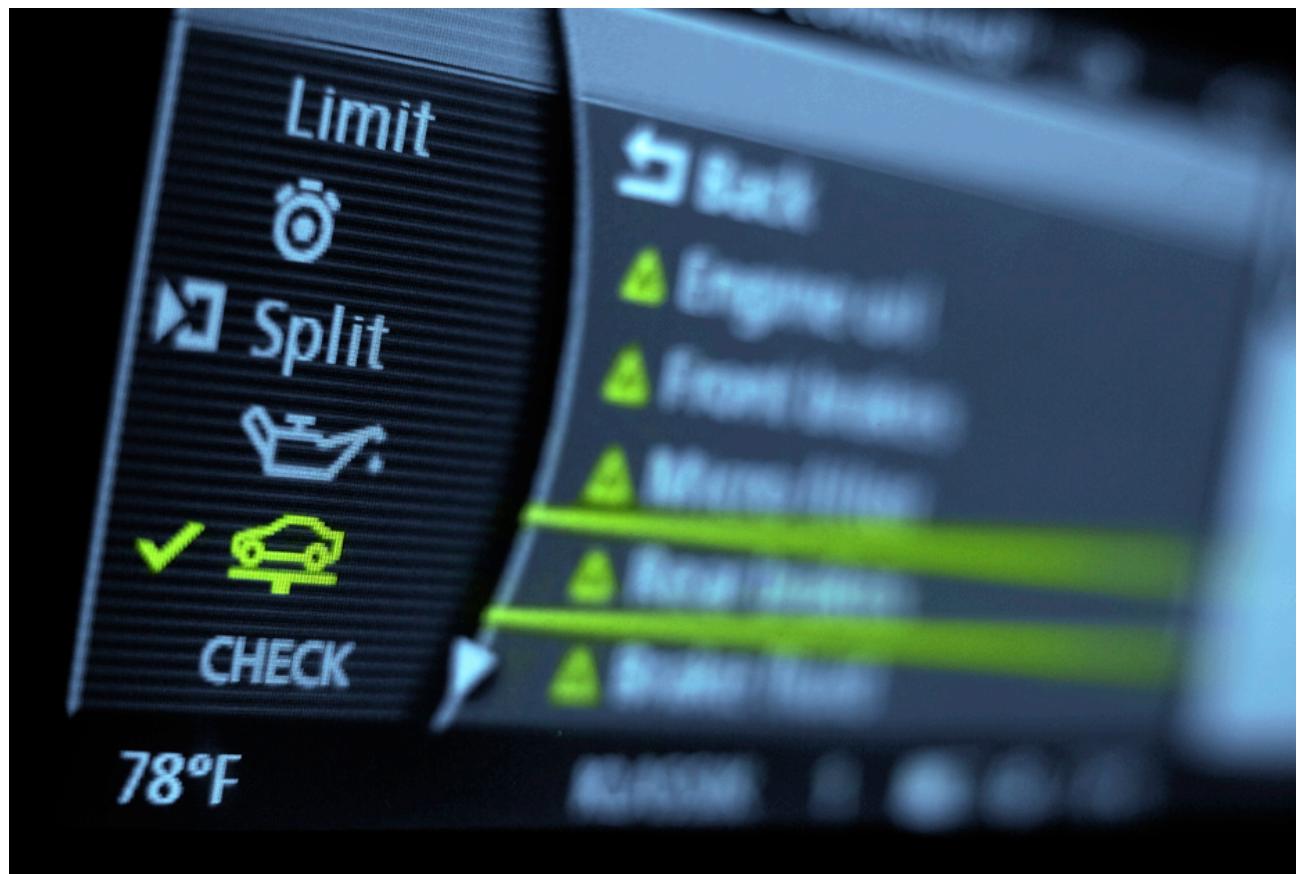




Elektrobit

EB tresos[®] AutoCore Generic 8 Diagnostic Stack documentation

product release 8.8.4





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

Technical support

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

Legal disclaimer

Confidential information.

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

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

Copyright 2021, Elektrobit Automotive GmbH.

Table of Contents

1. Overview of EB tresos AutoCore Generic 8 Diagnostic Stack documentation	47
2. Supported features	48
2.1. Overview	48
2.2. Product details	48
2.2.1. ACG8 Diagnostic Stack	49
2.2.2. ACG8 Diagnostic Stack OBD	50
2.3. Feature details	50
2.3.1. Supported Dcm features	51
2.3.1.1. Dcm feature sets	51
2.3.1.2. UDS services	54
2.3.1.3. OBD services	58
2.3.2. Supported Dem features	59
2.3.3. Supported FiM features	66
3. ACG8 Diagnostic Stack release notes	68
3.1. Overview	68
3.2. Scope of the release	68
3.2.1. Configuration tool	68
3.2.2. AUTOSAR modules	68
3.2.3. EB (Elektrobit) modules	68
3.2.4. MCAL modules and EB tresos AutoCore OS	69
3.3. Module release notes	69
3.3.1. Dcm module release notes	69
3.3.1.1. Change log	69
3.3.1.2. New features	115
3.3.1.3. EB-specific enhancements	121
3.3.1.4. Deviations	133
3.3.1.5. Limitations	180
3.3.1.6. Open-source software	193
3.3.2. Dem module release notes	193
3.3.2.1. Change log	193
3.3.2.2. New features	224
3.3.2.3. EB-specific enhancements	225
3.3.2.4. Deviations	240
3.3.2.5. Limitations	253
3.3.2.6. Open-source software	263
3.3.3. FiM module release notes	263
3.3.3.1. Change log	264
3.3.3.2. New features	271
3.3.3.3. EB-specific enhancements	271



3.3.3.4. Deviations	272
3.3.3.5. Limitations	273
3.3.3.6. Open-source software	275
4. ACG8 Diagnostic Stack user guide	276
4.1. Overview	276
4.2. Background information	276
4.2.1. Purpose of the diagnostic stack	277
4.2.1.1. Diagnostic stack interfaces	277
4.2.1.2. Use case example	278
4.2.2. Error notification	279
4.2.2.1. Development mode	279
4.2.2.2. Production mode	280
4.2.3. Diagnostic Trouble Code (DTC)	280
4.3. Integrating the diagnostic stack	280
4.3.1. Dcm integration issues	281
4.3.2. Dem integration issues	281
4.4. Dcm module user guide	288
4.4.1. Overview	288
4.4.2. Background information	288
4.4.2.1. Diagnostic protocol	289
4.4.2.2. Diagnostic session	290
4.4.2.2.1. Protocol timers	291
4.4.2.2.2. Diagnostic session switch in AUTOSAR 4.0.3 and AUTOSAR 4.0.2	291
4.4.2.3. Security level	292
4.4.2.3.1. AttemptCounter and DelayTimer	293
4.4.2.3.2. Application interfaces for security levels	297
4.4.2.3.2.1. Synchronous function callout security level operations	299
4.4.2.3.2.1.1. Synchronous function callout interface for the GetSeed operation	299
4.4.2.3.2.1.2. Synchronous function callout interface for the CompareKey operation	300
4.4.2.3.2.1.3. Synchronous function callout interface for the GetSecurityAttemptCounter operation	301
4.4.2.3.2.1.4. Synchronous function callout interface for the SetSecurityAttemptCounter operation	302
4.4.2.3.2.2. Asynchronous function callout security level operations	302
4.4.2.3.2.2.1. Asynchronous function callout interface for the GetSeed operation	303
4.4.2.3.2.2.2. Asynchronous function callout interface for the CompareKey operation	305
4.4.2.3.2.2.3. Asynchronous function callout interface for the GetSecurityAttemptCounter operation	307



4.4.2.3.2.2.4. Asynchronous function callout interface for the SetSecurityAttemptCounter operation	308
4.4.2.3.2.3. Synchronous ClientServerInterface operation security level operations	309
4.4.2.3.2.3.1. Synchronous ClientServerInterface operations for the GetSeed operation	310
4.4.2.3.2.3.2. Synchronous ClientServerInterface operations for the CompareKey operation	311
4.4.2.3.2.3.3. Synchronous ClientServerInterface operations for the GetSecurityAttemptCounter operation	312
4.4.2.3.2.3.4. Synchronous ClientServerInterface operations for the SetSecurityAttemptCounter operation	313
4.4.2.3.2.4. Asynchronous ClientServerInterface operation security level operations	313
4.4.2.3.2.4.1. Asynchronous ClientServerInterface operations for the GetSeed operation	315
4.4.2.3.2.4.2. Asynchronous ClientServerInterface operations for the CompareKey operation	317
4.4.2.3.2.4.3. Asynchronous ClientServerInterface operations for the GetSecurityAttemptCounter operation	318
4.4.2.3.2.4.4. Asynchronous ClientServerInterface operations for the SetSecurityAttemptCounter operation	319
4.4.2.4. Bootloader interaction	321
4.4.2.4.1. Notification of the ECU after the reset	321
4.4.2.4.2. Application interfaces for storing the ProgrammingConditions	322
4.4.2.4.3. Bootloader interaction in AUTOSAR 4.0.3 and AUTOSAR 4.0.2	322
4.4.2.5. Diagnostic services	323
4.4.2.5.1. Diagnostic service access	323
4.4.2.5.2. Manufacturer-specific or supplier-specific access verification	324
4.4.2.5.3. Disable diagnostic services while driving	326
4.4.2.5.4. Negative and positive response suppression	326
4.4.2.5.5. Internally managed diagnostic service handlers and subfunction handlers	327
4.4.2.5.6. Asynchronous service processing via separate task	327
4.4.2.5.7. ReadScalingDataByIdentifier (0x24) UDS service	329
4.4.2.5.8. SecurityAccess (0x27) UDS service	330
4.4.2.5.9. CommunicationControl (0x28) UDS service	331
4.4.2.5.10. RoutineControl (0x31) UDS service	332
4.4.2.5.11. Endianness interpretation for DID signals	333
4.4.2.5.12. User-defined functions for unsupported DIDs	336
4.4.2.5.12.1. UDS service ReadDataByIdentifier (0x22)	337
4.4.2.5.12.2. UDS service WriteDataByIdentifier (0x2E)	338



4.4.2.5.12.3. UDS service InputOutputControlByIdentifier (0x2F) with reset-ToDefault	339
4.4.2.5.12.4. UDS service InputOutputControlByIdentifier (0x2F) with return-ControlToEcu	339
4.4.2.5.12.5. UDS service InputOutputControlByIdentifier (0x2F) with freeze-CurrentState	340
4.4.2.5.12.6. UDS service InputOutputControlByIdentifier (0x2F) with short-TermAdjustment	341
4.4.2.5.12.7. Transition to the default session	342
4.4.2.5.13. OBD service Request Vehicle Information (0x09)	342
4.4.2.6. Diagnostic service request handling	343
4.4.2.6.1. Request processing	344
4.4.2.6.2. Dcm buffers	345
4.4.2.6.3. Communication with other basic software modules	346
4.4.2.6.4. Application diagnostic requests	348
4.4.2.6.5. EcuC configuration	349
4.4.2.6.6. Communication modes	349
4.4.2.6.7. Page buffering	350
4.4.2.7. BSW distribution	350
4.4.3. Configuring the Dcm module	351
4.4.3.1. Overview	351
4.4.3.2. Configuring diagnostic sessions	352
4.4.3.2.1. Configuring diagnostic session handling for different AUTOSAR versions	352
4.4.3.2.2. Creating diagnostic sessions	353
4.4.3.2.3. Modifying protocol timings	357
4.4.3.2.4. Configuring bootloader interaction	358
4.4.3.3. Configuring security levels	362
4.4.3.3.1. Configuring security level interfaces for different AUTOSAR versions	362
4.4.3.3.2. Creating security levels	363
4.4.3.3.3. Configuring DelayTimers and AttemptCounters	365
4.4.3.3.3.1. Configuring the Dcm	365
4.4.3.3.3.2. Configuring store and load time-outs for AttemptCounters	366
4.4.3.3.3.3. Configuring infinite DelayTimers upon AttemptCounter retrieval failure	367
4.4.3.3.3.4. Configuring AttemptCounter resetting upon DelayTimer expiration	368
4.4.3.3.3.5. Configuring AttemptCounter incrementation limiting	368
4.4.3.3.3.6. Configuring legacy AttemptCounter handling	369
4.4.3.3.3.7. Configuring a security level DelayTimer	370
4.4.3.3.3.8. Configuring infinite DelayTimers	371
4.4.3.3.3.9. Configuring OnBootDelayTimers	372



4.4.3.3.3.10. Configuring forced OnBootDelayTimers	373
4.4.3.3.3.11. Configuring a security level's AttemptCounter limit	373
4.4.3.3.3.12. Configuring a security level's AttemptCounter to be stored to non-volatile memory	374
4.4.3.4. Configuring diagnostic services	375
4.4.3.4.1. Configuring internally-managed diagnostic services and subfunctions....	375
4.4.3.4.1.1. Configuring internally-managed diagnostic services	375
4.4.3.4.1.2. Configuring internally-managed subfunctions	376
4.4.3.4.2. Configuring externally-managed diagnostic services and subfunctions...	377
4.4.3.4.2.1. Programming externally-managed diagnostic services	377
4.4.3.4.2.2. Configuring externally-managed diagnostic services	379
4.4.3.4.2.3. Programming externally-managed subfunctions	380
4.4.3.4.2.4. Configuring externally-managed subfunctions	382
4.4.3.4.3. Configuring negative response suppression	383
4.4.3.4.4. Configuring access conditions	384
4.4.3.4.4.1. Configuring the availability of diagnostic services in diagnostic sessions	384
4.4.3.4.4.2. Configuring the availability of subfunctions in diagnostic ses- sions	385
4.4.3.4.4.3. Configuring security access verification for diagnostic services...	385
4.4.3.4.4.4. Configuring security access verification for subfunctions	386
4.4.3.4.4.5. Configuring mode rule verification for diagnostic services	387
4.4.3.4.4.6. Configuring mode rule verification for subfunctions	387
4.4.3.4.4.7. Configuring a manufacturer/supplier-specific verification	388
4.4.3.4.4.8. Disabling writing memory diagnostic services while driving	389
4.4.3.4.5. Configuring asynchronous service processing	390
4.4.3.4.6. Configuring the <code>ReadScalingDataByIdentifier (0x24)</code> UDS service	391
4.4.3.4.6.1. Configuring the <code>ReadScalingDataByIdentifier (0x24)</code> UDS service handler	391
4.4.3.4.6.2. Configuring the <code>Xxx_GetScalingInformation</code> operation	392
4.4.3.4.6.3. Configuring <code>DcmDspDataScalingInfoSize</code>	393
4.4.3.4.6.4. Configuring <code>DcmDspDidReadScalingData</code>	393
4.4.3.4.6.4.1. Configuring <code>DcmDspDidReadScalingDataSecu-</code> <code>rityLevelRef</code>	394
4.4.3.4.6.4.2. Configuring <code>DcmDspDidReadScalingDataSession-</code> <code>Ref</code>	394
4.4.3.4.7. Configuring the <code>SecurityAccess (0x27)</code> UDS service	395
4.4.3.4.7.1. Configuring the <code>SecurityAccess (0x27)</code> UDS service han- dler	395
4.4.3.4.7.2. Configuring subfunctions of the <code>SecurityAccess (0x27)</code> UDS service handler	396



4.4.3.4.8. Configuring the CommunicationControl (0x28) UDS service	397
4.4.3.4.8.1. Configuring the CommunicationControl (0x28) UDS service handler	397
4.4.3.4.8.2. Configuring subfunctions of the CommunicationControl (0x28) UDS service	398
4.4.3.4.8.3. Configuring ModeDeclaration support for the CommunicationControl (0x28) UDS service	399
4.4.3.4.8.4. Configuring BswM interaction for the CommunicationControl (0x28) UDS service	400
4.4.3.4.8.5. Configuring a ModeRule to reset CommunicationControls	403
4.4.3.4.8.6. Configuring ComM channels for the CommunicationControl (0x28) service	404
4.4.3.4.8.6.1. Configuring ComM channels for CommunicationControl (0x28) UDS service requests with a subnetNumber between 0x1 and 0xE	405
4.4.3.4.8.6.2. Configuring ComM channels for CommunicationControl (0x28) UDS service requests with a subnetNumber of 0x0Q	406
4.4.3.4.8.6.3. Configuring ComM channels for CommunicationControl (0x28) UDS service requests with enhanced address information	407
4.4.3.4.9. Configuring the RoutineControl (0x31) UDS service	409
4.4.3.4.9.1. Configuring the RoutineControl (0x31) UDS service handler	409
4.4.3.4.9.2. RoutineControl (0x31) UDS service subfunctions	409
4.4.3.4.9.3. Configuring diagnostic routines controlled by the RoutineControl (0x31) UDS service	410
4.4.3.4.9.3.1. Configuring a DcmDspRoutine	410
4.4.3.4.9.3.2. Configuring a DcmDspRoutineInfo	412
4.4.3.4.9.3.3. Configuring DcmDspRoutineVariableLengthInBytes	414
4.4.3.4.9.4. Configuring application-defined interfaces for unsupported DcmDspRoutineIdentifiers	414
4.4.3.4.9.5. Configuring the application-defined interface for inserting the RoutineInfo byte in the diagnostic response as defined in ISO 14229-1/2013	416
4.4.3.4.10. Configuring the Request Vehicle Information (0x09) OBD service	416
4.4.3.4.10.1. Configuring the Request Vehicle Information (0x09) OBD service handler	416
4.4.3.4.10.2. Configuring DcmDspVehInfoASRVersion of Request Vehicle Information (0x09) OBD service	417



4.4.3.4.10.3. Configuring diagnostic vehicle information (<i>Infotype</i>) controlled by the Request Vehicle Information (0x09) OBD service	418
4.4.3.4.10.3.1. Configuring a DcmDspVehInfo	418
4.4.3.4.10.3.1.1. Configuring a DcmDspVehInfoData	418
4.4.3.5. Configuring the diagnostic service request handling	419
4.4.3.5.1. Configuring the diagnostic buffers	420
4.4.3.5.2. Configuring the diagnostic data flow	420
4.4.3.5.3. Configuring the communication mode handling	423
4.4.3.5.4. Enabling diagnostic services within a diagnostic protocol	424
4.4.3.5.5. Enabling diagnostic requests from the application	425
4.4.3.6. Configuring BSW distribution	426
4.4.4. Dcm integration notes	426
4.4.4.1. Generic information	427
4.4.4.2. Protocol preemption	428
4.4.4.3. Endianness interpretation	429
4.4.4.4. BSW distribution	429
4.5. Dem module user guide	429
4.5.1. Overview	429
4.5.2. Background information	429
4.5.2.1. Diagnostic events	430
4.5.2.2. Event processing	430
4.5.2.3. Event-related data (event memory)	432
4.5.2.3.1. Memory types	432
4.5.2.3.2. Event-related data types	433
4.5.2.3.2.1. Data elements and read functions	437
4.5.2.4. Operation cycle	437
4.5.2.4.1. OBD driving cycle (DCY)	438
4.5.2.4.2. OBD warm-up cycle (WUC)	439
4.5.2.5. Event aging	440
4.5.2.6. Debouncing	442
4.5.2.6.1. Calibration	449
4.5.2.6.1.1. Class concept	449
4.5.2.6.1.2. Performing the calibration	450
4.5.2.7. Enable conditions	452
4.5.2.8. Event combination	452
4.5.2.8.1. Event combination type 1 (Combination on storage)	453
4.5.2.9. Multi-event triggering	457
4.5.2.10. Replacement DTC	458
4.5.2.11. Monitor re-initialization	459
4.5.2.12. User-controlled WarningIndicatorRequested-bit	459
4.5.2.13. Support for BSW event reporting from multiple cores	460
4.5.2.14. Variant handling	461



