannels);</code>	
Service ID	0x07	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
Parameters (in)	SocketId	Socket identifier of the related local socket resource.
	MaxChannels	Maximum number of new parallel connections established on this listen connection.
Return Value	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.
Description	By this API service the TCP/IP stack is requested to listen on the TCP socket specified by the socket identifier.	

5.8.3.3.37. TcpIp_TcpReceived

Purpose	Confirm the reception of data.	
Synopsis	<code>Std_ReturnType TcpIp_TcpReceived (TcpIp_SocketIdType SocketId , uint32 Length);</code>	
Service ID	0x08	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
Parameters (in)	SocketId	Socket identifier of the related local socket resource.
	Length	Number of bytes finally consumed by the upper layer.
Return Value	Result of operation	
	E_OK	The request has been accepted
	E_NOT_OK	The request has not been accepted, e.g. invalid socket id.
Description	By this API service the reception of socket data is confirmed to the TCP/IP stack.	



5.8.3.3.38. Tcplp_TcpTransmit

Purpose	Requests to transmit data to remote destination via TCP protocol.		
Synopsis	<pre>Std_ReturnType Tcplp_TcpTransmit (TcpIp_SocketIdType SocketId , const uint8 * DataPtr , uint32 AvailableLength , boolean ForceRetrieve);</pre>		
Service ID	0x13		
Sync/Async	Asynchronous		
Reentrancy	Reentrant for different SocketIds. Non reentrant for the same SocketId.		
Parameters (in)	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).	
Return Value	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.	



Description	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).
--------------------	--

5.8.3.3.39. Tcplp_TlsClose

Purpose	Tls dedicated API, Closes a socket.	
Synopsis	Std_ReturnType TcpIp_TlsClose (TcpIp_SocketIdType SocketId , boolean Abort);	
Service ID	0xE1	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different SocketId	
Parameters (in)	SocketId	identifying the local socket resource.
	Abort	Terminate immediately?
Return Value	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).
Description	By this API service the TCP/IP stack is requested to close connection and free 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().	

5.8.3.3.40. Tcplp_TlsTcpReceived

Purpose	Tls specific API to confirm the reception of data.
Synopsis	Std_ReturnType TcpIp_TlsTcpReceived (TcpIp_SocketIdType SocketId , uint32 Length);
Service ID	0xE2
Sync/Async	Asynchronous
Reentrancy	Reentrant for different SocketIds. Non reentrant for the same SocketId.



Parameters (in)	SocketId	Socket identifier of the related local socket resource.
	Length	Number of bytes finally consumed by the upper layer.
Return Value	Result of operation	
	E_OK	The request has been accepted
	E_NOT_OK	The request has not been accepted, e.g. invalid socket id.
Description	By this API service the reception of socket data is confirmed by Tls to the TCP/IP stack.	

5.8.3.3.41. Tcplp_TlsTcpTransmit

Purpose	Tls specific API to request transmission of secured data to remote destination via TCP protocol.	
Synopsis	<pre>Std_ReturnType TcpIp_TlsTcpTransmit (TcpIp_SocketIdType SocketId, const uint8 * DataPtr, uint32 AvailableLength, boolean ForceRetrieve);</pre>	
Service ID	0xFC	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
Parameters (in)	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).
Return Value	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.
Description	This service requests transmission only from Tls of already secured 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).	

5.8.3.3.42. Tcplp_TlsUdpTransmit

Purpose	Tls specific API to request transmission of secured data to remote destination via UDP protocol.	
Synopsis	<pre>TcpIp_ReturnType TcpIp_TlsUdpTransmit (TcpIp_SocketIdType socketId , const uint8 * dataPtr , const TcpIp_SockAddrType * remoteAddrPtr , uint16 totalLength);</pre>	
Service ID	0xE0	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SocketId	
Parameters (in)	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.
Return Value		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.
Description	This service request transmission of UDP data secured via TLS to a remote node. The transmission of the data is immediately performed with this function call by forwarding it to EthIf. The parameter 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.	

5.8.3.3.43. Tcplp_UdpTransmit

Purpose	Requests to transmit data to a remote destination via UDP protocol.	
Synopsis	<pre>TcpIp_ReturnType TcpIp_UdpTransmit (TcpIp_SocketIdType socketId , const uint8 * dataPtr , const TcpIp_SockAddrType * remoteAddrPtr , uint16 totalLength);</pre>	
Service ID	0x12	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SocketId	
Parameters (in)	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 DataP-



		tr 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.
Return Value		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.
Description	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 EthIf. The parameter 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.	

5.8.3.3.44. Tcplp_UICopyTxData

Purpose	Forwards requests to copy data for transmission to the appropriate upper layer of Tcplp.	
Synopsis	BufReq_ReturnType TcpIp_UICopyTxData (TcpIp_SocketIdType SocketId , uint8 * BufPtr , uint16 BufLength);	
Service ID	0xFA	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SocketId	
Parameters (in)	SocketId	Socket identifier of the related local socket resource.
	BufLength	Length of provided data buffer.



Parameters (out)	BufPtr	Pointer to buffer for transmission data.
Return Value	Status of the copy request	
	BUFREQ_OK	Data has been copied to the transmit buffer completely as requested
	BUFREQ_E_NOT_OK	Data has not been copied. Request failed.

5.8.3.3.45. Tcplp_UIRxIndication

Purpose	Data reception of UDP datagram or TCP stream.	
Synopsis	<pre>void TcpIp_U1RxIndication (TcpIp_SocketIdType SocketId , const TcpIp_SockAddrType * RemoteAddrPtr , uint8 * BufPtr , uint16 Length);</pre>	
Service ID	0xF9	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	SocketId	Socket identifier of the related 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
Description	Indicates the reception of a new TCP segment or UDP datagram and provides all data via BufPtr.	

5.8.3.3.46. Tcplp_UITcpIpEvent

Purpose	Forwards the call to the appropriate upper layer of Tcplp to report the socket event for establishing the TLS handshake and the connection.	
Synopsis	<pre>void TcpIp_U1TcpIpEvent (TcpIp_SocketIdType SocketId , TcpIp_EventType Event);</pre>	
Service ID	0xFE	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SocketId	



Parameters (in)	SocketId	Socket identifier of the related local socket resource.
	Event	This parameter contains a description of the event just encountered.

5.8.3.3.47. Tcplp_UITxConfirmation

Purpose	Forwards the call to the appropriate upper layer of Tcplp after the data has been acknowledged by the peer for TCP.	
Synopsis	<pre>void TcpIp_UITxConfirmation (TcpIp_SocketIdType SocketId , uint16 DataToFree);</pre>	
Service ID	0xFB	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different SocketId	
Parameters (in)	SocketId	Socket identifier of the related local socket resource.
	DataToFree	Number of transmitted data bytes that have been acknowledged.

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

Protected data structures	All Tcplp socket 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.