4.5.2.14.1. Calibration	461
4.5.2.14.2. Dynamic event availability	462
4.5.2.14.3. Side allocation	462
4.5.2.14.4. DTC suppression	462
4.5.2.15. Dcm-Dem OBD interface for distance traveled PIDs (\$21/\$31)	463
4.5.2.16. Parallel event memory access	464
4.5.2.17. J1939 support	465
4.5.2.18. Reset of Dem NvM data after an incompatible configuration update	466
4.5.2.19. Considerations for vehicle production	466
4.5.2.19.1. Inhibit the storage of permanent faults during production	467
4.5.2.19.2. Reset the permanent memory at end-of-line	467
4.5.3. Configuring the Dem module	468
4.5.3.1. BSW Mode Manager (BswM)	468
4.5.3.2. Basic Software Scheduler (Rte)	468
4.5.3.3. NVRAM Manager (NvM)	468
4.5.3.4. Enabling BSW event reporting from multiple cores	469
4.5.3.4.1. Configuring the Dem	469
4.5.3.4.2. Configuring the Os	470
4.5.3.4.3. Configuring the Rte	471
4.6. FiM module user guide	472
4.6.1. Overview	472
4.6.2. Background information	473
4.6.2.1. Function Inhibition	473
4.6.2.2. Initialization	473
4.6.3. Configuring the FiM module	474
4.6.3.1. ECU State Manager (EcuM)	474
4.6.3.2. Diagnostic Event Manager (Dem)	474
4.6.3.3. Basic Software Scheduler (Rte)	474
4.6.4. Calibration support for the FiM module	474
4.6.4.1. Calibration parameter	474
5. ACG8 Diagnostic Stack module references	476
5.1. Overview	476
5.1.1. Notation in EB module references	476
5.1.1.1. Default value of configuration parameters	476
5.1.1.2. Range information of configuration parameters	476
5.2. Dcm	477
5.2.1. Configuration parameters	477
5.2.1.1. CommonPublishedInformation	478
5.2.1.2. DcmConfigSet	481
5.2.1.3. DcmDefensiveProgramming	482
5.2.1.4. DcmDsd	485
5.2.1.5. DcmDsdServiceTable	485



5.2.1.6. DcmDsdService	486
5.2.1.7. DcmDsdSubService	491
5.2.1.8. DcmDsl	494
5.2.1.9. DcmDslBuffer	498
5.2.1.10. DcmDslCallbackDCMRequestService	499
5.2.1.11. DcmDslApplicationDiagnosticRequests	499
5.2.1.12. DcmDslDiagResp	500
5.2.1.13. DcmDslProtocol	502
5.2.1.14. DcmDslProtocolRow	502
5.2.1.15. DcmDslConnection	512
5.2.1.16. DcmDslMainConnection	514
5.2.1.17. DcmDslProtocolRx	516
5.2.1.18. DcmDslProtocolTx	519
5.2.1.19. DcmDslPeriodicTransmission	520
5.2.1.20. DcmDslPeriodicConnection	521
5.2.1.21. DcmDslResponseOnEvent	522
5.2.1.22. DcmDslServiceRequestManufacturerNotification	523
5.2.1.23. DcmDslServiceRequestSupplierNotification	524
5.2.1.24. DcmDsp	524
5.2.1.25. DcmDspComControl	534
5.2.1.26. DcmDspComControlAllChannel	534
5.2.1.27. DcmDspComControlSetting	535
5.2.1.28. DcmDspComControlSpecificChannel	536
5.2.1.29. DcmDspComControlSubNode	536
5.2.1.30. DcmDspControlIDTCSetting	537
5.2.1.31. DcmDspData	538
5.2.1.32. DcmDspDataInfo	550
5.2.1.33. DcmDspDid	551
5.2.1.34. DcmDspDidExtRoe	554
5.2.1.35. DcmDspDidSignal	555
5.2.1.36. DcmDspDidInfo	556
5.2.1.37. DcmDspDidAccess	557
5.2.1.38. DcmDspDidControl	558
5.2.1.39. DcmDspDidControlEnableMask	564
5.2.1.40. DcmDspDidRead	564
5.2.1.41. DcmDspDidWrite	565
5.2.1.42. DcmDspDidReadScalingData	567
5.2.1.43. DcmDspDidRange	568
5.2.1.44. DcmDspMemory	572
5.2.1.45. DcmDspAddressAndLengthFormatIdentifier	575
5.2.1.46. DcmDspMemoryIdInfo	576
5.2.1.47. DcmDspReadMemoryRangeInfo	578



5.2.1.48. DcmDspReadMemoryRangeByLabelInfo	580
5.2.1.49. DcmDspWriteMemoryRangeInfo	582
5.2.1.50. DcmDspWriteMemoryRangeByLabelInfo	584
5.2.1.51. DcmDspPeriodicDidTransmission	586
5.2.1.52. DcmDspPeriodicTransmission	588
5.2.1.53. DcmDspPid	589
5.2.1.54. DcmDspPidData	592
5.2.1.55. DcmDspPidDataSupportInfo	593
5.2.1.56. DcmDspPidService01	594
5.2.1.57. DcmDspPidService02	596
5.2.1.58. DcmDspPidSupportInfo	596
5.2.1.59. DcmDspReadGenericInformation	597
5.2.1.60. DcmDspRequestControl	598
5.2.1.61. DcmDspRoe	601
5.2.1.62. DcmDspRoeEvent	602
5.2.1.63. DcmDspRoeEventProperties	604
5.2.1.64. DcmDspRoeOnChangeOfDatalIdentifier	604
5.2.1.65. DcmDspRoeOnDTCStatusChange	604
5.2.1.66. DcmDspRoeEventWindowTime	605
5.2.1.67. DcmDspRoutine	606
5.2.1.68. DcmDspRoutineInfo	613
5.2.1.69. DcmDspRoutineAuthorization	614
5.2.1.70. DcmDspRoutineRequestResIn	616
5.2.1.71. DcmDspRoutineRequestResInSignal	616
5.2.1.72. DcmDspRoutineRequestResOut	618
5.2.1.73. DcmDspRoutineRequestResOutSignal	618
5.2.1.74. DcmDspRoutineStopIn	621
5.2.1.75. DcmDspRoutineStopInSignal	621
5.2.1.76. DcmDspRoutineStopOut	624
5.2.1.77. DcmDspRoutineStopOutSignal	624
5.2.1.78. DcmDspStartRoutineIn	627
5.2.1.79. DcmDspStartRoutineInSignal	627
5.2.1.80. DcmDspStartRoutineOut	630
5.2.1.81. DcmDspStartRoutineOutSignal	630
5.2.1.82. DcmDspSecurity	633
5.2.1.83. DcmDspSecurityRow	637
5.2.1.84. DcmDspSession	646
5.2.1.85. DcmDspSessionRow	647
5.2.1.86. DcmDspTestResultByObdmid	650
5.2.1.87. DcmDspTestResultObdmidTid	652
5.2.1.88. DcmDspTestResultObdmidTids	653
5.2.1.89. DcmDspTestResultObdmidTidRefs	654



5.2.1.90. DcmDspTestResultTid	654
5.2.1.91. DcmDspVehInfo	655
5.2.1.92. DcmDspVehInfoData	656
5.2.1.93. DcmGeneral	659
5.2.1.94. DcmServiceAPI	673
5.2.1.95. DcmPageBufferCfg	674
5.2.1.96. DcmProcessingConditions	675
5.2.1.97. DcmModeCondition	675
5.2.1.98. DcmBswModeRef	676
5.2.1.99. DcmSwcModeRef	677
5.2.1.100. DcmModeRule	678
5.2.1.101. PublishedInformation	679
5.2.2. Recommended configurations	680
5.2.2.1. DcmRecConfigurationBasic	680
5.2.2.1.1. DcmConfigSet_0	680
5.2.2.1.2. DcmDsd	680
5.2.2.1.3. DcmDsdServiceTable_0	681
5.2.2.1.4. TesterPresent	681
5.2.2.1.5. DiagnosticSessionControl	682
5.2.2.1.6. defaultSession	682
5.2.2.1.7. programmingSession	682
5.2.2.1.8. extendedDiagnosticSession	682
5.2.2.1.9. EcuReset	683
5.2.2.1.10. hardReset	683
5.2.2.1.11. keyOffOnReset	683
5.2.2.1.12. softReset	684
5.2.2.1.13. enableRapidPowerShutDown	684
5.2.2.1.14. disableRapidPowerShutDown	684
5.2.2.1.15. SecurityAccess	684
5.2.2.1.16. ReadDid	685
5.2.2.1.17. WriteDid	685
5.2.2.1.18. ReadDTCInformation	685
5.2.2.1.19. reportNumberOfDTCByStatusMask	686
5.2.2.1.20. reportDTCByStatusMask	686
5.2.2.1.21. reportDTCSnapshotRecordIdentification	686
5.2.2.1.22. reportDTCSnapshotRecordByDTCNumber	686
5.2.2.1.23. reportDTCExtendedDataRecordByDTCNumber	687
5.2.2.1.24. reportSupportedDTCs	687
5.2.2.1.25. ClearDiagnosticInformation	687
5.2.2.1.26. CommunicationControl	687
5.2.2.1.27. enableRxAndTx	688
5.2.2.1.28. enableRxAndDisableTx	688



5.2.2.1.29. disableRxAndEnableTx	688
5.2.2.1.30. disableRxAndTx	689
5.2.2.1.31. ControlDTCSetting	689
5.2.2.1.32. DcmDsdSubService_0	689
5.2.2.1.33. DcmDsdSubService_1	689
5.2.2.1.34. DcmDsp	690
5.2.2.1.35. DcmDspComControl	691
5.2.2.1.36. DcmDspComControlSetting	691
5.2.2.1.37. DcmDspControlDTCSetting	691
5.2.2.1.38. DcmDspPeriodicDidTransmission	691
5.2.2.1.39. DcmDspPeriodicTransmission	692
5.2.2.1.40. DcmDspReadGenericInformation	692
5.2.2.1.41. DcmDspSession	692
5.2.2.1.42. DCM_DEFAULT_SESSION	693
5.2.2.1.43. DCM_EXTENDED_DIAGNOSTIC_SESSION	693
5.2.2.1.44. DCM_PROGRAMMING_SESSION	693
5.2.2.1.45. DCM_ALL_SESSION_LEVEL	693
5.2.2.1.46. DcmDspData_0x100	694
5.2.2.1.47. DcmDspData_0xF186	694
5.2.2.1.48. DcmDspDataInfo_0x100	695
5.2.2.1.49. DcmDspDataInfo_0xF186	695
5.2.2.1.50. DcmDspDid_0x100	695
5.2.2.1.51. DcmDspDidSignal_0	696
5.2.2.1.52. DcmDspDid_0xF186	696
5.2.2.1.53. DcmDspDidSignal_0XF186	696
5.2.2.1.54. DcmDspDidInfo_0x100	696
5.2.2.1.55. DcmDspDidAccess	697
5.2.2.1.56. DcmDspDidControl	697
5.2.2.1.57. DcmDspDidRead	697
5.2.2.1.58. DcmDspDidWrite	698
5.2.2.1.59. DcmDspDidInfo_0xF186	698
5.2.2.1.60. DcmDspDidAccess	698
5.2.2.1.61. DcmDspDidControl	698
5.2.2.1.62. DcmDspDidRead	699
5.2.2.1.63. DcmDspDidWrite	699
5.2.2.1.64. DcmDspRoe	699
5.2.2.1.65. DcmDspRoeEvent_0	699
5.2.2.1.66. DcmDspRoeEventProperties	699
5.2.2.1.67. DcmDspRoeOnChangeOfDataIdentifier	700
5.2.2.1.68. DcmDspRoeOnDTCStatusChange	700
5.2.2.1.69. DcmDspRoeEventWindowTime_0	700
5.2.2.1.70. DcmDspRoutine_0	700



5.2.2.1.71. DcmDspRoutineInfo_0	701
5.2.2.1.72. DcmDspRoutineAuthorization	701
5.2.2.1.73. DcmDspRoutineRequestResIn	701
5.2.2.1.74. DcmDspRoutineRequestResOut	702
5.2.2.1.75. DcmDspRoutineRequestResOutSignal_0	702
5.2.2.1.76. DcmDspRoutineRequestResOutSignal_1	702
5.2.2.1.77. DcmDspRoutineStopIn	702
5.2.2.1.78. DcmDspRoutineStopInSignal_0	703
5.2.2.1.79. DcmDspRoutineStopInSignal_1	703
5.2.2.1.80. DcmDspRoutineStopOut	703
5.2.2.1.81. DcmDspRoutineStopOutSignal_0	703
5.2.2.1.82. DcmDspRoutineStopOutSignal_1	704
5.2.2.1.83. DcmDspStartRoutineIn	704
5.2.2.1.84. DcmDspStartRoutineInSignal_0	704
5.2.2.1.85. DcmDspStartRoutineInSignal_1	704
5.2.2.1.86. DcmDspStartRoutineOut	705
5.2.2.1.87. DcmDspStartRoutineOutSignal_0	705
5.2.2.1.88. DcmDspStartRoutineOutSignal_1	705
5.2.2.1.89. DcmDspSecurity	705
5.2.2.1.90. DCM_SEC_LEV_L1	706
5.2.2.1.91. DCM_SEC_LEV_L3	706
5.2.2.1.92. DcmDspTestResultByObdmid	707
5.2.2.1.93. DcmDspMemory	707
5.2.2.1.94. DcmDspAddressAndLengthFormatIdentifier	707
5.2.2.1.95. DcmDspMemoryIdInfo_0	707
5.2.2.1.96. DcmDspReadMemoryRangeInfo_0	708
5.2.2.1.97. DcmDspWriteMemoryRangeInfo_0	708
5.2.2.1.98. DcmDsl	708
5.2.2.1.99. DcmDslBuffer_Common	708
5.2.2.1.100. DcmDslDiagResp	709
5.2.2.1.101. DcmDslProtocol	709
5.2.2.1.102. DcmDslProtocolRow_0	709
5.2.2.1.103. DcmDslConnection_0	710
5.2.2.1.104. DcmDslMainConnection	710
5.2.2.1.105. DcmDslProtocolRx_0	710
5.2.2.1.106. DcmDslProtocolRx_1	711
5.2.2.1.107. DcmDslProtocolTx	711
5.2.2.1.108. DcmDslPeriodicTransmission	711
5.2.2.1.109. DcmDslResponseOnEvent	711
5.2.2.1.110. DcmDefensiveProgramming	711
5.2.2.1.111. DcmGeneral	712
5.2.2.1.112. DcmServiceAPI	712



5.2.2.1.113. DcmPageBufferCfg	713
5.2.2.1.114. DcmProcessingConditions	713
5.2.2.2. DcmRecConfigurationDflt	713
5.2.2.2.1. DcmConfigSet_0	713
5.2.2.2.2. DcmDsd	714
5.2.2.2.3. DcmDsdServiceTable_0	714
5.2.2.2.4. TesterPresent	715
5.2.2.2.5. DiagnosticSessionControl	715
5.2.2.2.6. defaultSession	716
5.2.2.2.7. programmingSession	716
5.2.2.2.8. extendedDiagnosticSession	716
5.2.2.2.9. oem_BI_ProgrammingSession	716
5.2.2.2.10. sys_BI_ProgrammingSession	717
5.2.2.2.11. EcuReset	717
5.2.2.2.12. hardReset	717
5.2.2.2.13. keyOffOnReset	718
5.2.2.2.14. softReset	718
5.2.2.2.15. enableRapidPowerShutDown	718
5.2.2.2.16. disableRapidPowerShutDown	718
5.2.2.2.17. SecurityAccess	718
5.2.2.2.18. ReadDid	719
5.2.2.2.19. WriteDid	719
5.2.2.2.20. ReadDTCInformation	719
5.2.2.2.21. reportNumberOfDTCByStatusMask	720
5.2.2.2.22. reportDTCByStatusMask	720
5.2.2.2.23. reportDTCSnapshotRecordIdentification	720
5.2.2.2.24. reportDTCSnapshotRecordByDTCNumber	721
5.2.2.2.25. reportDTCExtendedDataRecordByDTCNumber	721
5.2.2.2.26. reportSupportedDTCs	721
5.2.2.2.27. ClearDiagnosticInformation	721
5.2.2.2.28. CommunicationControl	722
5.2.2.2.29. enableRxAndTx	722
5.2.2.2.30. enableRxAndDisableTx	722
5.2.2.2.31. disableRxAndEnableTx	723
5.2.2.2.32. disableRxAndTx	723
5.2.2.2.33. RoutineControl	723
5.2.2.2.34. RequestDownload	723
5.2.2.2.35. RequestUpload	724
5.2.2.2.36. TransferData	724
5.2.2.2.37. RequestTransferExit	724
5.2.2.2.38. LinkControl	725
5.2.2.2.39. verifyBaudrateTransitionWithFixedBaudrate	725



5.2.2.2.40. verifyBaudrateTransitionWithSpecificBaudrate	725
5.2.2.2.41. verifyBaudrateTransitionWithTransitionBaudrate	726
5.2.2.2.42. ControlDTCSetting	726
5.2.2.2.43. DcmDsdSubService_0	726
5.2.2.2.44. DcmDsdSubService_1	726
5.2.2.2.45. ReadMemoryByAddress	727
5.2.2.2.46. WriteMemoryByAddress	727
5.2.2.2.47. InputOutputControlByIdentifier	727
5.2.2.2.48. ResponseOnEvent	728
5.2.2.2.49. stopResponseOnEvent	728
5.2.2.2.50. onDTCStatusChange	728
5.2.2.2.51. onChangeOfDataIdentifier	729
5.2.2.2.52. reportActivatedEvents	729
5.2.2.2.53. startResponseOnEvent	729
5.2.2.2.54. clearResponseOnEvent	729
5.2.2.2.55. DcmDsp	730
5.2.2.2.56. DcmDspComControl	731
5.2.2.2.57. DcmDspComControlSetting	731
5.2.2.2.58. DcmDspControlDTCSetting	731
5.2.2.2.59. DcmDspPeriodicDidTransmission	731
5.2.2.2.60. DcmDspPeriodicTransmission	732
5.2.2.2.61. DcmDspReadGenericInformation	732
5.2.2.2.62. DcmDspSession	732
5.2.2.2.63. DCM_DEFAULT_SESSION	732
5.2.2.2.64. DCM_EXTENDED_DIAGNOSTIC_SESSION	733
5.2.2.2.65. DCM_PROGRAMMING_SESSION	733
5.2.2.2.66. DCM_ALL_SESSION_LEVEL	733
5.2.2.2.67. DcmDspSessionRow_0	734
5.2.2.2.68. DcmDspSessionRow_1	734
5.2.2.2.69. DcmDspData_0x100	734
5.2.2.2.70. DcmDspData_0xF186	735
5.2.2.2.71. DcmDspDataInfo_0x100	735
5.2.2.2.72. DcmDspDataInfo_0xF186	736
5.2.2.2.73. DcmDspDid_0x100	736
5.2.2.2.74. DcmDspDidSignal_0	736
5.2.2.2.75. DcmDspDid_0xF186	736
5.2.2.2.76. DcmDspDidSignal_0XF186	737
5.2.2.2.77. DcmDspDidInfo_0x100	737
5.2.2.2.78. DcmDspDidAccess	737
5.2.2.2.79. DcmDspDidControl	737
5.2.2.2.80. DcmDspDidRead	738
5.2.2.2.81. DcmDspDidWrite	738



5.2.2.2.82. DcmDspDidInfo_0xF186	738
5.2.2.2.83. DcmDspDidAccess	738
5.2.2.2.84. DcmDspDidControl	739
5.2.2.2.85. DcmDspDidRead	739
5.2.2.2.86. DcmDspDidWrite	739
5.2.2.2.87. DcmDspRoe	739
5.2.2.2.88. DcmDspRoeEvent_0	740
5.2.2.2.89. DcmDspRoeEventProperties	740
5.2.2.2.90. DcmDspRoeOnChangeOfDataIdentifier	740
5.2.2.2.91. DcmDspRoeOnDTCStatusChange	740
5.2.2.2.92. DcmDspRoeEventWindowTime_0	740
5.2.2.2.93. DcmDspRoutine_0	741
5.2.2.2.94. DcmDspRoutineInfo_0	741
5.2.2.2.95. DcmDspRoutineAuthorization	742
5.2.2.2.96. DcmDspRoutineRequestResIn	742
5.2.2.2.97. DcmDspRoutineRequestResOut	742
5.2.2.2.98. DcmDspRoutineRequestResOutSignal_0	742
5.2.2.2.99. DcmDspRoutineRequestResOutSignal_1	742
5.2.2.2.100. DcmDspRoutineStopIn	743
5.2.2.2.101. DcmDspRoutineStopInSignal_0	743
5.2.2.2.102. DcmDspRoutineStopInSignal_1	743
5.2.2.2.103. DcmDspRoutineStopOut	743
5.2.2.2.104. DcmDspRoutineStopOutSignal_0	744
5.2.2.2.105. DcmDspRoutineStopOutSignal_1	744
5.2.2.2.106. DcmDspStartRoutineIn	744
5.2.2.2.107. DcmDspStartRoutineInSignal_0	745
5.2.2.2.108. DcmDspStartRoutineInSignal_1	745
5.2.2.2.109. DcmDspStartRoutineOut	745
5.2.2.2.110. DcmDspStartRoutineOutSignal_0	745
5.2.2.2.111. DcmDspStartRoutineOutSignal_1	746
5.2.2.2.112. DcmDspSecurity	746
5.2.2.2.113. DCM_SEC_LEV_L1	746
5.2.2.2.114. DCM_SEC_LEV_L3	747
5.2.2.2.115. DcmDspTestResultByObdmid	747
5.2.2.2.116. DcmDspMemory	747
5.2.2.2.117. DcmDspAddressAndLengthFormatIdentifier	748
5.2.2.2.118. DcmDspMemoryIdInfo_0	748
5.2.2.2.119. DcmDspReadMemoryRangeInfo_0	748
5.2.2.2.120. DcmDspWriteMemoryRangeInfo_0	748
5.2.2.2.121. DcmDsl	749
5.2.2.2.122. DcmDslBuffer_Common	749
5.2.2.2.123. DcmDslBuffer_Rx	749



5.2.2.2.124. DcmDslBuffer_Tx	749
5.2.2.2.125. DcmDslDiagResp	750
5.2.2.2.126. DcmDslProtocol	750
5.2.2.2.127. DcmDslProtocolRow_0	750
5.2.2.2.128. DcmDslConnection_0	751
5.2.2.2.129. DcmDslMainConnection	751
5.2.2.2.130. DcmDslProtocolRx_0	751
5.2.2.2.131. DcmDslProtocolRx_1	752
5.2.2.2.132. DcmDslProtocolTx	752
5.2.2.2.133. DcmDslPeriodicTransmission	752
5.2.2.2.134. DcmDslResponseOnEvent	752
5.2.2.2.135. DcmDslConnection_1	752
5.2.2.2.136. DcmDslMainConnection	753
5.2.2.2.137. DcmDslProtocolTx	753
5.2.2.2.138. DcmDslPeriodicTransmission	753
5.2.2.2.139. DcmDslPeriodicConnection	753
5.2.2.2.140. DcmDslResponseOnEvent	754
5.2.2.2.141. DcmDslConnection_2	754
5.2.2.2.142. DcmDslMainConnection	754
5.2.2.2.143. DcmDslProtocolTx	754
5.2.2.2.144. DcmDslPeriodicTransmission	755
5.2.2.2.145. DcmDslResponseOnEvent	755
5.2.2.2.146. DcmDslProtocolRow_1	755
5.2.2.2.147. DcmDslConnection_3	756
5.2.2.2.148. DcmDslMainConnection	756
5.2.2.2.149. DcmDslProtocolRx_0	756
5.2.2.2.150. DcmDslProtocolRx_1	756
5.2.2.2.151. DcmDslProtocolTx	757
5.2.2.2.152. DcmDslPeriodicTransmission	757
5.2.2.2.153. DcmDslResponseOnEvent	757
5.2.2.2.154. DcmDslConnection_4	757
5.2.2.2.155. DcmDslMainConnection	758
5.2.2.2.156. DcmDslProtocolTx	758
5.2.2.2.157. DcmDslPeriodicTransmission	758
5.2.2.2.158. DcmDslPeriodicConnection_0	758
5.2.2.2.159. DcmDslResponseOnEvent	758
5.2.2.2.160. DcmDslConnection_5	759
5.2.2.2.161. DcmDslMainConnection	759
5.2.2.2.162. DcmDslProtocolTx	759
5.2.2.2.163. DcmDslPeriodicTransmission	759
5.2.2.2.164. DcmDslResponseOnEvent	760
5.2.2.2.165. DcmDefensiveProgramming	760



5.2.2.2.166. DcmGeneral	760
5.2.2.2.167. DcmServiceAPI	761
5.2.2.2.168. DcmPageBufferCfg	761
5.2.2.2.169. DcmProcessingConditions	761
5.2.2.3. DcmRecConfigurationWithRte	761
5.2.2.3.1. DcmConfigSet_0	762
5.2.2.3.2. DcmDsd	762
5.2.2.3.3. DcmDsdServiceTable_0	762
5.2.2.3.4. TesterPresent	763
5.2.2.3.5. DiagnosticSessionControl	764
5.2.2.3.6. defaultSession	764
5.2.2.3.7. programmingSession	764
5.2.2.3.8. extendedDiagnosticSession	765
5.2.2.3.9. oem_BI_ProgrammingSession	765
5.2.2.3.10. sys_BI_ProgrammingSession	765
5.2.2.3.11. EcuReset	765
5.2.2.3.12. hardReset	766
5.2.2.3.13. keyOffOnReset	766
5.2.2.3.14. softReset	766
5.2.2.3.15. enableRapidPowerShutDown	767
5.2.2.3.16. disableRapidPowerShutDown	767
5.2.2.3.17. SecurityAccess	767
5.2.2.3.18. ReadDid	767
5.2.2.3.19. WriteDid	768
5.2.2.3.20. ReadDTCInformation	768
5.2.2.3.21. reportNumberOfDTCByStatusMask	769
5.2.2.3.22. reportDTCByStatusMask	769
5.2.2.3.23. reportDTCSnapshotRecordIdentification	769
5.2.2.3.24. reportDTCSnapshotRecordByDTCNumber	769
5.2.2.3.25. reportDTCExtendedDataRecordByDTCNumber	769
5.2.2.3.26. reportSupportedDTCs	770
5.2.2.3.27. ClearDiagnosticInformation	770
5.2.2.3.28. CommunicationControl	770
5.2.2.3.29. enableRxAndTx	771
5.2.2.3.30. enableRxAndDisableTx	771
5.2.2.3.31. disableRxAndEnableTx	771
5.2.2.3.32. disableRxAndTx	771
5.2.2.3.33. RoutineControl	772
5.2.2.3.34. RequestDownload	772
5.2.2.3.35. RequestUpload	772
5.2.2.3.36. TransferData	773
5.2.2.3.37. RequestTransferExit	773



5.2.2.3.38. LinkControl	773
5.2.2.3.39. verifyBaudrateTransitionWithFixedBaudrate	774
5.2.2.3.40. verifyBaudrateTransitionWithSpecificBaudrate	774
5.2.2.3.41. verifyBaudrateTransitionWithTransitionBaudrate	774
5.2.2.3.42. ControlDTCSetting	774
5.2.2.3.43. DcmDsdSubService_0	775
5.2.2.3.44. DcmDsdSubService_1	775
5.2.2.3.45. ReadMemoryByAddress	775
5.2.2.3.46. WriteMemoryByAddress	776
5.2.2.3.47. InputOutputControlByIdentifier	776
5.2.2.3.48. ResponseOnEvent	776
5.2.2.3.49. stopResponseOnEvent	777
5.2.2.3.50. onDTCStatusChange	777
5.2.2.3.51. onChangeOfDataIdentifier	777
5.2.2.3.52. reportActivatedEvents	777
5.2.2.3.53. startResponseOnEvent	778
5.2.2.3.54. clearResponseOnEvent	778
5.2.2.3.55. DcmDsp	778
5.2.2.3.56. DcmDspComControl	779
5.2.2.3.57. DcmDspComControlSetting	779
5.2.2.3.58. DcmDspControlDTCSetting	780
5.2.2.3.59. DcmDspPeriodicDidTransmission	780
5.2.2.3.60. DcmDspPeriodicTransmission	780
5.2.2.3.61. DcmDspReadGenericInformation	780
5.2.2.3.62. DcmDspSession	780
5.2.2.3.63. DCM_DEFAULT_SESSION	781
5.2.2.3.64. DCM_EXTENDED_DIAGNOSTIC_SESSION	781
5.2.2.3.65. DCM_PROGRAMMING_SESSION	781
5.2.2.3.66. DCM_ALL_SESSION_LEVEL	782
5.2.2.3.67. DcmDspSessionRow_0	782
5.2.2.3.68. DcmDspSessionRow_1	782
5.2.2.3.69. DcmDspData_0x100	783
5.2.2.3.70. DcmDspData_0xF186	783
5.2.2.3.71. DcmDspDataInfo_0x100	784
5.2.2.3.72. DcmDspDataInfo_0xF186	784
5.2.2.3.73. DcmDspDidRange	784
5.2.2.3.74. DcmDspDid_0x100	785
5.2.2.3.75. DcmDspDidSignal_0	785
5.2.2.3.76. DcmDspDid_0xF186	785
5.2.2.3.77. DcmDspDidSignal_0XF186	785
5.2.2.3.78. DcmDspDidInfo_0x100	786
5.2.2.3.79. DcmDspDidAccess	786



5.2.2.3.80. DcmDspDidControl	786
5.2.2.3.81. DcmDspDidRead	787
5.2.2.3.82. DcmDspDidWrite	787
5.2.2.3.83. DcmDspDidInfo_0xF186	787
5.2.2.3.84. DcmDspDidAccess	787
5.2.2.3.85. DcmDspDidControl	788
5.2.2.3.86. DcmDspDidRead	788
5.2.2.3.87. DcmDspDidWrite	788
5.2.2.3.88. DcmDspRoe	788
5.2.2.3.89. DcmDspRoeEvent_0	788
5.2.2.3.90. DcmDspRoeEventProperties	789
5.2.2.3.91. DcmDspRoeOnChangeOfDataIdentifier	789
5.2.2.3.92. DcmDspRoeOnDTCStatusChange	789
5.2.2.3.93. DcmDspRoeEventWindowTime_0	789
5.2.2.3.94. DcmDspRoutine_0	790
5.2.2.3.95. DcmDspRoutineInfo_0	790
5.2.2.3.96. DcmDspRoutineAuthorization	790
5.2.2.3.97. DcmDspRoutineRequestResIn	791
5.2.2.3.98. DcmDspRoutineRequestResOut	791
5.2.2.3.99. DcmDspRoutineRequestResOutSignal_0	791
5.2.2.3.100. DcmDspRoutineRequestResOutSignal_1	791
5.2.2.3.101. DcmDspRoutineStopIn	791
5.2.2.3.102. DcmDspRoutineStopInSignal_0	792
5.2.2.3.103. DcmDspRoutineStopInSignal_1	792
5.2.2.3.104. DcmDspRoutineStopOut	792
5.2.2.3.105. DcmDspRoutineStopOutSignal_0	793
5.2.2.3.106. DcmDspRoutineStopOutSignal_1	793
5.2.2.3.107. DcmDspStartRoutineIn	793
5.2.2.3.108. DcmDspStartRoutineInSignal_0	793
5.2.2.3.109. DcmDspStartRoutineInSignal_1	794
5.2.2.3.110. DcmDspStartRoutineOut	794
5.2.2.3.111. DcmDspStartRoutineOutSignal_0	794
5.2.2.3.112. DcmDspStartRoutineOutSignal_1	794
5.2.2.3.113. DcmDspSecurity	795
5.2.2.3.114. DCM_SEC_LEV_L1	795
5.2.2.3.115. DCM_SEC_LEV_L3	795
5.2.2.3.116. DcmDspTestResultByObdmid	796
5.2.2.3.117. DcmDspMemory	796
5.2.2.3.118. DcmDspAddressAndLengthFormatIdentifier	796
5.2.2.3.119. DcmDspMemoryIdInfo_0	796
5.2.2.3.120. DcmDspReadMemoryRangeInfo_0	797
5.2.2.3.121. DcmDspWriteMemoryRangeInfo_0	797



5.2.2.3.122. DcmDsl	797
5.2.2.3.123. DcmDslBuffer_Common	798
5.2.2.3.124. DcmDslBuffer_Rx	798
5.2.2.3.125. DcmDslBuffer_Tx	798
5.2.2.3.126. DcmDslCallbackDCMRequestService_0	798
5.2.2.3.127. DcmDslDiagResp	798
5.2.2.3.128. DcmDslProtocol	799
5.2.2.3.129. DcmDslProtocolRow_0	799
5.2.2.3.130. DcmDslConnection_0	800
5.2.2.3.131. DcmDslMainConnection	800
5.2.2.3.132. DcmDslProtocolRx_0	800
5.2.2.3.133. DcmDslProtocolRx_1	801
5.2.2.3.134. DcmDslProtocolTx	801
5.2.2.3.135. DcmDslPeriodicTransmission	801
5.2.2.3.136. DcmDslResponseOnEvent	801
5.2.2.3.137. DcmDslConnection_1	801
5.2.2.3.138. DcmDslMainConnection	802
5.2.2.3.139. DcmDslProtocolTx	802
5.2.2.3.140. DcmDslPeriodicTransmission	802
5.2.2.3.141. DcmDslPeriodicConnection	802
5.2.2.3.142. DcmDslResponseOnEvent	803
5.2.2.3.143. DcmDslConnection_2	803
5.2.2.3.144. DcmDslMainConnection	803
5.2.2.3.145. DcmDslProtocolTx	803
5.2.2.3.146. DcmDslPeriodicTransmission	804
5.2.2.3.147. DcmDslResponseOnEvent	804
5.2.2.3.148. DcmDslProtocolRow_1	804
5.2.2.3.149. DcmDslConnection_3	805
5.2.2.3.150. DcmDslMainConnection	805
5.2.2.3.151. DcmDslProtocolRx_0	805
5.2.2.3.152. DcmDslProtocolRx_1	806
5.2.2.3.153. DcmDslProtocolTx	806
5.2.2.3.154. DcmDslPeriodicTransmission	806
5.2.2.3.155. DcmDslResponseOnEvent	806
5.2.2.3.156. DcmDslConnection_4	806
5.2.2.3.157. DcmDslMainConnection	807
5.2.2.3.158. DcmDslProtocolTx	807
5.2.2.3.159. DcmDslPeriodicTransmission	807
5.2.2.3.160. DcmDslPeriodicConnection_0	807
5.2.2.3.161. DcmDslResponseOnEvent	808
5.2.2.3.162. DcmDslConnection_5	808
5.2.2.3.163. DcmDslMainConnection	808



5.2.2.3.164. DcmDslProtocolTx	808
5.2.2.3.165. DcmDslPeriodicTransmission	809
5.2.2.3.166. DcmDslResponseOnEvent	809
5.2.2.3.167. DcmDslServiceRequestManufacturerNotification_0	809
5.2.2.3.168. DcmDslServiceRequestSupplierNotification_0	809
5.2.2.3.169. DcmDefensiveProgramming	809
5.2.2.3.170. DcmGeneral	810
5.2.2.3.171. DcmServiceAPI	810
5.2.2.3.172. DcmPageBufferCfg	811
5.2.2.3.173. DcmProcessingConditions	811
5.2.2.3.174. DcmModeCondition_0	811
5.2.2.3.175. DcmBswModeRef	811
5.2.2.3.176. DcmSwcModeRef	812
5.2.2.3.177. DcmModeCondition_1	812
5.2.2.3.178. DcmBswModeRef	812
5.2.2.3.179. DcmSwcModeRef	812
5.2.2.3.180. DcmModeRule_0	812
5.2.2.3.181. DcmModeRule_1	813
5.2.3. Application programming interface (API)	813
5.2.3.1. Type definitions	813
5.2.3.1.1. Dcm_CommunicationModeType	813
5.2.3.1.2. Dcm_ConfirmationStatusType	813
5.2.3.1.3. Dcm_DIDIDType	813
5.2.3.1.4. Dcm_DTRStatusType	813
5.2.3.1.5. Dcm_DidSupportedType	814
5.2.3.1.6. Dcm_EcuStartModeType	814
5.2.3.1.7. Dcm_IdContextType	814
5.2.3.1.8. Dcm_LinkControlRequestType	814
5.2.3.1.9. Dcm_MsgAddInfoType	814
5.2.3.1.10. Dcm_MsgContextPtrType	815
5.2.3.1.11. Dcm_MsgContextType	815
5.2.3.1.12. Dcm_MsgItemType	816
5.2.3.1.13. Dcm_MsgLenType	816
5.2.3.1.14. Dcm_MsgType	816
5.2.3.1.15. Dcm_NegativeResponseCodeType	816
5.2.3.1.16. Dcm_OpStatusType	816
5.2.3.1.17. Dcm_PduInfoPtrType	816
5.2.3.1.18. Dcm_ProgConditionsType	817
5.2.3.1.19. Dcm_ProtocolType	817
5.2.3.1.20. Dcm_ResetModeType	817
5.2.3.1.21. Dcm_ResponseConfirmationTypeType	817
5.2.3.1.22. Dcm_ReturnReadMemoryType	818



5.2.3.1.23. Dcm_ReturnWriteMemoryType	818
5.2.3.1.24. Dcm_RoeStateType	818
5.2.3.1.25. Dcm_SecLevelType	818
5.2.3.1.26. Dcm_ServiceIDType	818
5.2.3.1.27. Dcm_SesCtrlType	818
5.2.3.1.28. Dcm_StatusType	819
5.2.3.2. Macro constants	819
5.2.3.2.1. DCM_ALL_SESSION_LEVEL	819
5.2.3.2.2. DCM_API_CALLER_GENERIC_SVCH_IMPLEMENTATION_ID	819
5.2.3.2.3. DCM_AR_RELEASE_MAJOR_VERSION	819
5.2.3.2.4. DCM_AR_RELEASE_MINOR_VERSION	819
5.2.3.2.5. DCM_AR_RELEASE_REVISION_VERSION	819
5.2.3.2.6. DCM_ASYNC_PROCESSING_DONE	820
5.2.3.2.7. DCM_ASYNC_PROCESSING_WAIT	820
5.2.3.2.8. DCM_BOOTLOADER_RESET	820
5.2.3.2.9. DCM_CANCEL	820
5.2.3.2.10. DCM_COLD_START	820
5.2.3.2.11. DCM_CONFIRMED_NOK	820
5.2.3.2.12. DCM_CONFIRMED_OK	821
5.2.3.2.13. DCM_DEFAULT_SESSION	821
5.2.3.2.14. DCM_DID_NOT_SUPPORTED	821
5.2.3.2.15. DCM_DID_SUPPORTED	821
5.2.3.2.16. DCM_DISABLE_RAPID_POWER_SHUTDOWN_RESET	821
5.2.3.2.17. DCM_DISABLE_RX_ENABLE_TX_NM	821
5.2.3.2.18. DCM_DISABLE_RX_ENABLE_TX_NORM	821
5.2.3.2.19. DCM_DISABLE_RX_ENABLE_TX_NORM_NM	822
5.2.3.2.20. DCM_DISABLE_RX_TX_NM	822
5.2.3.2.21. DCM_DISABLE_RX_TX_NORM	822
5.2.3.2.22. DCM_DISABLE_RX_TX_NORM_NM	822
5.2.3.2.23. DCM_DTRSTATUS_INVISIBLE	822
5.2.3.2.24. DCM_DTRSTATUS_VISIBLE	822
5.2.3.2.25. DCM_ENABLE_RAPID_POWER_SHUTDOWN_RESET	823
5.2.3.2.26. DCM_ENABLE_RX_DISABLE_TX_NM	823
5.2.3.2.27. DCM_ENABLE_RX_DISABLE_TX_NORM	823
5.2.3.2.28. DCM_ENABLE_RX_DISABLE_TX_NORM_NM	823
5.2.3.2.29. DCM_ENABLE_RX_TX_NM	823
5.2.3.2.30. DCM_ENABLE_RX_TX_NORM	823
5.2.3.2.31. DCM_ENABLE_RX_TX_NORM_NM	824
5.2.3.2.32. DCM_EXTENDED_DIAGNOSTIC_SESSION	824
5.2.3.2.33. DCM_E_ABORT	824
5.2.3.2.34. DCM_E_BRAKESWITCH_NOTCLOSED	824
5.2.3.2.35. DCM_E_BUSYREPEATREQUEST	824