5.8.4.1.2. SCHM_TCPIP_EXCLUSIVE_AREA_1

Protected data structures	All IPv4 ARP table data that shall be protected from mutual access.
Recommended locking mechanism	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

Protected data structures	All state machine data that shall be protected from mutual access.
Recommended locking mechanism	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

Protected data structures	All state machine locking data that shall be protected from mutual access.
----------------------------------	--



Recommended locking mechanism	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:

Memory section
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

Description	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.
Rationale	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

Description	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.
Rationale	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

Description	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().
Rationale	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

Description	Eth and EthIf must not use receive or transmit interrupts. The Integrator must assure that neither Eth nor EthIf use interrupts. That is, EthIfEnableRxInterrupt, EthCtrlEn-
--------------------	--



	ableRxInterrupt, EthIfEnableTxInterrupt and EthCtrlEnableTxInterrupt shall be set to false. This prevents Tcplp_RxIndication() from interrupting Tcplp_MainFunction().
Rationale	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

Description	Tcplp_MainFunctionTx() shall have the same preemption constraints as Tcplp_MainFunction().
--------------------	--

5.8.4.4.6. Tcplp.EB_INTREQ_Tcplp_0006

Description	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

Description	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

Description	The 3rd argument (PhysAddrPtr) passed to Tcplp_GetRemotePhysAddr has to point to at least 6 bytes of allocated memory.
--------------------	--

5.8.4.4.15. Tcplp.EB_INTREQ.localCertificates

Description	The Integrator must assure that locally configured certificates are read-in before the initialization of Tcplp.
--------------------	---



5.8.4.4.16. Tcplp.EB_INTREQ.SecurityMainFunction

Description	The priority of the Task that runs Tcplp_SecurityMainFunction shall be smaller than any other task that calls an external Tcplp-Module API. That means that Tcplp_SecurityMainFunction shall not preempt any other external Tcplp module API.
--------------------	---

5.8.4.4.17. Tcplp.EB_INTREQ.IpSecGetStatus

Description	Depending on the value of ikeSaStatusPtr->ikeState the following values returned by Tcplp_IpSecGetStatus should not be used: If ikeSaStatusPtr->ikeState == TCPIP_IKEV2_SA_INIT_STATE (the local has send an INIT request and is waiting/processing the INIT response) the variable ikeSaStatusPtr->responderSpi shall not be used. If ikeSaStatusPtr->ikeState == TCPIP_IKEV2_AUTH_STATE (the local has send an AUTH request or INIT request and is waiting/processing the AUTH response or AUTH request) the pointer ikeSaStatusPtr->childSaStatusPtr and the variable ikeSaStatusPtr->numberOfChildSaElements shall not be used.
--------------------	---

5.8.4.4.18. Tcplp.EB_INTREQ.AppendingEsn

Description	If TcplpSecurityMode = 'IPSEC_IKE' and TcplpIpFragmentationRxEnabled = FALSE the config variable EthCtrlConfigIngressFifoBufLenByte in the Eth module configuration must be at least 22 bytes larger than config variable EthIfCtrlMtu in the EthIf module configuration. Rational: If an IPsec SA uses extended sequence numbers (ESN) and TcplpIpFragmentationRxEnabled = FALSE the 4 high bytes of the ESN are appended to the IP payload before the integrity value is verified. This integration requirement guarantees that additional 4 bytes are available after the IP payload in the Eth controller RX buffer.
--------------------	--

5.9. UdpNm

5.9.1. Configuration parameters

Containers included		
Container name	Multiplicity	Description



Containers included

CommonPublishedInformation	1..1	Label: Common Published Information Common container, aggregated by all modules. It contains published information about vendor and versions.
PublishedInformation	1..1	Label: EB Published Information Additional published parameters not covered by CommonPublishedInformation container.
UdpNmGeneral	1..1	
UdpNmGlobalConfig	1..1	This container contains all global configuration parameters of UDP NM configured from the NM Module perspective.
UdpNmDefensiveProgramming	1..1	Label: Defensive Programming Options Parameters for defensive programming

Parameters included

Parameter name	Multiplicity
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT	
Label	Config Variant	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	VariantPostBuild	
Range	VariantPostBuild	
Configuration class	VariantPostBuild:	VariantPostBuild

5.9.1.1. CommonPublishedInformation

Parameters included

Parameter name	Multiplicity
ArMajorVersion	1..1
ArMinorVersion	1..1
ArPatchVersion	1..1
SwMajorVersion	1..1
SwMinorVersion	1..1

Parameters included

SwPatchVersion	1..1
ModuleId	1..1
VendorId	1..1
Release	1..1

Parameter Name

Parameter Name	ArMajorVersion
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	3
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name

Parameter Name	ArMinorVersion
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	3
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name

Parameter Name	ArPatchVersion
Label	AUTOSAR Patch Version
Description	Patch level version number of AUTOSAR specification on which the appropriate implementation is based on.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	0
Configuration class	PublishedInformation:

Origin	Elektrobit Automotive GmbH
Parameter Name	SwMajorVersion
Label	Software Major Version
Description	Major version number of the vendor specific implementation of the module.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	2
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH
Parameter Name	SwMinorVersion
Label	Software Minor Version
Description	Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	9
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH
Parameter Name	SwPatchVersion
Label	Software Patch Version
Description	Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	11
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH
Parameter Name	ModuleId
Label	Numeric Module ID
Description	Module ID of this module from Module List
Multiplicity	1..1

Type	INTEGER_LABEL
Default value	33
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	VendorId
Label	Vendor ID
Description	Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list
Multiplicity	1..1
Type	INTEGER_LABEL
Default value	1
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

Parameter Name	Release
Label	Release Information
Multiplicity	1..1
Type	STRING_LABEL
Default value	
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

5.9.1.2. PublishedInformation

Parameters included	
Parameter name	Multiplicity
PbcfgMSupport	1..1

Parameter Name	PbcfgMSupport
Label	PbcfgM support
Description	Specifies whether or not the UdpNm can use the PbcfgM module for post-build support.
Multiplicity	1..1



Type	BOOLEAN
Default value	true
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

5.9.1.3. UdpNmGeneral

Parameters included	
Parameter name	Multiplicity
UdpNmMultiCoreSupport	1..1
UdpNmPnSupported	1..1
UdpNmASR412APICompatibility	1..1
UdpNmRelocatablePbcfgEnable	1..1
UdpNmMaxPn	0..1
UdpNmDetRuntimeChecks	1..1

Parameter Name	UdpNmMultiCoreSupport
Label	UdpNm multicore support
Description	Enables MultiCoreSupport.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild:
Origin	Elektrobit Automotive GmbH

Parameter Name	UdpNmPnSupported
Label	Support for Partial Network Cluster (PNC)
Description	Enables or disables support of partial networking. <ul style="list-style-type: none"> ▶ False: Partial Networking is disabled ▶ True: Partial Networking is enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	UdpNmASR412APICompatibility	
Label	UdpNm_SoAdIfRxIndication ASR412 Compatibility	
Description	<p>UdpNm_SoAdIfRxIndication compatibility with ASR 4.1.2.</p> <ul style="list-style-type: none"> ▶ False: UdpNm_SoAdIfRxIndication ASR 4.1.3 compatibility ▶ True: UdpNm_SoAdIfRxIndication ASR 4.1.2 compatibility 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	UdpNmRelocatablePbcfgEnable	
Label	UdpNmRelocatablePbcfgEnable	
Description	<p>Enables/disables support for relocatable postbuild configuration.</p> <ul style="list-style-type: none"> ▶ True: Postbuild configuration relocatable in memory. ▶ False: Postbuild configuration not relocatable in memory. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

Parameter Name	UdpNmMaxPn	
Label	UdpNmMaxPn	
Description	The maximum number of Partial Networking Clusters that can be configured.	
Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	



Parameter Name	UdpNmDetRuntimeChecks	
Label	DET runtime checks	
Description	<p>Enables or disables the reporting of Default Error Trace runtime checks for the module UdpNm via Det_ReportRuntimeError() and Det_ReportTransientFault().</p> <p>Note: This parameter enables/disables the reporting. The runtime check itself won't be disabled.</p> <ul style="list-style-type: none"> ▶ True: Reporting of runtime checks enabled ▶ False: Reporting of runtime checks disabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	

5.9.1.4. UdpNmGlobalConfig

Containers included		
Container name	Multiplicity	Description
UdpNmChannelConfig	1..n	Label: Channel Configuration This container holds the channel specific configuration parameter of the UdpNm.
UdpNmPnInfo	0..1	PN information configuration.