5.2.3.2.36. DCM_E_COMPARE_KEY_FAILED	825
5.2.3.2.37. DCM_E_CONDITIONSNOTCORRECT	825
5.2.3.2.38. DCM_E_DONE	825
5.2.3.2.39. DCM_E_ENGINEISNOTRUNNING	825
5.2.3.2.40. DCM_E_ENGINEISRUNNING	825
5.2.3.2.41. DCM_E_ENGINERUNTIMETOOLOW	826
5.2.3.2.42. DCM_E_EXCEEDNUMBEROFATTEMPTS	826
5.2.3.2.43. DCM_E_FAILUREPREVENTSEXECUTIONOFFEATUREDACTION	826
5.2.3.2.44. DCM_E_FORCE_RCRRP	826
5.2.3.2.45. DCM_E_GENERALPROGRAMMINGFAILURE	826
5.2.3.2.46. DCM_E_GENERALREJECT	827
5.2.3.2.47. DCM_E_INCORRECTMESSAGELENGTHORINVALIDFORMAT	827
5.2.3.2.48. DCM_E_INVALIDKEY	827
5.2.3.2.49. DCM_E_NORESPONSEFROMSUBNETCOMPONENT	827
5.2.3.2.50. DCM_E_NOT_OK	827
5.2.3.2.51. DCM_E_NO_DATA	828
5.2.3.2.52. DCM_E_OK	828
5.2.3.2.53. DCM_E_PENDING	828
5.2.3.2.54. DCM_E_PERIODICID_NOT_ACCEPTED	828
5.2.3.2.55. DCM_E_POSITIVERESPONSE	828
5.2.3.2.56. DCM_E_PROCESS_PAGE	828
5.2.3.2.57. DCM_E_PROTOCOL_NOT_ALLOWED	829
5.2.3.2.58. DCM_E_REQUESTCORRECTLYRECEIVEDRESPONSEPENDING ...	829
5.2.3.2.59. DCM_E_REQUESTOUTOFRANGE	829
5.2.3.2.60. DCM_E_REQUESTSEQUENCEERROR	830
5.2.3.2.61. DCM_E_REQUEST_ENV_NOK	830
5.2.3.2.62. DCM_E_REQUEST_NOT_ACCEPTED	830
5.2.3.2.63. DCM_E_REQUIREDTIMEDELAYNOTEXPIRED	830
5.2.3.2.64. DCM_E_RESPONSETOOLONG	831
5.2.3.2.65. DCM_E_ROE_NOT_ACCEPTED	831
5.2.3.2.66. DCM_E_RPMTOOHIGH	831
5.2.3.2.67. DCM_E_RPMTOOLOW	831
5.2.3.2.68. DCM_E_SECURITYACCESSDENIED	831
5.2.3.2.69. DCM_E_SERVICENOTSUPPORTED	832
5.2.3.2.70. DCM_E_SERVICENOTSUPPORTEDINACTIVESESSION	832
5.2.3.2.71. DCM_E_SESSION_NOT_ALLOWED	832
5.2.3.2.72. DCM_E_SHIFTERLEVERNOTINPARK	832
5.2.3.2.73. DCM_E_SUBFUNCTIONNOTSUPPORTED	833
5.2.3.2.74. DCM_E_SUBFUNCTIONNOTSUPPORTEDINACTIVESESSION	833
5.2.3.2.75. DCM_E_TEMPERATURETOOHIGH	833
5.2.3.2.76. DCM_E_TEMPERATURETOOLOW	833



5.2.3.2.77. DCM_E_THROTTLE_PEDALTOOHIGH	834
5.2.3.2.78. DCM_E_THROTTLE_PEDALTOOLOW	834
5.2.3.2.79. DCM_E_TI_PREPARE_INCONSISTENT	834
5.2.3.2.80. DCM_E_TI_PREPARE_LIMITS	834
5.2.3.2.81. DCM_E_TORQUECONVERTERCLUTCHLOCKED	834
5.2.3.2.82. DCM_E_TRANSFERDATASUSPENDED	834
5.2.3.2.83. DCM_E_TRANSMISSIONRANGENOTINGEAR	835
5.2.3.2.84. DCM_E_TRANSMISSIONRANGEOTINNEUTRAL	835
5.2.3.2.85. DCM_E_UPLOADDOWNLOADNOTACCEPTED	835
5.2.3.2.86. DCM_E_VEHICLESPEDTOOHIGH	835
5.2.3.2.87. DCM_E_VEHICLESPEDTOOLOW	835
5.2.3.2.88. DCM_E_VOLTAGETOOTHIGH	836
5.2.3.2.89. DCM_E_VOLTAGETOOLOW	836
5.2.3.2.90. DCM_E_WRONGBLOCKSEQUENCECOUNTER	836
5.2.3.2.91. DCM_E_WRONG_RECORD_NUM	836
5.2.3.2.92. DCM_FINALRESPONSE	836
5.2.3.2.93. DCM_FORCED_RCRRP	836
5.2.3.2.94. DCM_FORCE_RCRRP_OK	837
5.2.3.2.95. DCM_HARD_RESET	837
5.2.3.2.96. DCM_INITIAL	837
5.2.3.2.97. DCM_KEY_ON_OFF_RESET	837
5.2.3.2.98. DCM_MODULE_ID	837
5.2.3.2.99. DCM_NO_ACTIVE_PROTOCOL	837
5.2.3.2.100. DCM_NO_RESET	838
5.2.3.2.101. DCM_OBD_ON_CAN	838
5.2.3.2.102. DCM_OBD_ON_FLEXRAY	838
5.2.3.2.103. DCM_OBD_ON_IP	838
5.2.3.2.104. DCM_OPSTATUS_INVALID	838
5.2.3.2.105. DCM_PENDING	838
5.2.3.2.106. DCM_PERIODICTRANS_ON_CAN	838
5.2.3.2.107. DCM_PERIODICTRANS_ON_FLEXRAY	839
5.2.3.2.108. DCM_PERIODICTRANS_ON_IP	839
5.2.3.2.109. DCM_PROGRAMMING_SESSION	839
5.2.3.2.110. DCM_READ_FAILED	839
5.2.3.2.111. DCM_READ_FORCE_RCRRP	839
5.2.3.2.112. DCM_READ_OK	839
5.2.3.2.113. DCM_READ_PENDING	840
5.2.3.2.114. DCM_RESET_EXECUTION	840
5.2.3.2.115. DCM_RES_NEG_NOT_OK	840
5.2.3.2.116. DCM_RES_NEG_OK	840
5.2.3.2.117. DCM_RES_POS_NOT_OK	840
5.2.3.2.118. DCM_RES_POS_OK	840



5.2.3.2.119. DCM_ROE_ACTIVE	841
5.2.3.2.120. DCM_ROE_ON_CAN	841
5.2.3.2.121. DCM_ROE_ON_FLEXRAY	841
5.2.3.2.122. DCM_ROE_ON_IP	841
5.2.3.2.123. DCM_ROE_UNACTIVE	841
5.2.3.2.124. DCM_SAFETY_SYSTEM_DIAGNOSTIC_SESSION	841
5.2.3.2.125. DCM_SEC_LEV_LOCKED	841
5.2.3.2.126. DCM_SERVID_APPLICATIONTRANSMISSIONCONFIRMATION	842
5.2.3.2.127. DCM_SERVID_COMMFLLCOMMODEENTERED	842
5.2.3.2.128. DCM_SERVID_COMMFULLCOMMODEENTERED	842
5.2.3.2.129. DCM_SERVID_COMMNOCOMMODEENTERED	842
5.2.3.2.130. DCM_SERVID_COMMON_COMM_MODE_ENTRY	842
5.2.3.2.131. DCM_SERVID_COMMSSILENTCOMMODEENTERED	842
5.2.3.2.132. DCM_SERVID_DEFAULTDIDFREEZECURRENTSTATE_DEFAULTINTERFACE 843	
5.2.3.2.133. DCM_SERVID_DEFAULTDIDREAD_DEFAULTINTERFACE	843
5.2.3.2.134. DCM_SERVID_DEFAULTDIDRESETTODEFAULT_DEFAULTINTERFACE	843
5.2.3.2.135. DCM_SERVID_DEFAULTDIDRETURNCONTROLTOECUFORALL_DEFAULTINTERFACE	843
5.2.3.2.136. DCM_SERVID_DEFAULTDIDRETURNCONTROLTOECU_DEFAULTINTERFACE	843
5.2.3.2.137. DCM_SERVID_DEFAULTDIDSHORTTERMADJUSTMENT_DEFAULTINTERFACE	844
5.2.3.2.138. DCM_SERVID_DEFAULTDIDWRITE_DEFAULTINTERFACE	844
5.2.3.2.139. DCM_SERVID_DEMTRIGGERONDTC	844
5.2.3.2.140. DCM_SERVID_GETACTIVEPROTOCOL	844
5.2.3.2.141. DCM_SERVID_GETSECURITYLEVEL	844
5.2.3.2.142. DCM_SERVID_GETSESCRLTYPE	845
5.2.3.2.143. DCM_SERVID_GETVERSIONINFO	845
5.2.3.2.144. DCM_SERVID_INIT	845
5.2.3.2.145. DCM_SERVID_MAINFNCTION	845
5.2.3.2.146. DCM_SERVID_MAINFUNCTION	845
5.2.3.2.147. DCM_SERVID_PROCESSINGDONE	846
5.2.3.2.148. DCM_SERVID_RESETTODEFAULTSESSION	846
5.2.3.2.149. DCM_SERVID_SETACTIVEDIAGNOSTIC	846
5.2.3.2.150. DCM_SERVID_SETNEGRESPONSE	846
5.2.3.2.151. DCM_SERVID_TRIGGERONEVENT	846



5.2.3.2.152. DCM_SOFT_RESET	846
5.2.3.2.153. DCM_SS_BOOTLOADER_RESET	847
5.2.3.2.154. DCM_SUPPLIER_1	847
5.2.3.2.155. DCM_SUPPLIER_10	847
5.2.3.2.156. DCM_SUPPLIER_11	847
5.2.3.2.157. DCM_SUPPLIER_12	847
5.2.3.2.158. DCM_SUPPLIER_13	847
5.2.3.2.159. DCM_SUPPLIER_14	847
5.2.3.2.160. DCM_SUPPLIER_15	848
5.2.3.2.161. DCM_SUPPLIER_2	848
5.2.3.2.162. DCM_SUPPLIER_3	848
5.2.3.2.163. DCM_SUPPLIER_4	848
5.2.3.2.164. DCM_SUPPLIER_5	848
5.2.3.2.165. DCM_SUPPLIER_6	848
5.2.3.2.166. DCM_SUPPLIER_7	848
5.2.3.2.167. DCM_SUPPLIER_8	849
5.2.3.2.168. DCM_SUPPLIER_9	849
5.2.3.2.169. DCM_SVCID_COPYRXDATA	849
5.2.3.2.170. DCM_SVCID_COPYTXDATA	849
5.2.3.2.171. DCM_SVCID_GETPROGCONDITIONS	849
5.2.3.2.172. DCM_SVCID_READMEMORY	849
5.2.3.2.173. DCM_SVCID_REQUESTDOWNLOAD	850
5.2.3.2.174. DCM_SVCID_REQUESTUPLOAD	850
5.2.3.2.175. DCM_SVCID_REQUESTTRANSFEREXIT	850
5.2.3.2.176. DCM_SVCID_REQUESTDOWNLOADEXIT	850
5.2.3.2.177. DCM_SVCID_REQUESTTRANSFEREXIT	850
5.2.3.2.178. DCM_SVCID_REQUESTUPLOAD	851
5.2.3.2.179. DCM_SVCID_SETPROGCONDITIONS	851
5.2.3.2.180. DCM_SVCID_STARTOFRECEPTION	851
5.2.3.2.181. DCM_SVCID_TPRXINDICATION	851
5.2.3.2.182. DCM_SVCID_TPTXCONFIRMATION	851
5.2.3.2.183. DCM_SVCID_TXCONFIRMATION	851
5.2.3.2.184. DCM_SVCID_WRITEMEMORY	852
5.2.3.2.185. DCM_SW_MAJOR_VERSION	852
5.2.3.2.186. DCM_SW_MINOR_VERSION	852
5.2.3.2.187. DCM_SW_PATCH_VERSION	852
5.2.3.2.188. DCM_TIMEOUT_RCRRP	852
5.2.3.2.189. DCM_UDS_ON_CAN	852
5.2.3.2.190. DCM_UDS_ON_FLEXRAY	853
5.2.3.2.191. DCM_UDS_ON_IP	853
5.2.3.2.192. DCM_UDS_RDBPI_TRANSMISSIONMODE_FASTRATE	853
5.2.3.2.193. DCM_UDS_RDBPI_TRANSMISSIONMODE_MEDIUMRATE	853



5.2.3.2.194. DCM_UDS_RDBPI_TRANSMISSIONMODE_SLOWRATE	853
5.2.3.2.195. DCM_UDS_RDBPI_TRANSMISSIONMODE_STOPSENDING	853
5.2.3.2.196. DCM_UPDATE_PAGE	854
5.2.3.2.197. DCM_VENDOR_ID	854
5.2.3.2.198. DCM_WARM_START	854
5.2.3.2.199. DCM_WRITE_FAILED	854
5.2.3.2.200. DCM_WRITE_FORCE_RCRRP	854
5.2.3.2.201. DCM_WRITE_OK	854
5.2.3.2.202. DCM_WRITE_PENDING	855
5.2.3.3. Functions	855
5.2.3.3.1. Dcm_APICaller_GenericSvcHImplementation	855
5.2.3.3.2. Dcm_ApplicationTransmissionConfirmation	855
5.2.3.3.3. Dcm_BootloaderResetModeEntry	856
5.2.3.3.4. Dcm_CopyRxData	856
5.2.3.3.5. Dcm_CopyTxData	857
5.2.3.3.6. Dcm_DefaultDIDFreezeCurrentState	858
5.2.3.3.7. Dcm_DefaultDIDRead	859
5.2.3.3.8. Dcm_DefaultDIDResetToDefault	860
5.2.3.3.9. Dcm_DefaultDIDReturnControlToECU	861
5.2.3.3.10. Dcm_DefaultDIDReturnControlToECUForAll	862
5.2.3.3.11. Dcm_DefaultDIDShortTermAdjustment	863
5.2.3.3.12. Dcm_DefaultDIDWrite	864
5.2.3.3.13. Dcm_DemTriggerOnDTCStatus	865
5.2.3.3.14. Dcm_DisableRapidPowerShutdownResetModeEntry	866
5.2.3.3.15. Dcm_DisableRxEnableTxNmModeEntry	866
5.2.3.3.16. Dcm_DisableRxEnableTxNormModeEntry	866
5.2.3.3.17. Dcm_DisableRxEnableTxNormNmModeEntry	867
5.2.3.3.18. Dcm_DisableRxTxNmModeEntry	867
5.2.3.3.19. Dcm_DisableRxTxNormModeEntry	867
5.2.3.3.20. Dcm_DisableRxTxNormNmModeEntry	868
5.2.3.3.21. Dcm_EnableRapidPowerShutdownResetModeEntry	868
5.2.3.3.22. Dcm_EnableRxDisableTxNmModeEntry	868
5.2.3.3.23. Dcm_EnableRxDisableTxNormModeEntry	869
5.2.3.3.24. Dcm_EnableRxDisableTxNormNmModeEntry	869
5.2.3.3.25. Dcm_EnableRxTxNmModeEntry	869
5.2.3.3.26. Dcm_EnableRxTxNormModeEntry	870
5.2.3.3.27. Dcm_EnableRxTxNormNmModeEntry	870
5.2.3.3.28. Dcm_ExternalProcessingDone	870
5.2.3.3.29. Dcm_ExternalSetNegResponse	871
5.2.3.3.30. Dcm_GetActiveProtocol	871
5.2.3.3.31. Dcm_GetProgConditions	872
5.2.3.3.32. Dcm_GetSecurityLevel	872



5.2.3.3.33. Dcm_GetSesCtrlType	873
5.2.3.3.34. Dcm_GetSesCtrlTypeExt	873
5.2.3.3.35. Dcm_GetVersionInfo	874
5.2.3.3.36. Dcm_HardResetModeEntry	874
5.2.3.3.37. Dcm_Init	874
5.2.3.3.38. Dcm_InjectDiagnosticRequest	875
5.2.3.3.39. Dcm_KeyOnOffResetModeEntry	875
5.2.3.3.40. Dcm_MainFunction	876
5.2.3.3.41. Dcm_ProcessRequestDownload	876
5.2.3.3.42. Dcm_ProcessRequestTransferExit	876
5.2.3.3.43. Dcm_ProcessRequestUpload	877
5.2.3.3.44. Dcm_ReadMemory	878
5.2.3.3.45. Dcm_ResetToDefaultSession	878
5.2.3.3.46. Dcm_RxIndication	878
5.2.3.3.47. Dcm_SetProgConditions	879
5.2.3.3.48. Dcm_SoftResetModeEntry	879
5.2.3.3.49. Dcm_SsBootloaderResetModeEntry	879
5.2.3.3.50. Dcm_StartOfReception	880
5.2.3.3.51. Dcm_TpRxIndication	880
5.2.3.3.52. Dcm_TpTxConfirmation	881
5.2.3.3.53. Dcm_TriggerOnEvent	881
5.2.3.3.54. Dcm_TxConfirmation	882
5.2.3.3.55. Dcm_WriteMemory	882
5.2.4. Integration notes	883
5.2.4.1. Exclusive areas	883
5.2.4.1.1. SCHM_DCM_EXCLUSIVE_AREA_0	883
5.2.4.2. Production errors	883
5.2.4.3. Memory mapping	883
5.2.4.4. Integration requirements	884
5.2.4.4.1. dev.Dcm.External.ServiceAndSubService	884
5.2.4.4.2. intgr.Dcm.ReadDTCInformation.ResponseBuffer	884
5.2.4.4.3. intgr.Dcm.DiagnosticSessionControl.ExtSubService	885
5.2.4.4.4. intgr.Dcm.ModeDeclaration.RangeLimit	885
5.2.4.4.5. intgr.Dcm.Nvm.Timing	885
5.2.4.4.6. intgr.Dcm.DynamicallyDefineDataIdentifier.ExtSubService	886
5.2.4.4.7. intgr.Dcm.ReadDTCInformation.ReportDTCFaultDetectionCounter	886
5.2.4.4.8. intgr.Dcm.NvmBlocks.ServiceNeedsWizard	887
5.2.4.4.9. intgr.Dcm.SecurityAccess.DelayTimersMigration	887
5.3. Dem	888
5.3.1. Configuration parameters	888
5.3.1.1. CommonPublishedInformation	889
5.3.1.2. DemConfigSet	892



5.3.1.3. DemDTCClass	894
5.3.1.4. DemCallbackInitMForF	898
5.3.1.5. DemEventParameter	898
5.3.1.6. DemCallbackClearEventAllowed	908
5.3.1.7. DemCallbackEventDataChanged	910
5.3.1.8. DemCallbackEventStatusChanged	910
5.3.1.9. DemCallbackInitMForE	911
5.3.1.10. DemEventClass	912
5.3.1.11. DemDebounceAlgorithmClass	923
5.3.1.12. DemDebounceMonitorInternal	923
5.3.1.13. DemCallbackGetFDC	923
5.3.1.14. DemDebounceCounterBased	924
5.3.1.15. DemDebounceTimeBase	929
5.3.1.16. DemDebounceFrequencyBased	931
5.3.1.17. DemIndicatorAttribute	933
5.3.1.18. DemPidClass	937
5.3.1.19. DemPidDataElement	938
5.3.1.20. DemAdvDisplacementPassiveTableLine	938
5.3.1.21. DemPassiveEventRefs	939
5.3.1.22. DemGeneral	939
5.3.1.23. DemClient	1009
5.3.1.24. DemServiceAPI	1010
5.3.1.25. DemBSWDistribution	1017
5.3.1.26. DemSatelliteInstanceConfig	1019
5.3.1.27. DemCallbackDTCStatusChanged	1020
5.3.1.28. DemDataElementClass	1021
5.3.1.29. DemExternalCSDaDataElementClass	1021
5.3.1.30. DemExternalSRDataElementClass	1024
5.3.1.31. DemInternalDataElementClass	1025
5.3.1.32. DemDidClass	1029
5.3.1.33. DemEnableCondition	1030
5.3.1.34. DemEnableConditionGroup	1031
5.3.1.35. DemMultiEventTriggering	1031
5.3.1.36. DemExtendedDataClass	1033
5.3.1.37. DemExtendedDataRecordClass	1033
5.3.1.38. DemFreezeFrameClass	1035
5.3.1.39. DemFreezeFrameRecNumClass	1036
5.3.1.40. DemSPNClass	1036
5.3.1.41. DemJ1939FreezeFrameClass	1037
5.3.1.42. DemGeneralOBD	1038
5.3.1.43. DemGeneralJ1939	1045
5.3.1.44. DemGroupOfDTC	1049