Parameters included		
Parameter name	Multiplicity	
UdpNmActiveWakeupBitEnabled	1..1	
UdpNmBusSynchronizationEnabled	1..1	
UdpNmComControlEnabled	1..1	
UdpNmNodeIdCallback	0..1	
UdpNmNodeIdCallbackHeader	1..1	
UdpNmComUserDataSupport	1..1	
UdpNmCoordinatorSyncSupport	1..1	
UdpNmDevErrorDetect	1..1	



Parameters included

UdpNmImmediateRestartEnabled	1..1
UdpNmMainFunctionPeriod	1..1
UdpNmNumberOfChannels	1..1
UdpNmPostBuildRamSize	1..1
UdpNmPassiveModeEnabled	1..1
UdpNmPduRxIndicationEnabled	1..1
UdpNmPnEiraCalcEnabled	0..1
UdpNmPnResetTime	0..1
UdpNmRemoteSleepIndEnabled	1..1
UdpNmStateChangeIndEnabled	1..1
UdpNmUserDataEnabled	1..1
UdpNmVersionInfoApi	1..1
UdpNmPnEiraRxNSduRef	0..1
UdpNmPnShutdownMessageRetransmissionDuration	0..1
UdpNmPnSyncShutdownErrorReactionEnabled	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	<p>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.</p> <p>The following API function is provided:</p> <ul style="list-style-type: none"> ▶ <code>UdpNm_RequestBusSynchronization()</code> 	



	<p>Dependencies:</p> <ul style="list-style-type: none"> ▶ The value of this parameter has to be synchronized with the value of respective parameter in the Nm module. ▶ Passive Mode must be disabled.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECU

Parameter Name	UdpNmComControlEnabled
Label	Communication Control
Description	<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"> ▶ The value of this parameter has to be synchronized with the value of respective parameter in the Nm module. ▶ Passive Mode must be disabled.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECU

Parameter Name	UdpNmNodeIdCallback
Label	UdpNm Node Id callback
Description	Name of the callback function to be called if UdpNmNodeIdCallback is enabled.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	UdpNmNodeIdCallbackHeader
Label	Node Id callback header



Description	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.
Multiplicity	1..1
Type	STRING
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	UdpNmComUserDataSupport
Description	Enable/disable the user data support.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmCoordinatorSyncSupport
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> Enables/disables the coordinator synchronisation support.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmDevErrorDetect
Label	Enable Development Error Detection
Description	Pre-processor switch for enabling development error detection support.
Multiplicity	1..1
Type	BOOLEAN
Default value	true



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	UdpNmImmediateRestartEnabled	
Label	Immediate Restart	
Description	Enabling the asynchronous transmission of a NM PDU upon bus communication request in Prepare-Bus-Sleep mode.	
	Dependencies: ▶ Passive Mode must be disabled.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	UdpNmMainFunctionPeriod	
Label	Main Function Period [s]	
Description	Call cycle in seconds of UdpNm_MainFunction <code>UdpNm_MainFunction_x</code>	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.001	
Range	<=0.255 >=0.001	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	UdpNmNumberOfChannels	
Label	Number Of Channels	
Description	Maximum number of Udp NM channels allowed within one ECU.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Configuration class	VariantPostBuild:	VariantPostBuild



Origin	AUTOSAR_ECUC
Parameter Name	UdpNmPostBuildRamSize
Label	UdpNmPostBuildRamSize
Description	<p>Number of bytes for TX and RX buffers</p> <p>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 lenghts at postbuild time</p>
Multiplicity	1..1
Type	INTEGER
Default value	96
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH
Parameter Name	UdpNmPassiveModeEnabled
Label	Passive Mode
Description	<p>Pre-processor switch for enabling support of the Passive Mode. This parameter shall be derived from NM_PASSIVE_MODE_ENABLED.</p> <p>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.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC
Parameter Name	UdpNmPduRxIndicationEnabled
Label	PDU Receive Indication
Description	<p>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</p> <p>Dependencies:</p> <ul style="list-style-type: none"> ▶ The value of this parameter has to be synchronized with the value of respective parameter in the Nm module.



Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmPnEiraCalcEnabled
Description	Specifies if UdpNm calculates the PN request information for internal and external requests(EIRA). <ul style="list-style-type: none"> ▶ true: PN request are calculated ▶ false: PN request are not calculated
Multiplicity	0..1
Type	BOOLEAN
Default value	false
Configuration class	PreCompile: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmPnResetTime
Description	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.
Multiplicity	0..1
Type	FLOAT
Default value	0.01
Range	<=65.535 >=0.0010
Configuration class	PreCompile: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmRemoteSleepIndEnabled
Label	Remote Sleep Indication
Description	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.



	<p>Therefore the following API is provided:</p> <ul style="list-style-type: none"> ▶ <code>UdpNm_CheckRemoteSleepIndication()</code> <p>Dependencies:</p> <ul style="list-style-type: none"> ▶ The value of this parameter has to be synchronized with the value of respective parameter in the Nm module. ▶ Passive Mode must be disabled.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmStateChangeIndEnabled
Label	State Change Indication
Description	Pre-processor switch for enabling the UDP NM state change notification. This parameter shall be derived from NM_STATE_CHANGE_IND_ENABLED.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmUserDataEnabled
Label	User Data
Description	<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"> ▶ <code>UdpNm_GetUserData()</code> ▶ <code>UdpNm_SetUserData()</code> (<i>Only if Passive Mode Support is disabled</i>) <p>Dependencies:</p> <ul style="list-style-type: none"> ▶ The value of this parameter has to be synchronized with the value of respective parameter in the Nm module.



Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	UdpNmVersionInfoApi	
Label	Enable Version Info API	
Description	Pre-processor switch for enabling version info API support. Provide API function for retrieving version information: ► UdpNm_GetVersionInfo()	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	UdpNmPnEiraRxNSduRef	
Description	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.	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	UdpNmPnShutdownMessageRetransmissionDuration	
Label	UdpNmPnShutdownMessageRetransmissionDuration	
Description	Specifies the duration in seconds of the retransmission phase of a PN shutdown message. A retransmission shall be performed per affected NM channel, as long as the PN shutdown message could not be successfully sent and the retransmission timer is running. The value shall be a multiple integral of UdpNmMainFunctionPeriod.	
Multiplicity	0..1	
Type	FLOAT	
Configuration class	PreCompile:	VariantPostBuild



Origin	AUTOSAR_ECUC
Parameter Name	UdpNmPnSyncShutdownErrorReactionEnabled
Label	UdpNmPnSyncShutdownErrorReactionEnabled
Description	Pre-processor switch for enabling reaction, if a top-level PNC coordinator received a PN shutdown message on a NM-channel which refer to a ComM channel that is actively coordinated by a PNC gateway.
Multiplicity	0..1
Type	BOOLEAN
Default value	false
Configuration class	PreCompile: VariantPostBuild
Origin	AUTOSAR_ECUC

5.9.1.5. UdpNmChannelConfig

Containers included		
Container name	Multiplicity	Description
UdpNmRxPdu	1..n	<p>Label: Receive PDU ID and Reference</p> <p>This container holds the <code>UdpNmRxPduId</code> and the <code>UdpNmRxPduRef</code>.</p>
UdpNmTxPdu	0..1	<p>Label: Transmit PDU Reference</p> <p>This container contains the <code>UdpNmTxConfirmationPduId</code> and the <code>UdpNmTxPduRef</code>.</p>
UdpNmUserDataTxPdu	0..1	<p>Label: User Data Transmission PDU</p> <p>This optional container is used to configure the UserNm PDU. This container is only available if <code>UdpNmComUserDataSupport</code> is enabled.</p>
UdpNmUserDataRxPdu	0..1	<p>Label: 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
UdpNmNodeldEnabled	1..1



Parameters included

UdpNmRepeatMsgIndEnabled	1..1
UdpNmNodeDetectionEnabled	1..1
UdpNmAllNmMessagesKeepAwake	1..1
UdpNmCarWakeUpBitPosition	1..1
UdpNmCarWakeUpBytePosition	1..1
UdpNmCarWakeUpFilterEnabled	1..1
UdpNmCarWakeUpFilterNodeId	1..1
UdpNmCarWakeUpRxEnabled	1..1
UdpNmPnEnabled	0..1
UdpNmPnEraCalcEnabled	0..1
UdpNmPnHandleMultipleNetworkRequests	0..1
UdpNmSynchronizedPncShutdownEnabled	0..1
UdpNmPnEraRxNsduRef	0..1
UdpNmImmediateNmCycleTime	1..1
UdpNmImmediateNmTransmissions	1..1
UdpNmRetryFirstMessageRequest	1..1
UdpNmMsgCycleOffset	1..1
UdpNmMsgCycleTime	1..1
UdpNmMsgTimeoutTime	1..1
UdpNmNodeId	1..1
UdpNmPduCbvPosition	1..1
UdpNmPduNidPosition	1..1
UdpNmRemoteSleepIndTime	1..1
UdpNmRepeatMessageTime	1..1
UdpNmTimeoutTime	1..1
UdpNmWaitBusSleepTime	1..1
UdpNmComMNetworkHandleRef	1..1

Parameter Name	UdpNmNodeIdEnabled
Label	Node Identifier
Description	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>Therefore the following API functions are provided:</p> <ul style="list-style-type: none"> ▶ <code>Nm_GetNodeIdentifier()</code> ▶ <code>Nm_GetLocalNodeIdentifier()</code> <p>Dependencies:</p> <ul style="list-style-type: none"> ▶ The value of this parameter has to be synchronized with the value of the respective parameter in the Nm module.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC
Parameter Name	UdpNmRepeatMsgIndEnabled
Label	Repeat Message Indication
Description	Enable/disable the notification that a RepeatMessageRequest bit has been received. This parameter shall be derived from NM_REPEAT_MSG_IND_ENABLED. Dependencies: ▶ Node Detection must be enabled.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC
Parameter Name	UdpNmNodeDetectionEnabled
Label	Node Detection
Description	Pre-processor switch for enabling the node detection support. This parameter shall be derived from NM_NODE_DETECTION_ENABLED. This parameter shall only be enabled if UDPNM_NODE_ID_ENABLED is defined. If(UdpNmPduCbvPosition != UDPNM_PDU_OFF) then Equal(NmNodeDetectionEnabled) else Equal(False). For setting the <i>Repeat Message Request Bit</i> in NM messages following API function is provided:



	<ul style="list-style-type: none"> ▶ <code>UdpNm_RepeatMessageRequest()</code> <p>Dependencies:</p> <ul style="list-style-type: none"> ▶ The value of this parameter has to be synchronized with the value of the respective parameter in the Nm module. ▶ Support for Node Identifiers must be enabled. ▶ Passive Mode must be disabled.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmAllNmMessagesKeepAwake
Label	UdpNmAllNmMessagesKeepAwake
Description	Specifies if UdpNm drops irrelevant NM PDUs
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	UdpNmCarWakeUpBitPosition		
Description	Specifies the Bit position of the CWU within the UdpNmCarWakeUpBytePosition.		
Multiplicity	1..1		
Type	INTEGER		
Default value	0		
Range	<table border="1" style="width: 100%;"> <tr> <td><=7</td> </tr> <tr> <td>>=0</td> </tr> </table>	<=7	>=0
<=7			
>=0			
Configuration class	PreCompile: VariantPostBuild		
Origin	AUTOSAR_ECUC		

Parameter Name	UdpNmCarWakeUpBytePosition
-----------------------	-----------------------------------



Description	Specifies the Byte position of the CWU within the NM-Message.	
Multiplicity	1..1	
Type	INTEGER	
Default value	2	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	UdpNmCarWakeUpFilterEnabled	
Description	If CWU filtering is supported, only the CWU bit within the NM PDU message with source node identifier UdpNmCarWakeUpFilterNodeld is considered as CWU request.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	UdpNmCarWakeUpFilterNodeld	
Description	Source node identifier for CWU filtering. If CWU filtering is supported, only the CWU bit within the NM message with source node identifier UdpNmCarWakeUpFilterNodeld is considered as CWU request.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255 >=0	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	UdpNmCarWakeUpRxEnabled	
Description	Enables or disables support of CarWakeUp bit evaluation in received NM messages.	
Multiplicity	1..1	
Type	BOOLEAN	



Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmPnEnabled
Description	Enables or disables support of partial networking. <ul style="list-style-type: none">▶ true: Partial networking supported▶ false: Partial networking not supported
Multiplicity	0..1
Type	BOOLEAN
Default value	false
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmPnEraCalcEnabled
Description	Specifies if UdpNm calculates the PN request information for external requests.(ERA)
Multiplicity	0..1
Type	BOOLEAN
Default value	false
Configuration class	PreCompile: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmPnHandleMultipleNetworkRequests
Description	<ul style="list-style-type: none">▶ true: UdpNm_NetworkRequest triggers a change from NO to RM.▶ false: UdpNm_NetworkRequest is ignored in NO. <p>Dependencies:</p> <ul style="list-style-type: none">▶ Support for Partial Networks must be enabled for this channel
Multiplicity	0..1
Type	BOOLEAN
Default value	false
Configuration class	PostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC



Parameter Name	UdpNmSynchronizedPncShutdownEnabled	
Label	UdpNmSynchronizedPncShutdownEnabled	
Description	Specifies if UdpNm handle PN shutdown messages to support a synchronized PNC shutdown across a PN topology. This is only used for ECUs in the role of a top-level PNC coordinator or intermediate PNC coordinator. Thus, the PNC gateway functionality is enabled and therefore ERA calculation is used.	
Multiplicity	0..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	UdpNmPnEraRxNSduRef	
Description	Reference to a Pdu in the COM-Stack. The SduRef is required for every UdpNm Channel, because ERA is reported per channel.	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	UdpNmImmediateNmCycleTime	
Label	Immediate NM PDU cycle time	
Description	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: ▶ This parameter is active only if UdpNmImmediateNmTransmissions greater than one	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.001	
Configuration class	PreCompile:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	UdpNmImmediateNmTransmissions	
-----------------------	--------------------------------------	--



Label	Number of immediate NM PDUs
Description	<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"> ▶ <code>UdpNmImmediateNmCycleTime</code> is active only if this parameter greater than zero
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECU

Parameter Name	UdpNmRetryFirstMessageRequest
Label	UdpNmRetryFirstMessageRequest
Description	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).
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PreCompile: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	UdpNmMsgCycleOffset
Label	Message Cycle Offset [s]
Description	<p>Time offset in the periodic transmission node. It determines the start delay of the transmission. &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"> ▶ If Passive Mode is enabled this parameter is ignored.