5.3.1.45. DemIndicator	1050
5.3.1.46. DemNvRamBlockId	1050
5.3.1.47. DemOperationCycle	1051
5.3.1.48. DemAgingCycle	1052
5.3.1.49. DemRatioId	1052
5.3.1.50. DemEventStatusPort	1056
5.3.1.51. DemOpCyclePort	1057
5.3.1.52. DemAgingCyclePort	1058
5.3.1.53. DemEnableCondPort	1059
5.3.1.54. DemStorageCondPort	1060
5.3.1.55. DemIndStatusPort	1061
5.3.1.56. DemControlDTCSuppressionPort	1062
5.3.1.57. DemCddIfPort	1063
5.3.1.58. DemResetReadinessPort	1064
5.3.1.59. DemPowerTakeOffStatusPort	1065
5.3.1.60. DemEventPort	1066
5.3.1.61. DemEventInfoPort	1069
5.3.1.62. DemPfccCycleQualifiedPort	1074
5.3.1.63. DemOverflowIndPort	1076
5.3.1.64. DemStorageCondition	1077
5.3.1.65. DemStorageConditionGroup	1078
5.3.1.66. DemDefensiveProgramming	1079
5.3.1.67. PublishedInformation	1082
5.3.2. Recommended configurations	1082
5.3.2.1. DemRecConfigurationDtcGroupsOpCycle	1082
5.3.2.1.1. DemGeneral	1082
5.3.2.1.2. DEM_DTC_GROUP_EMISSION_REL_DTCS	1083
5.3.2.1.3. DEM_DTC_GROUP_POWERTRAIN_DTCS	1083
5.3.2.1.4. DEM_DTC_GROUP_CHASSIS_DTCS	1083
5.3.2.1.5. DEM_DTC_GROUP_BODY_DTCS	1083
5.3.2.1.6. DEM_DTC_GROUP_NETWORK_COM_DTCS	1084
5.3.2.1.7. DEM_OPACYC_POWER	1084
5.3.3. Application programming interface (API)	1084
5.3.3.1. Type definitions	1084
5.3.3.1.1. Dem_DTCFormatType	1084
5.3.3.1.2. Dem_DTCGroupType	1084
5.3.3.1.3. Dem_DTCKindType	1084
5.3.3.1.4. Dem_DTCOriginType	1085
5.3.3.1.5. Dem_DTCRequestType	1085
5.3.3.1.6. Dem_DTCSeverityType	1085
5.3.3.1.7. Dem_DTCStatusMaskType	1085
5.3.3.1.8. Dem_DTCTranslationFormatType	1085



5.3.3.1.9. Dem_DataElementDataType	1085
5.3.3.1.10. Dem_DataKindType	1086
5.3.3.1.11. Dem_DebounceCounterStatusType	1086
5.3.3.1.12. Dem_DebounceResetStatusType	1086
5.3.3.1.13. Dem_DebouncingStateType	1086
5.3.3.1.14. Dem_EntryDataType	1086
5.3.3.1.15. Dem_EntryStatusType	1086
5.3.3.1.16. Dem_EvSignificanceType	1087
5.3.3.1.17. Dem_EventMemoryEntryType	1087
5.3.3.1.18. Dem_EventStatusExtendedType	1088
5.3.3.1.19. Dem_EventStatusType	1088
5.3.3.1.20. Dem_FFIdxType	1089
5.3.3.1.21. Dem_FailureCycleCounterType	1089
5.3.3.1.22. Dem_GateEntryBaseFragmentType	1089
5.3.3.1.23. Dem_GateEntryBaseType	1090
5.3.3.1.24. Dem_GateEntryDataPtrVoidType	1090
5.3.3.1.25. Dem_GateEntryDataVoidType	1090
5.3.3.1.26. Dem_GateEntryMirrorFragmentType	1091
5.3.3.1.27. Dem_GateEntryOBDFPrimaryFragmentType	1091
5.3.3.1.28. Dem_GateEntryPrimaryFragmentType	1091
5.3.3.1.29. Dem_GateEntrySecondaryFragmentType	1091
5.3.3.1.30. Dem_IndicatorCycleCounterType	1092
5.3.3.1.31. Dem_IndicatorIdType	1092
5.3.3.1.32. Dem_IndicatorStatusType	1092
5.3.3.1.33. Dem_InitMonitorReasonType	1092
5.3.3.1.34. Dem_J1939DcmDTCStatusFilterType	1092
5.3.3.1.35. Dem_J1939DcmDiagnosticReadiness1Type	1093
5.3.3.1.36. Dem_J1939DcmDiagnosticReadiness2Type	1093
5.3.3.1.37. Dem_J1939DcmDiagnosticReadiness3Type	1093
5.3.3.1.38. Dem_J1939DcmLampStatusType	1094
5.3.3.1.39. Dem_J1939DcmSetClearFilterType	1094
5.3.3.1.40. Dem_J1939DcmSetFreezeFrameFilterType	1094
5.3.3.1.41. Dem_MultipleOBDEventMemoryEntryType	1094
5.3.3.1.42. Dem_NvDataType	1095
5.3.3.1.43. Dem_NvGateEntryDataPtrVoidType	1097
5.3.3.1.44. Dem_NvGateEntryDataVoidType	1097
5.3.3.1.45. Dem_NvGateEntryMirrorDataType	1097
5.3.3.1.46. Dem_NvGateEntryPrimaryDataType	1098
5.3.3.1.47. Dem_NvGateEntrySecondaryDataType	1098
5.3.3.1.48. Dem_OccOrderType	1098
5.3.3.1.49. Dem_OperationCycleIdType	1098
5.3.3.1.50. Dem_OperationCycleStateType	1099

5.3.3.1.51. Dem_RatioidType	1099
5.3.3.1.52. Dem_ReadinessEnableStatusType	1099
5.3.3.1.53. Dem_ReturnClearDTCType	1099
5.3.3.1.54. Dem_ReturnGetNextFilteredElementType	1099
5.3.3.1.55. Dem_ReturnGetNumberOfFilteredDTCType	1100
5.3.3.1.56. Dem_ReturnSetFilterType	1100
5.3.3.1.57. Dem_SRDataElementEndiannessType	1100
5.3.3.1.58. Dem_SingleOBDEventMemoryEntryType	1100
5.3.3.1.59. Dem_SizeEntryDataType	1100
5.3.3.1.60. Dem_StatusCounterType	1100
5.3.3.1.61. Dem_UdsStatusByteType	1101
5.3.3.2. Macro constants	1101
5.3.3.2.1. DEM_AR_RELEASE_MAJOR_VERSION	1101
5.3.3.2.2. DEM_AR_RELEASE_MINOR_VERSION	1101
5.3.3.2.3. DEM_AR_RELEASE_REVISION_VERSION	1101
5.3.3.2.4. DEM_ASR42_CLEAR_BUSY	1102
5.3.3.2.5. DEM_ASR42_CLEAR_FAILED	1102
5.3.3.2.6. DEM_ASR42_CLEAR_MEMORY_ERROR	1102
5.3.3.2.7. DEM_BIG_ENDIAN	1102
5.3.3.2.8. DEM_BUFFER_TOO_SMALL	1102
5.3.3.2.9. DEM_BUSY	1102
5.3.3.2.10. DEM_CLEAR_BUSY	1103
5.3.3.2.11. DEM_CLEAR_FAILED	1103
5.3.3.2.12. DEM_CLEAR_MEMORY_ERROR	1103
5.3.3.2.13. DEM_CLEAR_OK	1103
5.3.3.2.14. DEM_CLEAR_PENDING	1103
5.3.3.2.15. DEM_CLEAR_WRONG_DTC	1103
5.3.3.2.16. DEM_CLEAR_WRONG_DTCORIGIN	1104
5.3.3.2.17. DEM_CYCLE_STATE_END	1104
5.3.3.2.18. DEM_CYCLE_STATE_START	1104
5.3.3.2.19. DEM_DATAKIND_EXTENDEDDATA	1104
5.3.3.2.20. DEM_DATAKIND_FREEZEFRAME	1104
5.3.3.2.21. DEM_DEBOUNCE_STATUS_FREEZE	1104
5.3.3.2.22. DEM_DEBOUNCE_STATUS_RESET	1104
5.3.3.2.23. DEM_DTC_FORMAT_J1939	1105
5.3.3.2.24. DEM_DTC_FORMAT_OBD	1105
5.3.3.2.25. DEM_DTC_FORMAT_UDS	1105
5.3.3.2.26. DEM_DTC_GROUP_ALL_DTCS	1105
5.3.3.2.27. DEM_DTC_GROUP_EMISSION_REL_DTCS	1105
5.3.3.2.28. DEM_DTC_KIND_ALL_DTCS	1105
5.3.3.2.29. DEM_DTC_KIND_EMISSION_REL_DTCS	1106
5.3.3.2.30. DEM_DTC_ORIGIN_MIRROR_MEMORY	1106

5.3.3.2.31. DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY	1106
5.3.3.2.32. DEM_DTC_ORIGIN_PERMANENT_MEMORY	1106
5.3.3.2.33. DEM_DTC_ORIGIN_PRIMARY_MEMORY	1106
5.3.3.2.34. DEM_DTC_ORIGIN_SECONDARY_MEMORY	1106
5.3.3.2.35. DEM_DTC_TRANSLATION_ISO11992_4	1106
5.3.3.2.36. DEM_DTC_TRANSLATION_ISO14229_1	1107
5.3.3.2.37. DEM_DTC_TRANSLATION_ISO15031_6	1107
5.3.3.2.38. DEM_DTC_TRANSLATION_J2012DA_FORMAT_04	1107
5.3.3.2.39. DEM_DTC_TRANSLATION_SAEJ1939_73	1107
5.3.3.2.40. DEM_EVENT_ID_IGNORE	1107
5.3.3.2.41. DEM_EVENT_ID_INVALID	1107
5.3.3.2.42. DEM_EVENT_SIGNIFICANCE_FAULT	1108
5.3.3.2.43. DEM_EVENT_SIGNIFICANCE_OCCURRENCE	1108
5.3.3.2.44. DEM_EVENT_STATUS_FAILED	1108
5.3.3.2.45. DEM_EVENT_STATUS_PASSED	1108
5.3.3.2.46. DEM_EVENT_STATUS_PREFAILED	1108
5.3.3.2.47. DEM_EVENT_STATUS_PREPASSED	1108
5.3.3.2.48. DEM_E_INIT_FAILED	1108
5.3.3.2.49. DEM_E_INTEGRATION	1109
5.3.3.2.50. DEM_E_INVALID_OBDMID	1109
5.3.3.2.51. DEM_E_NODATAAVAILABLE	1109
5.3.3.2.52. DEM_E_NO_DTC_AVAILABLE	1109
5.3.3.2.53. DEM_E_NO_FDC_AVAILABLE	1109
5.3.3.2.54. DEM_E_PARAM_CONFIG	1109
5.3.3.2.55. DEM_E_PARAM_DATA	1110
5.3.3.2.56. DEM_E_PARAM_LENGTH	1110
5.3.3.2.57. DEM_E_PARAM_POINTER	1110
5.3.3.2.58. DEM_E_UNINIT	1110
5.3.3.2.59. DEM_E_WRONG_BUFFERSIZE	1110
5.3.3.2.60. DEM_E_WRONG_CONDITION	1110
5.3.3.2.61. DEM_E_WRONG_CONFIGURATION	1111
5.3.3.2.62. DEM_E_WRONG_CONTEXT	1111
5.3.3.2.63. DEM_E_WRONG_DIDNUMBER	1111
5.3.3.2.64. DEM_E_WRONG_RECORDNUMBER	1111
5.3.3.2.65. DEM_FILTERED_BUFFER_TOO_SMALL	1111
5.3.3.2.66. DEM_FILTERED_NO_MATCHING_ELEMENT	1111
5.3.3.2.67. DEM_FILTERED_OK	1111
5.3.3.2.68. DEM_FILTERED_PENDING	1112
5.3.3.2.69. DEM_FILTER_ACCEPTED	1112
5.3.3.2.70. DEM_FIRST_DET_CONFIRMED_DTC	1112
5.3.3.2.71. DEM_FIRST_FAILED_DTC	1112
5.3.3.2.72. DEM_INDICATOR_BLINKING	1112



5.3.3.2.73. DEM_INDICATOR_BLINK_CONT	1112
5.3.3.2.74. DEM_INDICATOR_CONTINUOUS	1113
5.3.3.2.75. DEM_INDICATOR_FAST_FLASH	1113
5.3.3.2.76. DEM_INDICATOR_OFF	1113
5.3.3.2.77. DEM_INDICATOR_SLOW_FLASH	1113
5.3.3.2.78. DEM_INIT_MONITOR_CLEAR	1113
5.3.3.2.79. DEM_INIT_MONITOR_REENABLED	1113
5.3.3.2.80. DEM_INIT_MONITOR_RESTART	1113
5.3.3.2.81. DEM_INIT_MONITOR_STORAGE_REENABLED	1114
5.3.3.2.82. DEM_J1939AWIL_ACTIVE	1114
5.3.3.2.83. DEM_J1939AWIL_CONT_UNAVAILABLE	1114
5.3.3.2.84. DEM_J1939AWIL_FAST_FLASH	1114
5.3.3.2.85. DEM_J1939DCM_EXPANDED_FREEZEFRAME	1114
5.3.3.2.86. DEM_J1939DCM_FREEZEFRAME	1114
5.3.3.2.87. DEM_J1939DCM_SPNS_IN_EXPANDED_FREEZEFRAME	1115
5.3.3.2.88. DEM_J1939DTC_ACTIVE	1115
5.3.3.2.89. DEM_J1939DTC_CLEAR_ALL	1115
5.3.3.2.90. DEM_J1939DTC_CLEAR_PREVIOUSLY_ACTIVE	1115
5.3.3.2.91. DEM_J1939DTC_CURRENTLY_ACTIVE	1115
5.3.3.2.92. DEM_J1939DTC_PENDING	1115
5.3.3.2.93. DEM_J1939DTC_PERMANENT	1115
5.3.3.2.94. DEM_J1939DTC_PREVIOUSLY_ACTIVE	1116
5.3.3.2.95. DEM_J1939MIL_ACTIVE	1116
5.3.3.2.96. DEM_J1939MIL_CONT_UNAVAILABLE	1116
5.3.3.2.97. DEM_J1939MIL_FAST_FLASH	1116
5.3.3.2.98. DEM_J1939PIL_ACTIVE	1116
5.3.3.2.99. DEM_J1939PIL_CONT_UNAVAILABLE	1116
5.3.3.2.100. DEM_J1939PIL_FAST_FLASH	1117
5.3.3.2.101. DEM_J1939RSIL_ACTIVE	1117
5.3.3.2.102. DEM_J1939RSIL_CONT_UNAVAILABLE	1117
5.3.3.2.103. DEM_J1939RSIL_FAST_FLASH	1117
5.3.3.2.104. DEM_LITTLE_ENDIAN	1117
5.3.3.2.105. DEM_MAXNUM_OCCURRENCE_KINDS	1117
5.3.3.2.106. DEM_MODULE_ID	1118
5.3.3.2.107. DEM_MOST_RECENT_FAILED_DTC	1118
5.3.3.2.108. DEM_MOST_REC_DET_CONFIRMED_DTC	1118
5.3.3.2.109. DEM_NO_EVENT_AVAILABLE	1118
5.3.3.2.110. DEM_NO SUCH ELEMENT	1118
5.3.3.2.111. DEM_NUMBER_FAILED	1118
5.3.3.2.112. DEM_NUMBER_OK	1118
5.3.3.2.113. DEM_NUMBER_PENDING	1119
5.3.3.2.114. DEM_NUM_OBD_RDY_GROUPS	1119

5.3.3.2.115. DEM_NUM_OF_DEVFF_KIND	1119
5.3.3.2.116. DEM_OPAQUE	1119
5.3.3.2.117. DEM_PENDING	1119
5.3.3.2.118. DEM_READ_DATA_CS_PROTOTYPE_BASIC	1119
5.3.3.2.119. DEM_READ_DATA_CS_PROTOTYPE_WITH_EVENTID	1120
5.3.3.2.120. DEM_READ_DATA_SR_DATA_TYPE_UINT16	1120
5.3.3.2.121. DEM_READ_DATA_SR_DATA_TYPE_UINT32	1120
5.3.3.2.122. DEM_READ_DATA_SR_DATA_TYPE_UINT8	1120
5.3.3.2.123. DEM_SEVERITY_CHECK_AT_NEXT_HALT	1120
5.3.3.2.124. DEM_SEVERITY_CHECK_IMMEDIATELY	1120
5.3.3.2.125. DEM_SEVERITY_MAINTENANCE_ONLY	1120
5.3.3.2.126. DEM_SEVERITY_NO_SEVERITY	1121
5.3.3.2.127. DEM_SEVERITY_WWHOBD_CLASS_A	1121
5.3.3.2.128. DEM_SEVERITY_WWHOBD_CLASS_B1	1121
5.3.3.2.129. DEM_SEVERITY_WWHOBD_CLASS_B2	1121
5.3.3.2.130. DEM_SEVERITY_WWHOBD_CLASS_C	1121
5.3.3.2.131. DEM_SEVERITY_WWHOBD_CLASS_NO_CLASS	1121
5.3.3.2.132. DEM_SI30_STATUS_AGEDDT	1122
5.3.3.2.133. DEM_SI30_STATUS_EMISSIONRELATEDDT	1122
5.3.3.2.134. DEM_SI30_STATUS_SYMPTOM_SLC	1122
5.3.3.2.135. DEM_SI30_STATUS_TFSLC_AGED	1122
5.3.3.2.136. DEM_SI30_STATUS_UNCONFIRMEDDT	1122
5.3.3.2.137. DEM_SI30_STATUS_UNCONFIRMEDDT_SLC	1122
5.3.3.2.138. DEM_SI30_STATUS_UNCONFIRMEDDT_TOC	1122
5.3.3.2.139. DEM_SI30_STATUS_WIR_SLC	1123
5.3.3.2.140. DEM_SID_AssembleEDOnFDCThresholdData	1123
5.3.3.2.141. DEM_SID_AssembleEDSegIntData	1123
5.3.3.2.142. DEM_SID_CalloutDynamicDTCFnc	1123
5.3.3.2.143. DEM_SID_ClearDTC	1123
5.3.3.2.144. DEM_SID_ClearPrestoredFreezeFrame	1123
5.3.3.2.145. DEM_SID_DcmGetAvailableOBDMIDs	1124
5.3.3.2.146. DEM_SID_DcmGetDTCOfOBDFreezeFrame	1124
5.3.3.2.147. DEM_SID_DcmGetDTRData	1124
5.3.3.2.148. DEM_SID_DcmGetInfoTypeValue08	1124
5.3.3.2.149. DEM_SID_DcmGetInfoTypeValue0B	1124
5.3.3.2.150. DEM_SID_DcmGetNumTIDsOfOBDMID	1124
5.3.3.2.151. DEM_SID_DcmReadDataOfOBDFreezeFrame	1124
5.3.3.2.152. DEM_SID_DcmReadDataOfPID01	1125
5.3.3.2.153. DEM_SID_DcmReadDataOfPID1C	1125
5.3.3.2.154. DEM_SID_DcmReadDataOfPID21	1125
5.3.3.2.155. DEM_SID_DcmReadDataOfPID30	1125
5.3.3.2.156. DEM_SID_DcmReadDataOfPID31	1125