	<ul style="list-style-type: none"> ▶ The Message Cycle Offset must be smaller than the Message Cycle Time. ▶ The value must be multiple of the Main Function Period.
Multiplicity	1..1
Type	FLOAT
Default value	0.001
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC
Parameter Name	UdpNmMsgCycleTime
Label	Message Cycle Time [s]
Description	<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"> ▶ If Passive Mode is enabled this parameter is ignored. ▶ The value must be multiple of the Main Function Period.
Multiplicity	1..1
Type	FLOAT
Default value	0.002
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC
Parameter Name	UdpNmMsgTimeoutTime
Label	Message Timeout Time [s]
Description	<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"> ▶ If Passive Mode is enabled this parameter is ignored. ▶ Value must be a multiple of the Main Function Period. ▶ Value must be less than the Message Cycle Time.
Multiplicity	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"> ▶ This parameter is only valid if UdpNmPassiveModeEnabled = False ▶ If the Node Identifier Position is set to UDPNM_PDU_OFF this parameter is ignored.
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<p><=255</p> <p>>=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; 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; 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"> ▶ If Node Detection support is disabled this parameter is ignored. ▶ The Control Bit Vector must not occupy the same byte as the Node Id.
Multiplicity	1..1
Type	ENUMERATION
Default value	UDPNM_PDU_BYTE_1
Range	UDPNM_PDU_BYTE_0 UDPNM_PDU_BYTE_1 UDPNM_PDU_OFF
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECU

Parameter Name	UdpNmPduNidPosition
Label	PDU Node Identifier Position
Description	<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 && UDPNM_PDU_CBV_POSITION != UDPNM_PDU_OFF) then UDPNM_PDU_NID_POSITION != UDPNM_PDU_CBV_POSITION if (UDPNM_PDU_NID_POSITION != UDPNM_PDU_OFF && 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"> ▶ If Node Id support is disabled this parameter is ignored. ▶ The Node Id must not occupy the same byte as the Control Bit Vector
Multiplicity	1..1
Type	ENUMERATION
Default value	UDPNM_PDU_BYTE_0
Range	UDPNM_PDU_BYTE_0 UDPNM_PDU_BYTE_1 UDPNM_PDU_OFF
Configuration class	VariantPostBuild: VariantPostBuild



Origin	AUTOSAR_ECUC
---------------	--------------

Parameter Name	UdpNmRemoteSleepIndTime	
Label	Remote Sleep Indication Time [s]	
Description	<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: $n * \text{UDPNM_MSG_CYCLE_TIME}$, where n denotes the number of NM packets that are normally sent before Remote Sleep Indication is detected. The value of n 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"> ▶ $\text{UdpNmRemoteSleepIndTime} \geq \text{UdpNmMsgCycleTime}$ 	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.001	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	UdpNmRepeatMessageTime	
Label	Repeat Message Time [s]	
Description	<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: $n * \text{UDPNM_MSG_CYCLE_TIME}$, where n denotes the number of NM packets that are normally sent in the Repeat Message State. The value of n 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"> ▶ If Passive Mode is enabled this parameter is ignored. ▶ Value must be a multiple of the Main Function Period. 	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.001	



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	UdpNmTimeoutTime	
Label	Timeout Time [s]	
Description	<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: $x * UDPNM_MSG_CYCLE_TIME$, 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"> ▶ Value must be a multiple of the Main Function Period. ▶ The UdpNmTimeoutTime must be greater than UdpNmPnResetTime. ▶ The Timeout Time must be greater than Message cycle time. ▶ The Timeout Time must be a multiple of the Main Function Period. 	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.004	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECU	

Parameter Name	UdpNmWaitBusSleepTime	
Label	Wait Bus Sleep Time [s]	
Description	<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"> ▶ Value must be a multiple of the Main Function Period. 	
Multiplicity	1..1	
Type	FLOAT	



Default value	0.004
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmComMNetworkHandleRef	
Description	This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.9.1.6. UdpNmRxPdu

Parameters included	
Parameter name	Multiplicity
UdpNmRxPduld	1..1
UdpNmRxPduRef	1..1

Parameter Name	UdpNmRxPduld
Label	Receive PDU ID
Description	ID of the RxPdu that will be used by a RxIndication of the lower layer.
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmRxPduRef
Label	Receive PDU Reference
Description	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.
Multiplicity	1..1
Type	REFERENCE



Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.9.1.7. UdpNmTxPdu

Parameters included	
Parameter name	Multiplicity
UdpNmTxConfirmationPduld	1..1
UdpNmTxPduRef	1..1

Parameter Name	UdpNmTxConfirmationPduld
Description	Id of the TxPdu that will be used by a TxConfirmation from the lower layer.
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmTxPduRef
Label	Transmit PDU Reference
Description	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.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild:
Origin	AUTOSAR_ECUC

5.9.1.8. UdpNmUserDataTxPdu

Parameters included	
Parameter name	Multiplicity
UdpNmTxUserDataPduld	1..1



Parameters included

UdpNmTxUserDataPduRef	1..1
---------------------------------------	------

Parameter Name	UdpNmTxUserDataPduld
Description	This parameter defines the Handle ID of the NM User Data I-PDU.
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECU

Parameter Name	UdpNmTxUserDataPduRef
Description	Reference to the NM User Data I-PDU in the global PDU collection.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECU

5.9.1.9. UdpNmUserDataRxPdu

Parameters included	
Parameter name	Multiplicity
UdpNmRxUserDataPduRef	1..1

Parameter Name	UdpNmRxUserDataPduRef
Description	Reference to the Rx NM User Data I-PDU in the global PDU collection
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

5.9.1.10. UdpNmPnInfo

Containers included		
Container name	Multiplicity	Description



Containers included

UdpNmPnFilterMaskByte	0..63	PN information configuration.
---------------------------------------	-------	-------------------------------

Parameters included

Parameter name	Multiplicity
UdpNmPnInfoLength	1..1
UdpNmPnInfoOffset	1..1

Parameter Name	UdpNmPnInfoLength
Description	Specifies the length of the PN request information in the NM message.
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<=63 >=1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

Parameter Name	UdpNmPnInfoOffset
Description	Specifies the offset of the PN request information in the NM message.
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<=63 >=1
Configuration class	VariantPostBuild: VariantPostBuild
Origin	AUTOSAR_ECUC

5.9.1.11. UdpNmPnFilterMaskByte

Parameters included

Parameter name	Multiplicity
UdpNmPnFilterMaskByteIndex	1..1



Parameters included

UdpNmPnFilterMaskByteValue	1..1
--	------

Parameter Name	UdpNmPnFilterMaskByteIndex	
Description	Index of the filter mask byte. Specifies the position within the filter mask	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

Parameter Name	UdpNmPnFilterMaskByteValue	
Description	Parameter to configure the filter mask byte.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255 >=0	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	AUTOSAR_ECUC	

5.9.1.12. UdpNmDefensiveProgramming

Parameters included	
Parameter name	Multiplicity
UdpNmDefProgEnabled	1..1
UdpNmPrecondAssertEnabled	1..1
UdpNmPostcondAssertEnabled	1..1
UdpNmStaticAssertEnabled	1..1
UdpNmUnreachAssertEnabled	1..1
UdpNmInvariantAssertEnabled	1..1

Parameter Name	UdpNmDefProgEnabled
----------------	-------------------------------------