5.3.3.2.157. DEM_SID_DcmReadDataOfPID41	1125
5.3.3.2.158. DEM_SID_DcmReadDataOfPID4D	1126
5.3.3.2.159. DEM_SID_DcmReadDataOfPID4E	1126
5.3.3.2.160. DEM_SID_DcmReadDataOfPID91	1126
5.3.3.2.161. DEM_SID_DisableDTCRecordUpdate	1126
5.3.3.2.162. DEM_SID_DisableDTCSetting	1126
5.3.3.2.163. DEM_SID_EnableDTCRecordUpdate	1126
5.3.3.2.164. DEM_SID_EnableDTCSetting	1126
5.3.3.2.165. DEM_SID_GetDTCByOccurrenceTime	1127
5.3.3.2.166. DEM_SID_GetDTCOfEvent	1127
5.3.3.2.167. DEM_SID_GetDTCSelectionResult	1127
5.3.3.2.168. DEM_SID_GetDTCSelectionResultForClearDTC	1127
5.3.3.2.169. DEM_SID_GetDTCStatusAvailabilityMask	1127
5.3.3.2.170. DEM_SID_GetDebouncingOfEvent	1127
5.3.3.2.171. DEM_SID_GetEventExtendedDataRecord	1128
5.3.3.2.172. DEM_SID_GetEventExtendedDataRecordEx	1128
5.3.3.2.173. DEM_SID_GetEventFailed	1128
5.3.3.2.174. DEM_SID_GetEventFreezeFrameData	1128
5.3.3.2.175. DEM_SID_GetEventFreezeFrameDataEx	1128
5.3.3.2.176. DEM_SID_GetEventMemoryOverflow	1128
5.3.3.2.177. DEM_SID_GetEventStatus	1128
5.3.3.2.178. DEM_SID_GetEventTested	1129
5.3.3.2.179. DEM_SID_GetFaultDetectionCounter	1129
5.3.3.2.180. DEM_SID_GetFunctionalUnitOfDTC	1129
5.3.3.2.181. DEM_SID_GetIndicatorStatus	1129
5.3.3.2.182. DEM_SID_GetNextExtendedDataRecord	1129
5.3.3.2.183. DEM_SID_GetNextFilteredDTC	1129
5.3.3.2.184. DEM_SID_GetNextFilteredDTCAndFDC	1130
5.3.3.2.185. DEM_SID_GetNextFilteredDTCAndSeverity	1130
5.3.3.2.186. DEM_SID_GetNextFilteredRecord	1130
5.3.3.2.187. DEM_SID_GetNextFilteredStoredDTC	1130
5.3.3.2.188. DEM_SID_GetNextFreezeFrameData	1130
5.3.3.2.189. DEM_SID_GetNumberOfEventMemoryEntries	1130
5.3.3.2.190. DEM_SID_GetNumberOfFilteredDTC	1130
5.3.3.2.191. DEM_SID_GetPfcCycleQualified	1131
5.3.3.2.192. DEM_SID_GetSI30Status	1131
5.3.3.2.193. DEM_SID_GetSeverityOfDTC	1131
5.3.3.2.194. DEM_SID.GetSizeOfExtendedDataRecordSelection	1131
5.3.3.2.195. DEM_SID.GetSizeOfFreezeFrameSelection	1131
5.3.3.2.196. DEM_SID_GetStatusOfDTC	1131
5.3.3.2.197. DEM_SID_GetTranslationType	1132
5.3.3.2.198. DEM_SID_GetVersionInfo	1132



5.3.3.2.199. DEM_SID_Init	1132
5.3.3.2.200. DEM_SID_J1939DcmClearDTC	1132
5.3.3.2.201. DEM_SID_J1939DcmFirstDTCwithLampStatus	1132
5.3.3.2.202. DEM_SID_J1939DcmGetNextDTCwithLampStatus	1132
5.3.3.2.203. DEM_SID_J1939DcmGetNextFilteredDTC	1132
5.3.3.2.204. DEM_SID_J1939DcmGetNextFilteredRatio	1133
5.3.3.2.205. DEM_SID_J1939DcmGetNextFreezeFrame	1133
5.3.3.2.206. DEM_SID_J1939DcmGetNextSPNInFreezeFrame	1133
5.3.3.2.207. DEM_SID_J1939DcmGetNumberOfFilteredDTC	1133
5.3.3.2.208. DEM_SID_J1939DcmReadDiagnosticReadiness1	1133
5.3.3.2.209. DEM_SID_J1939DcmReadDiagnosticReadiness2	1133
5.3.3.2.210. DEM_SID_J1939DcmReadDiagnosticReadiness3	1134
5.3.3.2.211. DEM_SID_J1939DcmSetDTCFilter	1134
5.3.3.2.212. DEM_SID_J1939DcmSetFreezeFrameFilter	1134
5.3.3.2.213. DEM_SID_J1939DcmSetRatioFilter	1134
5.3.3.2.214. DEM_SID_MainFunction	1134
5.3.3.2.215. DEM_SID_PreInit	1134
5.3.3.2.216. DEM_SID_PrestoreFreezeFrame	1134
5.3.3.2.217. DEM_SID_ProcessUnconfirmedThreshold	1135
5.3.3.2.218. DEM_SID_ReIUMPRDenLock	1135
5.3.3.2.219. DEM_SID_ReIUMPRDenRelease	1135
5.3.3.2.220. DEM_SID_ReIUMPRFaultDetect	1135
5.3.3.2.221. DEM_SID_ReportErrorStatus	1135
5.3.3.2.222. DEM_SID_ResetEventDebounceStatus	1135
5.3.3.2.223. DEM_SID_ResetEventStatus	1136
5.3.3.2.224. DEM_SID_ResetReadiness	1136
5.3.3.2.225. DEM_SID_SelectDTC	1136
5.3.3.2.226. DEM_SID_SelectExtendedDataRecord	1136
5.3.3.2.227. DEM_SID_SelectFreezeFrameData	1136
5.3.3.2.228. DEM_SID_SetAgingCycleCounterValue	1136
5.3.3.2.229. DEM_SID_SetAgingCycleState	1136
5.3.3.2.230. DEM_SID_SetCycleQualified	1137
5.3.3.2.231. DEM_SID_SetDTCFilter	1137
5.3.3.2.232. DEM_SID_SetDTCSuppression	1137
5.3.3.2.233. DEM_SID_SetEnableCondition	1137
5.3.3.2.234. DEM_SID_SetEventAvailable	1137
5.3.3.2.235. DEM_SID_SetEventDisabled	1137
5.3.3.2.236. DEM_SID_SetEventStatus	1138
5.3.3.2.237. DEM_SID_SetFreezeFrameRecordFilter	1138
5.3.3.2.238. DEM_SID_SetOperationCycleCntValue	1138
5.3.3.2.239. DEM_SID_SetOperationCycleState	1138
5.3.3.2.240. DEM_SID_SetPfcCycleQualified	1138



5.3.3.2.241. DEM_SID_SetPtoStatus	1138
5.3.3.2.242. DEM_SID_SetSI30Notification	1138
5.3.3.2.243. DEM_SID_SetSI30Symptom	1139
5.3.3.2.244. DEM_SID_SetStorageCondition	1139
5.3.3.2.245. DEM_SID_SetStoredDTCFilter	1139
5.3.3.2.246. DEM_SID_SetWIRStatus	1139
5.3.3.2.247. DEM_SID_Shutdown	1139
5.3.3.2.248. DEM_SID_StoreImmediatelyNext	1139
5.3.3.2.249. DEM_SIZE_OF_AGING_COUNTER	1140
5.3.3.2.250. DEM_SIZE_STORAGE_ORDER_ID	1140
5.3.3.2.251. DEM_SW_MAJOR_VERSION	1140
5.3.3.2.252. DEM_SW_MINOR_VERSION	1140
5.3.3.2.253. DEM_SW_PATCH_VERSION	1140
5.3.3.2.254. DEM_UDS_STATUS_CDTC	1140
5.3.3.2.255. DEM_UDS_STATUS_PDTC	1140
5.3.3.2.256. DEM_UDS_STATUS_TF	1141
5.3.3.2.257. DEM_UDS_STATUS_TFSLC	1141
5.3.3.2.258. DEM_UDS_STATUS_TFTOC	1141
5.3.3.2.259. DEM_UDS_STATUS_TNCSLC	1141
5.3.3.2.260. DEM_UDS_STATUS_TNCTOC	1141
5.3.3.2.261. DEM_UDS_STATUS_WIR	1141
5.3.3.2.262. DEM_VENDOR_ID	1142
5.3.3.2.263. DEM_WRONG_DTC	1142
5.3.3.2.264. DEM_WRONG_DTCORIGIN	1142
5.3.3.2.265. DEM_WRONG_FILTER	1142
5.3.3.3. Objects	1142
5.3.3.3.1. Dem_NvData	1142
5.3.3.3.2. Dem_NvGateEntryMirrorData	1142
5.3.3.3.3. Dem_NvGateEntryPrimaryData	1143
5.3.3.3.4. Dem_NvGateEntrySecondaryData	1143
5.3.3.4. Functions	1143
5.3.3.4.1. Dem_ASR42_GetEventFreezeFrameData	1143
5.3.3.4.2. Dem_ClearDTC	1144
5.3.3.4.3. Dem_ClearPrestoredFreezeFrame	1145
5.3.3.4.4. Dem_DcmGetAvailableOBDMIDs	1146
5.3.3.4.5. Dem_DcmGetDTCOfOBDFreezeFrame	1146
5.3.3.4.6. Dem_DcmGetDTRData	1147
5.3.3.4.7. Dem_DcmGetInfoTypeValue08	1148
5.3.3.4.8. Dem_DcmGetInfoTypeValue0B	1149
5.3.3.4.9. Dem_DcmGetNextFilteredStoredDTC	1149
5.3.3.4.10. Dem_DcmGetNumTIDsOfOBDMID	1150
5.3.3.4.11. Dem_DcmReadDataOfOBDFreezeFrame	1151



5.3.3.4.12. Dem_DcmReadDataOfPID01	1152
5.3.3.4.13. Dem_DcmReadDataOfPID1C	1152
5.3.3.4.14. Dem_DcmReadDataOfPID21	1152
5.3.3.4.15. Dem_DcmReadDataOfPID30	1153
5.3.3.4.16. Dem_DcmReadDataOfPID31	1153
5.3.3.4.17. Dem_DcmReadDataOfPID41	1153
5.3.3.4.18. Dem_DcmReadDataOfPID4D	1154
5.3.3.4.19. Dem_DcmReadDataOfPID4E	1154
5.3.3.4.20. Dem_DcmReadDataOfPID91	1155
5.3.3.4.21. Dem_DcmSetStoredDTCFilter	1155
5.3.3.4.22. Dem_DisableDTCRecordUpdate	1156
5.3.3.4.23. Dem_DisableDTCSetting	1157
5.3.3.4.24. Dem_EnableDTCRecordUpdate	1158
5.3.3.4.25. Dem_EnableDTCSetting	1158
5.3.3.4.26. Dem_GetDTCByOccurrenceTime	1159
5.3.3.4.27. Dem_GetDTCOfEvent	1160
5.3.3.4.28. Dem_GetDTCSelectionResult	1161
5.3.3.4.29. Dem_GetDTCSelectionResultForClearDTC	1161
5.3.3.4.30. Dem_GetDTCStatusAvailabilityMask	1162
5.3.3.4.31. Dem_GetDebouncingOfEvent	1163
5.3.3.4.32. Dem_GetEventExtendedDataRecord	1163
5.3.3.4.33. Dem_GetEventExtendedDataRecordEx	1164
5.3.3.4.34. Dem_GetEventFailed	1165
5.3.3.4.35. Dem_GetEventFreezeFrameData	1166
5.3.3.4.36. Dem_GetEventFreezeFrameDataEx	1167
5.3.3.4.37. Dem_GetEventMemoryOverflow	1168
5.3.3.4.38. Dem_GetEventStatus	1168
5.3.3.4.39. Dem_GetEventTested	1169
5.3.3.4.40. Dem_GetFaultDetectionCounter	1169
5.3.3.4.41. Dem_GetFunctionalUnitOfDTC	1170
5.3.3.4.42. Dem_GetIndicatorStatus	1171
5.3.3.4.43. Dem_GetNextExtendedDataRecord	1171
5.3.3.4.44. Dem_GetNextFilteredDTC	1172
5.3.3.4.45. Dem_GetNextFilteredDTCAndFDC	1173
5.3.3.4.46. Dem_GetNextFilteredDTCAndSeverity	1174
5.3.3.4.47. Dem_GetNextFilteredRecord	1175
5.3.3.4.48. Dem_GetNextFreezeFrameData	1176
5.3.3.4.49. Dem_GetNumberOfEventMemoryEntries	1177
5.3.3.4.50. Dem_GetNumberOfFilteredDTC	1178
5.3.3.4.51. Dem_GetPfcCycleQualified	1178
5.3.3.4.52. Dem_GetSI30Status	1179
5.3.3.4.53. Dem_GetSeverityOfDTC	1179



5.3.3.4.54. Dem_GetSizeOfExtendedDataRecordSelection	1180
5.3.3.4.55. Dem_GetSizeOfFreezeFrameSelection	1181
5.3.3.4.56. Dem_GetStatusOfDTC	1181
5.3.3.4.57. Dem_GetTranslationType	1182
5.3.3.4.58. Dem_GetVersionInfo	1183
5.3.3.4.59. Dem_Init	1183
5.3.3.4.60. Dem_J1939DcmClearDTC	1184
5.3.3.4.61. Dem_J1939DcmFirstDTCwithLampStatus	1184
5.3.3.4.62. Dem_J1939DcmGetNextDTCwithLampStatus	1185
5.3.3.4.63. Dem_J1939DcmGetNextFilteredDTC	1185
5.3.3.4.64. Dem_J1939DcmGetNextFilteredRatio	1186
5.3.3.4.65. Dem_J1939DcmGetNextFreezeFrame	1187
5.3.3.4.66. Dem_J1939DcmGetNextSPNInFreezeFrame	1187
5.3.3.4.67. Dem_J1939DcmGetNumberOfFilteredDTC	1188
5.3.3.4.68. Dem_J1939DcmReadDiagnosticReadiness1	1188
5.3.3.4.69. Dem_J1939DcmReadDiagnosticReadiness2	1189
5.3.3.4.70. Dem_J1939DcmReadDiagnosticReadiness3	1189
5.3.3.4.71. Dem_J1939DcmSetDTCFilter	1190
5.3.3.4.72. Dem_J1939DcmSetFreezeFrameFilter	1191
5.3.3.4.73. Dem_J1939DcmSetRatioFilter	1191
5.3.3.4.74. Dem_MainFunction	1192
5.3.3.4.75. Dem_NvMEventStatusBlockCallback	1192
5.3.3.4.76. Dem_NvMGateEntryMirrorBlockCallback	1193
5.3.3.4.77. Dem_NvMGateEntryPrimaryBlockCallback	1193
5.3.3.4.78. Dem_NvMGateEntrySecondaryBlockCallback	1193
5.3.3.4.79. Dem_NvMReadCopyEventStatusBlock	1194
5.3.3.4.80. Dem_NvMReadCopyPermanentMemory	1194
5.3.3.4.81. Dem_NvMWriteCopyEventStatusBlock	1194
5.3.3.4.82. Dem_NvMWriteCopyPermanentMemory	1195
5.3.3.4.83. Dem_NvMWriteFinishedPermanentMemory	1195
5.3.3.4.84. Dem_PreInit	1195
5.3.3.4.85. Dem_PrestoreFreezeFrame	1196
5.3.3.4.86. Dem_ReplUMPRDenLock	1197
5.3.3.4.87. Dem_ReplUMPRDenRelease	1197
5.3.3.4.88. Dem_ReplUMPRFaultDetect	1198
5.3.3.4.89. Dem_ReportErrorStatus	1198
5.3.3.4.90. Dem_ResetEventDebounceStatus	1199
5.3.3.4.91. Dem_ResetEventStatus	1200
5.3.3.4.92. Dem_ResetReadiness	1200
5.3.3.4.93. Dem_SelectDTC	1201
5.3.3.4.94. Dem_SelectExtendedDataRecord	1201
5.3.3.4.95. Dem_SelectFreezeFrameData	1202

5.3.3.4.96. Dem_SetAgingCycleCounterValue	1203
5.3.3.4.97. Dem_SetAgingCycleState	1204
5.3.3.4.98. Dem_SetCycleQualified	1204
5.3.3.4.99. Dem_SetDTCFilter	1205
5.3.3.4.100. Dem_SetDTCSuppression	1206
5.3.3.4.101. Dem_SetEnableCondition	1207
5.3.3.4.102. Dem_SetEventAvailable	1208
5.3.3.4.103. Dem_SetEventDisabled	1208
5.3.3.4.104. Dem_SetEventStatus	1209
5.3.3.4.105. Dem_SetFreezeFrameRecordFilter	1209
5.3.3.4.106. Dem_SetOperationCycleCntValue	1210
5.3.3.4.107. Dem_SetOperationCycleState	1211
5.3.3.4.108. Dem_SetPfcCycleQualified	1211
5.3.3.4.109. Dem_SetPtoStatus	1212
5.3.3.4.110. Dem_SetSI30Notification	1212
5.3.3.4.111. Dem_SetSI30Symptom	1212
5.3.3.4.112. Dem_SetStorageCondition	1213
5.3.3.4.113. Dem_SetWIRStatus	1213
5.3.3.4.114. Dem_Shutdown	1214
5.3.4. Integration notes	1215
5.3.4.1. Exclusive areas	1215
5.3.4.1.1. SCHM DEM EXCLUSIVE AREA 0	1215
5.3.4.1.2. Exclusive areas for BSW distribution functionality	1215
5.3.4.2. Production errors	1215
5.3.4.3. Memory mapping	1215
5.3.4.4. Integration requirements	1216
5.4. FiM	1216
5.4.1. Configuration parameters	1216
5.4.1.1. CommonPublishedInformation	1217
5.4.1.2. FiMConfigSet	1220
5.4.1.3. FiMEventSummary	1220
5.4.1.4. FiMFID	1221
5.4.1.5. FiMMassage	1224
5.4.1.6. FiMInhibitionConfiguration	1225
5.4.1.7. FiMInhEventId	1227
5.4.1.8. FiMInhRefChoice	1227
5.4.1.9. FiMInhChoiceDemRef	1227
5.4.1.10. FiMInhChoiceSumRef	1228
5.4.1.11. FiMSummaryEventId	1228
5.4.1.12. FiMGeneral	1229
5.4.1.13. PublishedInformation	1236
5.4.2. Application programming interface (API)	1237