Label	Enable Defensive Programming
Description	<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"> 1. Enable development error detection 2. Enable defensive programming 3. Enable assertions as required
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	UdpNmPrecondAssertEnabled
Label	Enable Precondition Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>UdpNmDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>UdpNmDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	UdpNmPostcondAssertEnabled
Label	Enable Postcondition Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>UdpNmDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>UdpNmDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	UdpNmStaticAssertEnabled
Label	Enable Static Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>UdpNmDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>UdpNmDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPostBuild: VariantPostBuild
Origin	Elektrobit Automotive GmbH

Parameter Name	UdpNmUnreachAssertEnabled
Label	Enable Unreachable Code Assertions
Description	<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"> ▶ Enable Development Error Detection (<code>UdpNmDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>UdpNmDefProgEnabled</code>): must be enabled
Multiplicity	1..1



Type	BOOLEAN	
Default value	false	
Configuration class	VariantPostBuild:	VariantPostBuild
Origin	Elektrobit Automotive GmbH	
Parameter Name	UdpNmInvariantAssertEnabled	
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"> ▶ Enable Development Error Detection (<code>UdpNmDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>UdpNmDefProgEnabled</code>): must be enabled 	
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.
---------	--------------------------------



Value	1U
--------------	----

5.9.2.1.3. UDPNM_AR_RELEASE_REVISION_VERSION

Purpose	AUTOSAR release revision version.
Value	3U

5.9.2.1.4. UDPNM_E_BUSSLEEPMODE

Purpose	Error code for case in which SchM_Call for Nm_BusSleepMode fails.
Value	248U

5.9.2.1.5. UDPNM_E_CARWAKEUPINDICATION

Purpose	Error code for case in which SchM_Call for Nm_CarWakeUpIndication fails.
Value	241U

5.9.2.1.6. UDPNM_E_INIT_FAILED

Purpose	UdpNm initialization has failed, e.g. selected configuration set doesn't exist.
Value	0x04U

5.9.2.1.7. UDPNM_E_INVALID_CHANNEL

Purpose	Error Code for Invalid channel.
Value	0x02U

5.9.2.1.8. UDPNM_E_INVALID_FUNCTION_ARG

Purpose	Error code for other invalid API function argument in API.
Value	0x23U



5.9.2.1.9. UDPNM_E_INVALID_PDUID

Purpose	API service called with wrong PDU ID.
Value	0x03U

5.9.2.1.10. UDPNM_E_INVALID_PN_SYNC_SHUTDOWN_REQUEST

Purpose	Error code for case in which a PN synchronize shutdown message is received on an active channel.
Value	0x20U

5.9.2.1.11. UDPNM_E_NETWORKMODE

Purpose	Error code for case in which SchM_Call for Nm_NetworkMode fails.
Value	249U

5.9.2.1.12. UDPNM_E_NETWORKSTARTINDICATION

Purpose	Error code for case in which SchM_Call for Nm_NetworkStartIndication fails.
Value	250U

5.9.2.1.13. UDPNM_E_NO_INIT

Purpose	Initialization status before module initilaization.
Value	0x01U

5.9.2.1.14. UDPNM_E_NULL_POINTER

Purpose	Error code for NULL pointers.
Value	0x12U

5.9.2.1.15. UDPNM_E_PDURXINDICATION

Purpose	Error code for case in which SchM_Call for Nm_PduRxIndication fails.
---------	--



Value	244U
--------------	------

5.9.2.1.16. UDPNM_E_PREPAREBUSSLEEPMODE

Purpose	Error code for case in which SchM_Call for Nm_PreparesBusSleepMode fails.
Value	247U

5.9.2.1.17. UDPNM_E_REMOTESENTRYCANCELLATION

Purpose	Error code for case in which SchM_Call for Nm_RemoteSleepCancellation fails.
Value	245U

5.9.2.1.18. UDPNM_E_REMOTESENTRYINDICATION

Purpose	Error code for case in which SchM_Call for Nm_RemoteSleepIndication fails.
Value	246U

5.9.2.1.19. UDPNM_E_REPEATMESSAGEINDICATION

Purpose	Error code for case in which SchM_Call for Nm_RepeatMessageIndication fails.
Value	242U

5.9.2.1.20. UDPNM_E_STATECHANGENOTIFICATION

Purpose	Error code for case in which SchM_Call for Nm_StateChangeNotification fails.
Value	243U

5.9.2.1.21. UDPNM_E_TRANSMISSION_OF_PN_SHUTDOWN_MESSAGE_FAILED

Purpose	Error code for case in which a PN synchronize shutdown message is received on an active channel.
----------------	--



Value	0x21
--------------	------

5.9.2.1.22. UDPNM_INSTANCE_ID

Purpose	Instance Id of UdpNm.
Value	0U

5.9.2.1.23. UDPNM_INVALID_PDU_INSTANCE_ID

Purpose	Instance Id of UdpNm when an invalid PDU is passed.
Value	255U

5.9.2.1.24. UDPNM_MODULE_ID

Purpose	AUTOSAR module identification.
Value	33U

5.9.2.1.25. UDPNM_SERVID_CHECKREMOTESLEEPINDICATION

Purpose	AUTOSAR API service ID.
Value	0x11U
Description	Definition of UDPNM_SERVID_CHECKREMOTESLEEPINDICATION.

5.9.2.1.26. UDPNM_SERVID_DISABLECOMMUNICATION

Purpose	AUTOSAR API service ID.
Value	0x0CU
Description	Definition of UDPNM_SERVID_DISABLECOMMUNICATION.

5.9.2.1.27. UDPNM_SERVID_ENABLECOMMUNICATION

Purpose	AUTOSAR API service ID.
----------------	-------------------------



Value	0x0DU
Description	Definition of UDPNM_SERVID_ENABLECOMMUNICATION.

5.9.2.1.28. UDPNM_SERVID_GETLOCALNODEIDENTIFIER

Purpose	AUTOSAR API service ID.
Value	0x07U
Description	Definition of UDPNM_SERVID_GETLOCALNODEIDENTIFIER.

5.9.2.1.29. UDPNM_SERVID_GETNODEIDENTIFIER

Purpose	AUTOSAR API service ID.
Value	0x06U
Description	Definition of UDPNM_SERVID_GETNODEIDENTIFIER.

5.9.2.1.30. UDPNM_SERVID_GETPDUDATA

Purpose	AUTOSAR API service ID.
Value	0x0AU
Description	Definition of UDPNM_SERVID_GETPDUDATA.

5.9.2.1.31. UDPNM_SERVID_GETSTATE

Purpose	AUTOSAR API service ID.
Value	0x0BU
Description	Definition of UDPNM_SERVID_GETSTATE.

5.9.2.1.32. UDPNM_SERVID_GETUSERDATA

Purpose	AUTOSAR API service ID.
Value	0x05U
Description	Definition of UDPNM_SERVID_GETUSERDATA.



5.9.2.1.33. UDPNM_SERVID_GETVERSIONINFO

Purpose	AUTOSAR API service ID.
Value	0x09U
Description	Definition of UDPNM_SERVID_GETVERSIONINFO.

5.9.2.1.34. UDPNM_SERVID_INIT

Purpose	AUTOSAR API service ID.
Value	0x01U
Description	Definition of UDPNM_SERVID_INIT.

5.9.2.1.35. UDPNM_SERVID_MAINFUNCTION_X

Purpose	AUTOSAR API service ID.
Value	0x13U
Description	Definition of UDPNM_SERVID_MAINFUNCTION_X.

5.9.2.1.36. UDPNM_SERVID_NETWORKGWERAREQUEST

Purpose	AUTOSAR API service ID.
Value	0xFEU
Description	Definition of UDPNM_SERVID_NETWORKGWERAREQUEST.

5.9.2.1.37. UDPNM_SERVID_NETWORKRELEASE

Purpose	AUTOSAR API service ID.
Value	0x03U
Description	Definition of UDPNM_SERVID_NETWORKRELEASE.

5.9.2.1.38. UDPNM_SERVID_NETWORKREQUEST

Purpose	AUTOSAR API service ID.
----------------	-------------------------



Value	0x02U
Description	Definition of UDPNM_SERVID_NETWORKREQUEST.

5.9.2.1.39. UDPNM_SERVID_PASSIVESTARTUP

Purpose	AUTOSAR API service ID.
Value	0x0eU
Description	Definition of UDPNM_SERVID_PASSIVESTARTUP

5.9.2.1.40. UDPNM_SERVID_REPEATMESSAGEREQUEST

Purpose	AUTOSAR API service ID.
Value	0x08U
Description	Definition of UDPNM_SERVID_REPEATMESSAGEREQUEST.

5.9.2.1.41. UDPNM_SERVID_REQUESTBUSSYNCHRONIZATION

Purpose	AUTOSAR API service ID.
Value	0x14U
Description	Definition of UDPNM_SERVID_REQUESTBUSSYNCHRONIZATION.

5.9.2.1.42. UDPNM_SERVID_REQUESTSYNCHRONIZEDPNCSHUTDOWN

Purpose	AUTOSAR API service ID.
Value	0xF3U
Description	Definition of UDPNM_SERVID_REQUESTSYNCHRONIZEDPNCSHUTDOWN.

5.9.2.1.43. UDPNM_SERVID_RXINDICATION

Purpose	AUTOSAR API service ID.
Value	0x10U



Description	Definition of UDPNM_SERVID_RXINDICATION.
-------------	--

5.9.2.1.44. UDPNM_SERVID_SETUSERDATA

Purpose	AUTOSAR API service ID.
Value	0x04U
Description	Definition of UDPNM_SERVID_SETUSERDATA.

5.9.2.1.45. UDPNM_SERVID_TXCONFIRMATION

Purpose	AUTOSAR API service ID.
Value	0x0FU
Description	Definition of UDPNM_SERVID_TXCONFIRMATION.

5.9.2.1.46. UDPNM_SW_MAJOR_VERSION

Purpose	AUTOSAR module major version.
Value	2U

5.9.2.1.47. UDPNM_SW_MINOR_VERSION

Purpose	AUTOSAR module minor version.
Value	9U

5.9.2.1.48. UDPNM_SW_PATCH_VERSION

Purpose	AUTOSAR module patch version.
Value	11U

5.9.2.1.49. UDPNM_VENDOR_ID

Purpose	AUTOSAR vendor identification: Elektrobit Automotive GmbH.
Value	1U



5.9.2.2. Functions

5.9.2.2.1. UdpNm_CheckRemoteSleepIndication

Purpose	Check if sleep indication has taken place.	
Synopsis	<code>Std_ReturnType UdpNm_CheckRemoteSleepIndication (NetworkHandleType nmChannelHandle , boolean * nmRemoteSleepIndPtr);</code>	
Service ID	0x011	
Sync/Async	Asynchronous	
Reentrancy	Reentrant (but not for the same NM-Channel)	
Parameters (in)	nmChannelHandle	Identification of the NM-channel.
Parameters (out)	nmRemoteSleepIndPtr	Pointer where check result of remote sleep indication shall be copied to.
Return Value	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Checking of remote sleep indication bits has failed/not executed.
Description	This function checks if remote sleep indication has taken place or not.	

5.9.2.2.2. UdpNm_DisableCommunication

Purpose	Disable NM PDU transmission.	
Synopsis	<code>Std_ReturnType UdpNm_DisableCommunication (NetworkHandleType nmChannelHandle);</code>	
Service ID	0x0c	
Sync/Async	Asynchronous	
Reentrancy	Reentrant (But not for the same NM Channel)	
Parameters (in)	nmChannelHandle	Identification of the NM channel.
Return Value	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Disabling of NM PDU transmission ability has failed/not executed.
Description	This function disables the NM PDU transmission ability due to a ISO14229 Communication Control (28hex) service.	



5.9.2.2.3. UdpNm_EnableCommunication

Purpose	Enable NM PDU transmission.	
Synopsis	<code>Std_ReturnType UdpNm_EnableCommunication (NetworkHandleType nmChannelHandle);</code>	
Service ID	0x0d	
Sync/Async	Asynchronous	
Reentrancy	Reentrant (But not for the same NM Channel)	
Parameters (in)	nmChannelHandle	Identification of the NM channel.
Return Value	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Enabling of NM PDU transmission ability has failed/not executed.
Description	This function enables the NM PDU transmission ability due to a ISO14229 Communication Control (28hex) service.	

5.9.2.2.4. UdpNm_GetLocalNodeIdentifier

Purpose	Get Local Node Identifier.	
Synopsis	<code>Std_ReturnType UdpNm_GetLocalNodeIdentifier (NetworkHandleType nmChannelHandle , uint8 * nmNodeIdPtr);</code>	
Service ID	0x07	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	nmChannelHandle	Identification of the NM channel.
Parameters (out)	nmNodeIdPtr	Pointer where node identifier of the local node shall be copied to.
Return Value	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Getting of the node identifier of the local node has failed.
Description	This function gets the node identifier configured as the local node.	



5.9.2.2.5. UdpNm_GetNodeIdentifier

Purpose	Get Node Identifier.	
Synopsis	<code>Std_ReturnType UdpNm_GetNodeIdentifier (NetworkHandleType nmChannelHandle , uint8 * nmNodeIdPtr);</code>	
Service ID	0x06	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	nmChannelHandle	Identification of the NM channel.
Parameters (out)	nmNodeIdPtr	Pointer where node identifier out of the most recently received NM PDU shall be copied to.
Return Value	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.	
Description	This function gets the node identifier out of the most recently received NM PDU.	

5.9.2.2.6. UdpNm_GetPduData

Purpose	Retrieve the data of the last received NM message.	
Synopsis	<code>Std_ReturnType UdpNm_GetPduData (NetworkHandleType nmChannelHandle , uint8 * nmPduDataPtr);</code>	
Service ID	0x0a	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	nmChannelHandle	Identification of the NM channel.
	nmPduDataPtr	Pointer where NM PDU data shall be copied to.
Return Value	Standard Return Code E_OK No Error.	



	E_NOT_OK	Getting of NM PDU data has failed.
Description	<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"> ► The channel handle should be valid and the module should have been initialized for this channel (checked). 	

5.9.2.2.7. UdpNm_GetState

Purpose	Get the State and mode of the Network Management.	
Synopsis	Std_ReturnType UdpNm_GetState (NetworkHandleType nmChannelHandle , Nm_StateType * nmStatePtr , Nm_ModeType * nmModePtr);	
Service ID	0x0b	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	nmChannelHandle	Identification of the NM-channel.
Parameters (out)	nmStatePtr	Pointer to state of network management.
	nmModePtr	Pointer to mode of network management.
Return Value	Standard Return Code E_OK No Error. E_NOT_OK Getting of NM state has failed.	
Description	This function returns the state and the mode of the network management.	