5.4.2.1. Macro constants	1237
5.4.2.1.1. FIM_AR_RELEASE_MAJOR_VERSION	1237
5.4.2.1.2. FIM_AR_RELEASE_MINOR_VERSION	1237
5.4.2.1.3. FIM_AR_RELEASE_REVISION_VERSION	1237
5.4.2.1.4. FIM_EXTENSION_LVL1	1238
5.4.2.1.5. FIM_EXTENSION_LVL2	1238
5.4.2.1.6. FIM_E_EVENTID_OUT_OF_RANGE	1238
5.4.2.1.7. FIM_E_FID_OUT_OF_RANGE	1238
5.4.2.1.8. FIM_E_INVALID_POINTER	1238
5.4.2.1.9. FIM_E_MESSAGEID_OUT_OF_RANGE	1239
5.4.2.1.10. FIM_E_UNINIT	1239
5.4.2.1.11. FIM_E_WRONG_PERMISSION_REQ	1239
5.4.2.1.12. FIM_E_WRONG_TRIGGER_ON_EVENT	1239
5.4.2.1.13. FIM_INSTANCE_ID	1239
5.4.2.1.14. FIM_MODULE_ID	1240
5.4.2.1.15. FIM_NO_EXTENSION	1240
5.4.2.1.16. FIM_SID_DEM_INIT	1240
5.4.2.1.17. FIM_SID_DEM_TRIGGER_ON_EVENT	1240
5.4.2.1.18. FIM_SID_DEM_TRIGGER_ON_UNCONFIRMED	1240
5.4.2.1.19. FIM_SID_GET_FUNCTION_PERMISSION	1240
5.4.2.1.20. FIM_SID_GET_MESSAGE_STATUS	1241
5.4.2.1.21. FIM_SID_GET_VERSION_INFO	1241
5.4.2.1.22. FIM_SID_INIT	1241
5.4.2.1.23. FIM_SID_MAIN_FUNCTION	1241
5.4.2.1.24. FIM_SW_MAJOR_VERSION	1241
5.4.2.1.25. FIM_SW_MINOR_VERSION	1242
5.4.2.1.26. FIM_SW_PATCH_VERSION	1242
5.4.2.1.27. FIM_VENDOR_ID	1242
5.4.2.2. Objects	1242
5.4.2.2.1. FiM_FidInhibitionFlag	1242
5.4.2.3. Functions	1242
5.4.2.3.1. FiM_DemInit	1242
5.4.2.3.2. FiM_DemTriggerOnEventStatus	1243
5.4.2.3.3. FiM_GetFunctionPermission	1243
5.4.2.3.4. FiM_GetVersionInfo	1244
5.4.2.3.5. FiM_Init	1244
5.4.2.3.6. FiM_MainFunction	1245
5.4.3. Integration notes	1245
5.4.3.1. Exclusive areas	1245
5.4.3.1.1. SCHM_FIM_EXCLUSIVE_AREA_0	1245
5.4.3.2. Production errors	1245
5.4.3.3. Memory mapping	1245

5.4.3.4. Integration requirements	1246
6. ACG8 Diagnostic Stack tutorial	1247
6.1. Overview	1247
6.1.1. Goal of this tutorial	1247
6.1.2. Scope of this tutorial	1247
6.1.3. Prerequisites	1247
6.2. Background information	1248
6.2.1. Purpose of the Diagnostic Stack	1248
6.2.2. Concept of this configuration	1249
6.3. Creating a new project	1249
6.4. Configuring the Diagnostic Event Manager (Dem)	1250
6.5. Configuring the data routing and communication mode (EcuC, PduR, ComM)	1255
6.6. Configuring the Diagnostic Communication Manager (Dcm)	1259
6.7. Configuring the Function Inhibition Manager (Fim)	1260



1. Overview of EB tresos AutoCore Generic 8 Diagnostic Stack documentation

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

This document provides:

- ▶ [Chapter 2, “Supported features”](#): list of features supported by the ACG8 Diagnostic Stack
- ▶ [Chapter 3, “ACG8 Diagnostic Stack release notes”](#): release notes for the ACG8 Diagnostic Stack modules
- ▶ [Chapter 4, “ACG8 Diagnostic Stack user guide”](#): background information and instructions
- ▶ [Chapter 5, “ACG8 Diagnostic Stack module references”](#): information about configuration parameters and the application programming interface
- ▶ [Chapter 6, “ACG8 Diagnostic Stack tutorial”](#): how to configure the basic functions of ACG8 Diagnostic Stack modules



2. Supported features

2.1. Overview

This chapter provides an overview of the products of ACG8 Diagnostic Stack and the features that are currently supported.

[Section 2.2, “Product details”](#) contains an overview of the products of ACG8 Diagnostic Stack.

[Section 2.3.1, “Supported Dcm features”](#) contains an overview of Dcm features.

[Section 2.3.2, “Supported Dem features”](#) contains an overview of Dem features.

[Section 2.3.3, “Supported FiM features”](#) contains an overview of FiM features.

2.2. Product details

The ACG8 Diagnostic Stack product group comprises the following individual articles which are implemented according to the respective AUTOSAR 4.0.3 specifications and selected features of newer AUTOSAR releases.

- ▶ **ACG8 Diagnostic Stack:** This product contains the AUTOSAR basic software modules Dcm, Dem, and FiM which support the unified diagnostic services (UDS) ISO 14229-1 protocol.
- ▶ **ACG8 Diagnostic Stack OBD:** The features of this product enhance the ACG8 Diagnostic Stack with OBD II functionality according to ISO 15031-5 and SAE J1979 (2012). This product must be licensed in addition to ACG8 Diagnostic Stack.



NOTE



The OBD-relevant functionalities of the Diagnostic Stack depend on various aspects specified by the customer or determined by the customer's implementation, such as:

- ▶ configuration
- ▶ integration approach
- ▶ complementing SWCs and CDDs
- ▶ monitors

The legislative requirements for the OBD system (e.g., CARB CCR 13 section 1968.2, UN-ECE GTR 5) differ based on the market, vehicle type, model year, etc. Elektrobit is not responsible for the compliance with the customer's relevant legislative requirements and the related OBD homologation/type approval certification process.

Elektrobit provides a standard product and does not warrant or guarantee that the standard product can be integrated, used, or certified without further adaptation or customization in each and any customer configuration, implementation, or OBD system in compliance with the applicable legislation. The customer is solely responsible to implement and ensure compliance and certification.

ACG8 Diagnostic Stack contains the diagnostic-specific AUTOSAR modules Dcm, Dem and FiM. The Dcm module provides an AUTOSAR-compatible implementation of the unified diagnostic services (UDS) ISO 14229-1 to be used by external diagnostic tools during development, manufacturing and service. The Dem module is responsible for processing and storing diagnostic events (errors) and associated data. It provides fault information to the Dcm and offers interfaces to the application layer and to other BSW modules. The FiM module provides a control mechanism to deactivate certain functionality of software components, e.g. upon certain Dem diagnostic events.

ACG8 Diagnostic Stack includes the following basic software modules:

Basic software modules	Module abbreviation
Diagnostic Communication Manager	Dcm
Diagnostic Event Manager	Dem
Function Inhibition Manager	FiM

2.2.1. ACG8 Diagnostic Stack

ACG8 Diagnostic Stack provides AUTOSAR modules for the EB tresos AutoCore Generic (ACG) product line. ACG8 Diagnostic Stack is based on AUTOSAR 4.0.3 with selected features from later AUTOSAR releases (e.g. 4.3 and 4.4) and EB-specific enhancements implemented compatible to the AUTOSAR standard.

2.2.2. ACG8 Diagnostic Stack OBD

The features of this product enhance the ACG8 Diagnostic Stack with OBD II functionality according to ISO 15031-5 and SAE J1979 (2012). ECUs with OBD support can be divided into three classes:

OBD class	Description	Required Article
Primary ECU	Primary OBD ECUs are capable of directly communicating with an OBD scan tool. Primary OBD ECUs can collect OBD-related information from secondary OBD ECUs. Additional OEM-specific AUTOSAR software components (SWCs) or complex device drivers (CDDs) are needed to supply OBD functionality.	ACG8 Diagnostic Stack and ACG8 Diagnostic Stack OBD Implementation of necessary OEM-specific functionality can be offered upon request.
Master ECU (one per vehicle)	A MIL Master ECU is a primary OBD ECU with additional functionality. The main task of the MIL Master ECU is to total the malfunction indicator lamp (MIL) status reported by the primary ECUs and report it to the instrument cluster.	MIL Master ECU functionality can be offered upon request.
Dependent /secondary ECU	ECU without direct communication to an (OBD) scan tool. OEM-specific AUTOSAR software components (SWCs) or complex device drivers (CDDs) are needed to supply the OBD functionality and the communication path between primary and secondary OBD ECUs.	ACG8 Diagnostic Stack Implementation of necessary OEM functionality can be offered upon request.

Remark: The monitor functionality and the provision of OBD II data which are not part of the Dem module must be implemented by the integrator of the Diagnostic Stack in a specific ECU project.

This product must be licensed in addition to ACG8 Diagnostic Stack. It provides the additional features marked with OBD in [Section 2.3, “Feature details”](#).

2.3. Feature details

This chapter contains an overview of the supported and unsupported features. They are mainly grouped according to the SWS documents but EB-specific and OEM-specific features are also part of the feature list. The values in the column *Support* define if the feature is supported directly in the base package or if a feature package is necessary in addition to the base package:

- ▶ no = feature is not supported



- ▶ yes = feature is supported in the base product ACG8 Diagnostic Stack
- ▶ OBD = feature is part of ACG8 Diagnostic Stack OBD

2.3.1. Supported Dcm features

2.3.1.1. Dcm feature sets

Feature set	Feature	Support
Diagnostic protocol handling	Handling of different diagnostic protocols	yes
	Support of protocol preemption	yes
	Support of BusyRepeatRequest handling	yes
	Parallel processing of OBD and UDS	OBD
	Buffer configuration with support of shared buffers	yes
	Support of paged buffered transmission for service 0x19	yes
	Support of paged buffered transmission for other services	no
Request handling	Support of concurrent <i>TesterPresent</i> (<i>keep alive logic</i>)	yes
	Request arbitration	yes
	Support for generic interface to inject service requests from application software	yes
	Support of MetaData handling via EB-specific EcuC API for reception	yes
Response handling	Guarantee response timing to tester	yes
	Support of periodic transmission/UUDT support	yes
	Support of ResponseOnEvent (ROE) transmission with limitations	yes
	Support of ResponseOnEvent (ROE) transmission according to AUTOSAR 4.2.1	yes
	Support of segmented response (only for ReadDtclnformation)	yes
	Support of ResponsePending	yes
	Support of parallel execution of protocols and periodic transmissions	yes
	Suppression of responses according to ISO 14229 2006 and 2013	yes



Feature set	Feature	Support
	Support for selectable NRC behavior for unsupported OBD services	yes
	Support of MetaData handling via EB-specific EcuC API for transmission	yes
Security level handling	Manage security level	yes
	Persistent storage of SecurityAttemptCounter with EB-specific interface	yes
	Persistent storage of SecurityAttemptCounter with AUTOSAR 4.2.1 specific interface	yes
	Delay time on boot	yes
Session state handling	Manage session state	yes
	Keep track of active non-default sessions	yes
	Allow to modify protocol timings	yes
	Support of user-defined sessions not specified in ISO	yes
	Notification callout on S3Timeout	yes
Communication mode handling	Handling of communication requirements (full/silent/no communication)	yes
	Indicating of active/inactive diagnostic	yes
Lower layer interfaces	PduR interface according to AUTOSAR 4.x except generic connection handling	yes
	PduR interface according to AUTOSAR 3.x	no
Request verification	Verification of service identifier	yes
	Handling of suppressPosRspMsgIndicationBit	yes
	Verification of diagnostic session	yes
	Verification of ModeRules	yes
	Verification of service security access levels	yes
	Check format and subfunction support	yes
	Verification of the manufacturer application environment/permission	yes
	Verification of the supplier application environment/permission	yes
General DSP features	Mode switching via BswM callouts	yes
	Diagnostic mode declaration groups	yes



Feature set	Feature	Support
	Sender/Receiver Communication - Services 0x22 and 0x2E	yes
	Sender/Receiver Communication - Service 0x2F	no
	Sender/Receiver Communication - Services 0x01 and 0x02	no
Service processing	Asynchronous service processing via separate task	yes
	External/user-defined service handler	yes
	Possibility to call the generic service handler from an external service handler	yes
	External/user-defined subservice handler except for SID: 0x27, 0x3E, and 0x31	yes
	Bootloader interaction (jump to/from)	yes
	Reset condition management	no
UDS services	Service table per protocol	yes
	UDS services according to separate table below	yes
OBD services	Service table per protocol	OBD
	OBD services according to separate table below	OBD
Calibration	Calibration of used value for selected configuration parameter: PIDs, Services, MIDs, VehInfos, RequestControl ^a	OBD
Configuration	Support of pre-compile configuration	yes
	Support of post-build configuration	no
Variant management	Support of different variants of an ECU (different supported DIDs) in one configuration	no
	Switches between different OBD variants must be implemented in Xxx_Indication() function as part of the integration code.	no
BSW distribution	Support of Dcm and ComM on different partitions (cores)	yes
Miscellaneous	Support Memory Labels as configuration of memory ranges for UDS Services 0x23 and 0x3D	yes
	Support for project specific callout for TxConfirmation	yes
	Support for generic end-of-line extensions for RIDs (Routine IDs)	yes
	Support for Routine Info Byte handling for Routine Control service (0x31) when a byte from the application is requested to be added on the response	yes



2.3.1.2. UDS services

Service ID	Service name	Subfunction	Support
0x10	DiagnosticSessionControl	Support for NO_BOOT Sessions	yes
		Support for BOOT Sessions with final response from boot	yes
		Support for BOOT Sessions with final response from application	no
0x11	ECUReset	0x01 hardReset	yes
		0x02 keyOffOnReset	yes
		0x03 softReset	yes
		0x04 enableRapidPowerShutDown	yes
		0x05 disableRapidPowerShutDown	yes
0x14	ClearDiagnosticInformation (according to AUTOSAR 4.4.0)		yes
0x19	ReadDTCInformation (Dem-Dcm interface based on AUTOSAR 4.-3.1 interface names)	0x01 reportNumberOfDTCByStatusMask	yes
		0x02 reportDTCByStatusMask	yes
		0x03 reportDTCSnapshotIdentification	yes
		0x04 reportDTCSnapshotRecordByDTCNumber	yes
		0x05 reportDTCStoredDataByRecordNumber	no
		0x06 reportDTCExtendedDataRecordByDTCNumber	yes
		0x07 reportNumberOfDTCBySeverityMaskRecord	yes
		0x08 reportDTCBySeverityMaskRecord	yes
		0x09 reportSeverityInformationOfDTC	yes
		0x0A reportSupportedDTC	yes
		0x0B reportFirstTestFailedDTC	yes
		0x0C reportFirstConfirmedDTC	yes
		0x0D reportMostRecentTestFailedDTC	yes
		0x0E reportMostRecentConfirmedDTC	yes
		0x0F reportMirrorMemoryDTCByStatusMask	no
		0x10 reportMirrorMemoryDTCExtDataRecordByDTCNumber	no
		0x11 reportNumberOfMirrorMemoryDTCByStatusMask	no



Service ID	Service name	Subfunction	Support
		0x12 reportNumberOfEmissionsOBDDTCByStatus-Mask	no
		0x13 reportEmissionsRelatedOBDDTCByStatus-Mask	yes
		0x14 reportDTCFaultDetectionCounter	yes
		0x15 reportDTCWithPermanentStatus	yes
		0x16 reportDTCExtDataRecordByRecordNumber	no
		0x17 reportUserDefMemoryDTCByStatusMask	yes
		0x18 reportUserDefMemoryDTCSnapshotRecordBy-DTCNumber	yes
		0x19 reportUserDefMemoryDTCExtDataRecordBy	yes
		0x1A reportSupportedDTCExtDataRecord	no
		0x42 reportWWHOBDDTCByMaskRecord	no
		0x55 reportWWHOBDDTCWithPermanentStatus	no
		0x56 reportDTCInformationByDTCReadinessGroupIdentifier	no
0x22	ReadDataByIdentifier	C/S interface (AUTOSAR 4.0.3 and 4.2.1)	yes
		Callouts (AUTOSAR 4.0.3 and 4.2.1)	yes
		NvM interface	yes
		S/R interface	yes
		IoHwAb	no
		DID Ranges (AUTOSAR 4.2.1)	yes
		OBD DIDs in range (F400 .. F8FF)	OBD
		Dynamical defined DIDs	yes
		Endianness interpretation for DID signals according to AUTOSAR 4.3	yes
		Support for generic end-of-line extensions for DIDs	yes
0x23	ReadMemoryByAddress		yes
0x24	ReadScalingDataByIdentifier	C/S interface	yes
		Callouts	yes
0x27	SecurityAccess	0x01, 0x03, .. 0x41 requestSeed	yes
		0x02, 0x04, .. 0x42 sendKey	yes



Service ID	Service name	Subfunction	Support
0x28	CommunicationControl	0x00 enableRxAndTx	yes
		0x01 enableRxAndDisableTx	yes
		0x02 disableRxAndEnableTx	yes
		0x03 disableRxAndTx	yes
		0x04 enableRxAndDisableTxWithEnhancedAddressInformation	yes
		0x05 enableRxAndTxWithEnhancedAddressInformation	yes
0x29	Authentication		no
0x2A	ReadDataByPeriodicIdentifier	0x01 sendAtSlowRate	yes
		0x02 sendAtMediumRate	yes
		0x03 sendAtFastRate	yes
		0x04 stopSending	yes
		UUDT on CAN	yes
		UUDT on FlexRay - The integrator shall provide a ComplexDeviceDriver (CDD) in order to set the ISO14229-4 C_AI and FPL parameters	yes
0x2C	DynamicallyDefineDataIdentifier	0x01 defineByIdentifier	yes
		0x02 defineByMemoryAddress	yes
		0x03 clearDynamicallyDefinedDataIdentifier	yes
0x2E	WriteDataByIdentifier	Support for static DIDs only	yes
		C/S interface	yes
		Callouts	yes
		NvM interface	yes
		S/R interface	yes
		IoHwAb Support	no
		DID Ranges Support	yes
		Endianness interpretation for DID signals according to AUTOSAR 4.3	yes
		Support for generic end-of-line extensions for DIDs	yes
0x2F	InputOutputcontrolByIdentifier	C/S interface	yes
		Callouts	yes



Service ID	Service name	Subfunction	Support
		S/R interface	no
		Support for generic end-of-line extensions for DIDs	yes
		Support for ControlMaskRecord handling according to AUTOSAR 4.4	yes
0x31	RoutineControl	0x01 startRoutine (AUTOSAR 4.0.3 and 4.2.1)	yes
		0x02 stopRoutine (AUTOSAR 4.0.3 and 4.2.1)	yes
		0x03 requestRoutineResults (AUTOSAR 4.0.3 and 4.2.1)	yes
		OBD TIDs in the range 0xE000-0xE0FF	OBD
		Support for array types (AUTOSAR 4.3)	yes
		Storage of output parameter values between service handler calls	yes
0x34	RequestDownload		yes
0x35	RequestUpload		yes
0x36	TransferData		yes
0x37	RequestTransferExit		yes
0x38	RequestFileTransfer		no
0x3D	WriteMemoryByAddress		yes
0x3E	TesterPresent	0x00, 0x80	yes
		not 0x00 and not 0x80 NRC SubFunctionNotSupported as specified by AUTOSAR	yes
0x85	ControlDTCSetting	0x01 on	yes
		0x02 off	yes
		Support of configurable request length check with parameter DTCSettingControlOptionRecord	yes
0x86	ResponseOnEvent	0x00 stopResponseOnEvent non-persistent and persistent storeEvent	yes
		0x01 onDTCStatusChange non-persistent and persistent storeEvent	yes
		0x03 onChangeOfDataIdentifier	no
		0x04 reportActivatedEvents non-persistent storeEvent	yes



Service ID	Service name	Subfunction	Support
		0x05 startResponseOnEvent non-persistent and persistent storeEvent	yes
		0x06 clearResponseOnEvent non-persistent and persistent storeEvent	yes
		0x07 onComparisonOfValues NRC SubFunctionNotSupported as specified by AUTOSAR	yes
		0x08 reportMostRecentDtcOnStatusChange	no
		0x09 reportDTCRecordInformationOnDtcStatusChange	no
		Type1 response messages	yes
		Type2 response messages	yes
		Support for queuing/retry of ROE messages	no
0x87	LinkControl	0x01 verifyBaudrateTransitionWithFixedBaudrate	yes
		0x02 verifyBaudrateTransitionWithSpecificBaudrate	yes
		0x03 transitionBaudrate	yes
0xAF ^a	ReadGenericInformation	0x04 reportGenericSnapshotByDTCNumber	OBD
		0x06 reportGenericExtendedDataByDTCNumber	OBD
0xXX	All other services	NRC ServiceNotSupported as specified in AUTOSAR	yes

^aVolvo Car Corporation specific extension, according to SWRS_31835919-007-VCC AUTOSAR Dcm

2.3.1.3. OBD services

Service ID	Service name	Details	Support
0x01	Request current powertrain diagnostic data	Support for “availability PIDs” PIDs supplied by the application Read PIDs supplied by the Dem module S/R interfaces C/S interfaces C-Callback	OBD
0x02	Request powertrain freeze frame data	Support for “availability PIDs” Support to query freeze frame DTC (PID \$02)	OBD