5.9.2.2.8. UdpNm_GetUserData

Purpose	Get User Data from NM messages.	
Synopsis	Std_ReturnType UdpNm_GetUserData (NetworkHandleType nmChannelHandle , uint8 * nmUserDataPtr);	
Service ID	0x05	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	nmChannelHandle	Identification of the NM channel.



	nmUserDataPtr	Pointer to where user data out of the most recently received NM message shall be copied to.
Return Value	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Getting of user data has failed.
Description	<p>This function retrieves the user data from the last received NM message.</p> <p>Preconditions:</p> <ul style="list-style-type: none"> ➤ The channel handle should be valid and the module should have been initialized for this channel (checked). 	

5.9.2.2.9. UdpNm_GetVersionInfo

Purpose	This service returns the version information of this module.	
Synopsis	<pre>void UdpNm_GetVersionInfo (Std_VersionInfoType * versioninfo) ;</pre>	
Service ID	0x09	
Sync/Async	synchronous	
Reentrancy	reentrant	
Parameters (out)	versioninfo	Pointer to where to store the version information of this module.
Description	This service returns the version information of this module.	

5.9.2.2.10. UdpNm_Init

Purpose	Initialization of UdpNm module.	
Synopsis	<pre>void UdpNm_Init (const UdpNm_ConfigType * udpmConfigPtr) ;</pre>	
Service ID	0x01	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	udpmConfigPtr	Pointer to selected configuration structure.
Description	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

Purpose	Validate configuration.
Synopsis	Std_ReturnType UdpNm_IsValidConfig (const void * voidConfigPtr) ;
Service ID	0x60
Sync/Async	Synchronous
Reentrancy	Reentrant
Return Value	E_OK if the given module configurations is valid otherwise E_NOT_OK.
Description	Checks if the post build configuration fits to the link time configuration part.

5.9.2.2.12. UdpNm_NetworkGwEraRequest

Purpose	Network Gateway Era Request.	
Synopsis	Std_ReturnType UdpNm_NetworkGwEraRequest (NetworkHandleType nmChannelHandle) ;	
Service ID	254	
Sync/Async	Asynchronous	
Reentrancy	Reentrant (But not for the same NM Channel)	
Parameters (in)	nmChannelHandle	Identification of the NM channel.
Return Value	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Requesting of network has failed.
Description	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

Purpose	Release the Network.
Synopsis	Std_ReturnType UdpNm_NetworkRelease (NetworkHandleType nmChannelHandle) ;



Service ID	0x03	
Sync/Async	Asynchronous	
Reentrancy	Reentrant (But not for the same NM Channel)	
Parameters (in)	nmChannelHandle	Identification of the NM channel.
Return Value	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Releasing of network has failed/not executed.
Description	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

Purpose	Network Request.	
Synopsis	Std_ReturnType UdpNm_NetworkRequest (NetworkHandleType nmChannelHandle);	
Service ID	0x02	
Sync/Async	Asynchronous	
Reentrancy	Reentrant (But not for the same NM Channel)	
Parameters (in)	nmChannelHandle	Identification of the NM channel.
Return Value	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Requesting of network has failed.
Description	This function request the network when bus communication is needed. Network state shall be changed to requested.	

5.9.2.2.15. UdpNm_PassiveStartUp

Purpose	Passive startup of UdpNm module.	
Synopsis	Std_ReturnType UdpNm_PassiveStartUp (NetworkHandleType nmChannelHandle);	
Service ID	0x0e	
Sync/Async	Asynchronous	



Reentrancy	Reentrant (But not for the same NM Channel)	
Return Value	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Passive startup of network management has failed/not executed.
Description	<p>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.</p> <p>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.</p>	

5.9.2.2.16. UdpNm_RepeatMessageRequest

Purpose	Set the Repeat Message Request Bit.	
Synopsis	Std_ReturnType UdpNm_RepeatMessageRequest (NetworkHandleType nmChannelHandle);	
Service ID	0x08	
Sync/Async	Asynchronous	
Reentrancy	Reentrant (but not for the same NM Channel)	
Parameters (in)	nmChannelHandle	Identification of the NM channel.
Return Value	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Setting of Repeat Message Request Bit has failed/not executed.
Description	This function sets the Repeat Message Request Bit for NM messages transmitted next on the bus.	

5.9.2.2.17. UdpNm_RequestBusSynchronization

Purpose	Request Bus Syncrhization.	
Synopsis	Std_ReturnType UdpNm_RequestBusSynchronization (NetworkHandleType nmChannelHandle);	
Service ID	0x014	



Sync/Async	synchronous	
Reentrancy	Reentrant (but not for the same NM Channel)	
Parameters (in)	nmChannelHandle	Identification of the NM-channel.
Return Value	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Requesting of bus synchronization has failed/not executed.
Description	This function requests bus synchronization.	

5.9.2.2.18. UdpNm_RequestSynchronizedPncShutdown

Purpose	Request synchronized shutdown.	
Synopsis	<pre>Std_ReturnType UdpNm_RequestSynchronizedPncShutdown (Net- workHandleType nmChannelHandle , PNCHandleType pncId);</pre>	
Service ID	0x4b	
Sync/Async	Synchronous	
Reentrancy	Reentrant (but not for the same NM Channel)	
Parameters (in)	nmChannelHandle	Identification of the NM channel.
	pncId	Identifier of the PNC requested for a synchronized shutdown
Return Value	Standard Return Code	
	E_OK	Request has been accepted.
	E_NOT_OK	Request has not been accepted.
Description	<p>Requests transmission of a NM-PDU with PNSR bit set to 1 (PN shutdown message).</p> <p>Availability:</p> <p>Preconditions:</p> <ul style="list-style-type: none"> ▶ The API is only available if UdpNmSynchronizedPncShutdownEnabled is TRUE for at least one channel. 	

5.9.2.2.19. UdpNm_SetUserData

Purpose	Set User Data for NM messages.
----------------	--------------------------------



Synopsis	<code>Std_ReturnType UdpNm_SetUserData (NetworkHandleType nmChannelHandle , const uint8 * nmUserDataPtr);</code>	
Service ID	0x04	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	nmChannelHandle	Identification of the NM channel.
	nmUserDataPtr	Pointer where the user data for the next transmitted NM message shall be copied from.
Return Value	Standard Return Code	
	E_OK	No Error.
	E_NOT_OK	Setting of user data has failed.
Description	<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"> ➤ The channel handle should be valid and the module should have been initialized for this channel (checked). 	

5.9.2.2.20. `UdpNm_SoAdIfRxIndication`

Purpose	Indicates a received transmission.	
Synopsis	<code>void UdpNm_SoAdIfRxIndication (PduIdType RxPduId , const PduInfoType * PduInfoPtr);</code>	
Parameters (in)	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.
Description	This function indicates the reception of an NM-message PDU.	

5.9.2.2.21. `UdpNm_SoAdIfTxConfirmation`

Purpose	Confirms a transmission.
----------------	--------------------------



Synopsis	<code>void UdpNm_SoAdIfTxConfirmation (PduIdType TxPduId);</code>	
Parameters (in)	TxPduId	Identification of the network through PDU-ID.
Description	<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"> ▶ The UdpNm module is initialized correctly. 	

5.9.2.2.22. UdpNm_Transmit

Purpose	Dummy function.	
Synopsis	<code>Std_ReturnType UdpNm_Transmit (PduIdType UdpNmTxPduId , const PduInfoType * PduInfoPtr);</code>	
Service ID	0x015	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	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.
Return Value	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.
Description	<p>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.</p> <p>Preconditions:</p> <ul style="list-style-type: none"> ▶ 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

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.

`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