Service ID	Service name	Details	Support
		Access freeze frame data of all PIDs of mode \$01	OBD
0x03	Request emission-related diagnostic trouble codes	Access to DTCs supplied by the Dem module	OBD
0x04	Clear/reset emission-related diagnostic information	Clear all stored emission-related diagnostic stored by the Dem module	OBD
0x06	Request on-board monitoring test results for specific monitored systems	Support for "availability OBDMIDs"	OBD
		OBD MIDs supplied by the application (AUTOSAR 4.0.3)	OBD
		OBD MIDs supplied by the Dem (AUTOSAR 4.2.1)	OBD
0x07	Request emission-related diagnostic trouble codes detected during current or last completed driving cycle	Access to DTCs supplied by the Dem module	OBD
0x08	Request control of on-board system, test or component	Support for <i>availability TIDs</i>	OBD
		Support for TIDs provided by the application	OBD
0x09	Request vehicle information	Support for <i>availability InfoTypes</i>	OBD
		Support for InfoTypes provided by the application	OBD
		Support for InfoTypes (IUMPR) provided by Dem	no
		Support of internal NoOfDataItems calculation	OBD
		Support of NoOfDataItems obtained from the application according to AUTOSAR 4.4	OBD
0x0A	Request emission-related diagnostic trouble codes with permanent status	Access to DTCs supplied by the Dem module	OBD

2.3.2. Supported Dem features

Feature set	Feature	Support
Startup behavior	Two step initialization with Dem_PrelInit and Dem_Init	yes
Monitor re-initialization	Monitor re-initialization with callback DemInitMonitorForEvent	yes
	DEM_INIT_MONITOR_CLEAR, DEM_INIT_MONITOR_RESTART	yes
	DEM_INIT_MONITOR_REENABLED (AUTOSAR 4.2)	yes



Feature set	Feature	Support
	DEM_INIT_MONITOR_STORAGE_REENABLED (AUTOSAR 4.2)	no
Diagnostic event	Event priority	yes
	Event occurrence	yes
	Event kind (BSW/SWC)	yes
	Event destination: primary, mirror, secondary	yes
	Event destination: permanent	OBD
	Event destination (UserDefined) (AUTOSAR 4.2)	no
	Event dependencies	no
Diagnostic trouble code	UDS DTC kind and format	yes
	OBD DTC kind and format	OBD
	J1939 DTC kind and format	yes
	DTC groups	yes
	DTC severity	yes
	Functional unit	yes
	DTC significance	yes
	Suppress DTC output via calibration/Availability of events via calibration	yes
	Suppress DTC output via API	yes
	Non-volatile storage of suppressed DTC status	no
	Availability of events via API	yes
	Support for replacement DTCs	yes
Monitored components	Monitored components (AUTOSAR 4.2)	no
Event memory: event status management	Dem_SetEventStatus/Dem_GetEventStatus	yes
	Dem_GetEventFailed/Dem_GetEventTested	yes
	Dem_ResetEventStatus	yes
	Dem_GetDebouncingOfEvent (AUTOSAR 4.2)	no
	[Event/DTC]StatusChangedCallback	yes
	Confirmed bit without event memory entry - configurable	yes
	Pending bit without event memory entry - configurable	yes
	Non-volatile storage of TestFailed bit - configurable globally	yes
	Non-volatile storage of TestFailed bit - configurable per event	yes



Feature set	Feature	Support
	Intermediate storage of the DTC status - DTC status is stored immediately in the NVRAM due to an update of event-related data	yes
Event memory: event memory management	Event retention	yes
	Asynchronous retention in MainFunction	yes
	Configurable retention with TEST_FAILED, PENDING, and CONFIRMED	no
	Clearing event memory entries (supported APIs for CDDs, UDS, OBD, J1939)	yes
	Configurable clear behavior with VOLATILE, NON-VOLATILE_TRIGGER, and NONVOLATILE_FINISH	yes
	Configurable clear event allowed behavior	yes
	Event memory overflow indication	yes
	Read number of event memory entries	yes
	Configurable event displacement with NONE, FULL, and PRIO_OCC (AUTOSAR 4.2)	yes
	OBD event displacement	OBD
	Advanced event displacement algorithm (passive, SI30, and lookup table) ^a	OBD
	Reporting order of event memory entries	yes
	Event burst reduction for clear DTC processing	yes
Event memory: debouncing of diagnostic events	Counter-based debounce algorithm	yes
	Time-based debounce algorithm	yes
	Frequency-based debounce algorithm (AUTOSAR 3.1)	yes
	Monitor internal debounce algorithm	yes
	Reset of running debouncing with new operation cycle/ClearDTC	yes
	Reset/freeze behavior for debouncing (API/EnableConditions/0x85)	yes
	Fault detection counter retrieval	yes
	Calculation for fault detection counter reporting:	
	▶ FDC12 similar to DEM_MAX_FDC_SINCE_LAST_CLEAR ^a	yes
	▶ DEM_MAX_FDC_DURING_CURRENT_CYCLE	no



Feature set	Feature	Support
Event memory: fault confirmation	Fault confirmation and confirmed state (ConfirmedDTCBit)	yes
	Grouping of association of events for OBD purpose	no
Event memory: event combination	Combination on storage	yes
	Combination on retrieval	no
	Support for combined OBD DTCs	OBD
Event memory: conditions	Enable conditions	yes
	Storage conditions	no
Event memory: event-related data	Non-emission-related freeze frame data:	
	▶ Trigger storage time	yes
	▶ Trigger FDC threshold ^a	yes
	▶ Trigger TEST_FAILED	yes
	▶ Triggers PENDING and CONFIRMED (AUTOSAR 4.1)	no
	▶ Configurable trigger point at FreezeFrameRecordClass level	no
	▶ Capture mode synchronous	yes
	▶ Capture mode asynchronous	yes
	▶ Pre-storage of freeze frame data	yes
	▶ Common freeze frame to simplify the configuration	yes
	▶ Development freeze frames Rec30 and Rec40 for testing and debugging ^a	yes
	Emission-related freeze frame data:	
	▶ Multiple OBD freeze frame	OBD
	▶ Global OBD freeze frame with event priority-based displacement behavior ^a	OBD
	Extended data:	
	▶ Trigger storage time	yes
	▶ Trigger FDC threshold ^a	yes
	▶ Trigger TEST_FAILED	yes
	▶ Triggers PENDING and CONFIRMED (AUTOSAR 4.1)	no
	▶ Configurable trigger point at ExtendedDataRecordClass level	no
	▶ Capture mode synchronous	yes



Feature set	Feature	Support
	<ul style="list-style-type: none"> ▶ Capture mode asynchronous 	yes
	Interfaces and internal data:	
	<ul style="list-style-type: none"> ▶ C/S interface 	yes
	<ul style="list-style-type: none"> ▶ S/R interface 	yes
	<ul style="list-style-type: none"> ▶ C-Callback 	yes
	<ul style="list-style-type: none"> ▶ Support for extended DataService interface with EventID (C/S and C-API) 	yes
Event memory: event-related data	Internal data elements for extended data:	
	<ul style="list-style-type: none"> ▶ Aging, fault detection and occurrence counter, overflow indication, significance 	yes
	<ul style="list-style-type: none"> ▶ Configurable aging with UPCNT or DOWNCNT 	yes
	<ul style="list-style-type: none"> ▶ SI30, current fault detection counter, FDC12, OCC1-7^a 	yes
	<ul style="list-style-type: none"> ▶ Cycle since first and last failed 	yes
	<ul style="list-style-type: none"> ▶ Failed cycles and max FDC 	no
	<ul style="list-style-type: none"> ▶ Additional internal data element priority^a 	yes
	<ul style="list-style-type: none"> ▶ Additional internal data element event ID 	yes
	<ul style="list-style-type: none"> ▶ Additional internal data element MIL-ON-Counter/MIL-OFF-Counter 	no
	Notification of data changes (AUTOSAR 4.0.3 and 4.2.1)	yes
Event memory: operation cycle management	Operation cycle management	yes
	Configurable autostart behavior for operation cycles	no
	Configurable automatic end behavior for operation cycles	yes
	Operation cycle states available over the power cycle (non-volatile storage)	yes
	Usage of OBD driving cycle qualification status to delay event confirmation and MIL status update	yes
Event memory: aging of diagnostic events	Internal aging:	
	<ul style="list-style-type: none"> ▶ Based on Dem_SetOperationCycleState 	yes
	<ul style="list-style-type: none"> ▶ Based on Dem_SetAgingCycleState 	yes
	External aging	no
Event memory: healing of diagnostic events	Warning indicator handling	yes
	Dem_GetIndicatorStatus	yes



Feature set	Feature	Support
	Dem_SetIndicatorStatus	no
	User-controlled WarningIndicatorRequested-bit	yes
	Handling of the warning indicator lamp (MIL)	yes
BSW error handling	Support BSW event reports	yes
	Queuing of BSW event reports between Pre_Init and Init	yes
	Explicit configuration of events that are reported before Init	no
OBD: PIDs provided by Dem	PID \$01 monitor status since DTCs cleared (4 bytes)	OBD
	PID \$02 DTC that caused required freeze frame storage (2 bytes)	OBD
	PID \$1C OBD requirements to which vehicle or engine is certified (1 byte)	OBD
	PID \$21 distance traveled while MIL is activated (2 bytes)	OBD
	PID \$30 number of warm-up cycles (WUC) since DTCs cleared (1 byte)	OBD
	PID \$31 distance traveled since DTCs cleared (2 bytes)	OBD
	PID \$41 monitor status this driving cycle (4 bytes)	OBD
	PID \$4D engine run time while MIL is activated (2 bytes)	no
	PID \$4E engine run time since DTCs cleared (2 bytes)	no
	PID \$91 WWH-OBD ECU OBD system information (5 bytes)	no
OBD: readiness status	Readiness status	OBD
OBD: In-use monitor performance ratio (IUMPR)	In-use monitor performance ratio (IUMPR) Support	no
OBD for light duty (OBD2)	Destination of OBD events	OBD
	Service \$01 read current powertrain diagnostic data	OBD
	Service \$02 read powertrain freeze frame data	OBD
	Service \$04 clear reset emission-related diagnostic information	OBD
	Service \$06 support of central DTR handling	no
	Service \$07 read emission-related diagnostic trouble codes	OBD
	Service \$0A read emission-related diagnostic trouble codes with permanent status	OBD
WWH-OBD		no
J1939	Read DTC	yes



Feature set	Feature	Support
	J1939 Clear DTC	yes
	J1939 Freeze Frame DTC	yes
	Diagnostic Readiness 1	yes
Interfaces	Interaction with Diagnostic Communication Manager (Dcm):	
	▶ Access DTCs and status information (AUTOSAR 4.3.1)	yes
	▶ Access event-related data (AUTOSAR 4.3.1)	yes
	▶ Clear diagnostic information (AUTOSAR 4.3.1)	yes
	▶ Control DTC setting (AUTOSAR 4.3.1)	yes
	▶ Asynchronous Dcm operations (only for clear)	yes
	▶ Dem_GetFreezeFrameDataByRecord()	no
	▶ Dem_GetDTCByOccurrenceTime() (AUTOSAR 4.3.1)	yes
	Interaction with J1939 Diagnostic Communication Manager (J1939Dcm)	yes
	Interaction with Function Inhibition Manager (FiM)	yes
	Interaction with NVRAM Manager (NvM)	yes
OEM extensions for VCC ^a	Interaction with Development Error Tracer (Det)	yes
	Interaction with Diagnostic Log & Trace (Dlt)	no
	Event burst reduction <i>Configurable limitation of the number of events processed per main function cycle in Dem.</i>	OBD
	Enhanced readiness groups <i>Enhancement of readiness group support to fulfill SAE J1979 (2012), e.g. support for fuel system monitoring and CC monitoring.</i>	OBD
	Operation Cycle Counter 5 (OCC5) <i>Operation Cycle Counter 5 (OCC5) in Dem: Number of warm-up cycles since the DTC commanded the MIL to switch off (since DTC information was last cleared)</i>	OBD
	OBD aging for Dem <i>Aging takes operation cycle counter 2 (OCC2), operation cycle counter 5 (OCC5) DemOBDAgingCycleCounterThreshold and DemAgingCycleCounterThreshold into account</i>	OBD



Feature set	Feature	Support
	Dem extended FiM support for inhibition triggers, FID messages, FID symptom	OBD
Calibration	Calibration of selected data structures	OBD
Configuration	Support of pre-compile configuration	yes
	Support of post-build configuration	no
Variant management	Switchable parameters: indicator failure cycle counter threshold and OBD compliance	OBD
	Support of different variants of an ECU (different supported DTCs) in one configuration	no
	Side-allocation: change DTC value during runtime (with API)	yes
	Availability of events via post-build	no
BSW distribution	Support for BSW event reporting from different cores	yes

^aVolvo Car Corporation specific extension, according to SWRS_31835918-007-VCC AUTOSAR Dem

2.3.3. Supported FiM features

Feature set	Feature	Support
FiM core	Full implementation according to AUTOSAR 4.0.3 SWS	yes
Additional inhibition triggers ^a	FIM_LAST_UNCONFIRMED: SI30 UnconfirmedDTC is set	OBD
	FIM_UNCONFIRMED_THIS_OP_CYCLE: SI30 UnconfirmedDTCThisOperationCycle is set	OBD
FID messages ^a	Calculation of an error message status based on FID status	OBD
	Control of the error message status via a condition event	OBD
	Provision of C-API functions for BSW modules and Rte interface	OBD
	GetMessageStatus for reading the current error message status	OBD
FID symptom ^a	Symptom bit per FID that indicates that FID inhibition criteria apply	OBD
	Indication of symptom to Dem, based on FID status	OBD
	Control of the symptom indication via a condition event	OBD
FID forced release ^a	Optional release event per FID, which permits the FID on event status FAILED to be released to prevent dead-lock situations	OBD



Feature set	Feature	Support
Recovery failed event ^a	Optional recovery event per FID, which keeps the error message status of the FID enabled in FiM	OBD
Calibration	Calibration of selected FiM data structures	OBD
Configuration	Support of pre-compile configuration	yes
	Support of post-build configuration	no
Variant management	Support of different variants of an ECU in one configuration	no

^aVolvo Car Corporation specific extension, according to SWRS_31836996-005-VCC AUTOSAR FiM



3. ACG8 Diagnostic Stack release notes

3.1. Overview

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

3.2. Scope of the release

3.2.1. Configuration tool

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

- ▶ EB tresos Studio: 28.2.0 b211016-0103

3.2.2. AUTOSAR modules

The following table lists the AUTOSAR modules that are part of this ACG8 Diagnostic Stack release.

Module name	AUTOSAR version and revision	SWS version and revision	Module version	Supplier
Dcm	4.0.3 []	4.2.0 [3]	5.0.10	Elektrobit Automotive GmbH
Dem	4.0.3 []	4.2.0 [3]	6.4.3	Elektrobit Automotive GmbH
FiM	4.0.3 []	2.2.0 [3]	2.5.4	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. Dcm module release notes

- ▶ AUTOSAR R4.0 Rev 3
- ▶ AUTOSAR SWS document version: 4.2.0
- ▶ Module version: 5.0.10.B478580
- ▶ Supplier: Elektrobit Automotive GmbH

3.3.1.1. Change log

This chapter lists the changes between different versions.

Module version 5.0.10

2021-10-08

- ▶ ASCDCM-5668 Fixed known issue: The Dcm triggers the creation of redundant NvM blocks
- ▶ ASCDCM-5693 Fixed known issue: The Dcm notifies application with wrong confirmation type for RequestCorrectlyReceivedResponsePending (0x78) negative response code transmission

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



- ▶ ASCDCM-5743 Fixed known issue: The DCM returns positive response with no data to a related diagnostic DID service request if the requested DID has the DcmDspDataSize configuration parameter set to maximum value
- ▶ ASCDCM-5822 Fixed known issue: The Dcm triggers the deletion of created NvM blocks
- ▶ ASCDCM-5813 Fixed known issue: The Dcm returns a positive response with incomplete data to a related diagnostic DID service request if the requested DID has a variable signal that returns a data length of greater than 8192 bytes

Module version 5.0.9

2021-06-25

- ▶ ASCDCM-5641 Fixed known issue: The Dcm declares the user callouts for service InputOutputControlByIdentifier (0x2F) with an incorrect parameter name
- ▶ ASCDCM-5584 Fixed known issue: The Dcm transmits incorrect responses to requests for the UDS service ReadDTCInformation (0x19) with subfunction reportDTCSnapshotIdentification (0x03)
- ▶ ASCDCM-5619 Fixed known issue: Service RequestTransferExit (0x37) behaves unexpectedly when DcmDataTransferServicesASRVersion is set to 'AUTOSAR_43' or 'AUTOSAR_422'
- ▶ ASCDCM-5588 Fixed known issue: The Dcm does not process a service request when replying immediately with response pending (RCRRP) transmission
- ▶ ASCDCM-5575 Fixed known issue: The DCM blocks the protocol on which a related diagnostic DID service is requested if the configuration of DCM does not contain any DID signal
- ▶ ASCDCM-5501 Fixed known issue: Generation fails if call-back functions are used for a routine with configuration parameter DcmDspRoutineTidRef enabled
- ▶ ASCDCM-5578 Fixed known issue: The Dcm may become unresponsive after confirmation of response pending (RCRRP) transmission
- ▶ ASCDCM-5700 Fixed known issue: The Dcm might use incorrect Pdu MetaData information for TYPE2 periodic transmissions

Module version 5.0.8

2021-05-21

- ▶ ASCDCM-5475 Fixed known issue: Dcm suppresses the final response for an UDS request after RCRRP (0x78) response transmission
- ▶ ASCDCM-5524 Fixed known issue: Unnecessary routine APIs are generated if port interface is used for a routine with configuration parameter DcmDspRoutineTidRef enabled
- ▶ ASCDCM-5555 Out-of-bounds memory access may occur when the UDS service InputOutputControlByIdentifier (0x2F) is requested for a DID with no signals



- ▶ ASCDCM-5568 Fixed known issue: Compiler errors might occur when memory mapping is used
- ▶ Implemented UDS service `ReadScalingDataByIdentifier(0x24)`
- ▶ ASCDCM-5559 Fixed known issue: Out-of-bounds memory access may occur in case of a `TesterPresent` (0x3E) request or `BusyRepeatRequest` (0x21) NRC transmission when `DcmDslEnableTxConfirmationNotification` is set to true
- ▶ ASCDCM-5545 Fixed known issue: The Dcm does not clear `ApplUpdated` and `ResponseRequired` flags after a jump from Bootloader/ECUReset
- ▶ ASCDCM-5580 Fixed known issue: Compilation error occurs if the configuration of DCM does not contain any DID signal and service `WriteDataByIdentifier(0x2E)` is configured
- ▶ Updated Pdu metadata support for Dcm communication interfaces based on EcuC APIs with metadata routing to the lower layer.

Module version 5.0.7

2021-03-05

- ▶ ASCDCM-5440 Fixed known issue: Incorrect validity check for service `DynamicallyDefineDataIdentifier(0x2C)`
- ▶ ASCDCM-5429 Fixed known issue: The Dcm transmits NRC 0x11 for the OBD service 0x00 and the OBD services between 0x0B to 0x0F if these services are not supported
- ▶ ASCDCM-5440 Fixed known issue: Incorrect validity check for service `DynamicallyDefineDataIdentifier(0x2C)`
- ▶ ASCDCM-5419 Fixed known issue: Dcm transmits incorrect response after maximum number of RCRRP response transmissions for OBD service \$09 - Request vehicle information
- ▶ ASCDCM-5453 Fixed known issue: Dcm transmits an incorrect response for OBD service \$09 - Request vehicle information
- ▶ ASCDCM-5490 Fixed known issue: The Dcm does not handle configuration with infinite number of RCRRP (0x78) response pending transmissions correctly
- ▶ ASCDCM-5378 Fixed known issue: Undefined behavior might occur if the transmission of a `RequestCorrectlyReceivedResponsePending` negative response code is not confirmed
- ▶ Implemented Pdu metadata support for Dcm communication interfaces based on EcuC APIs without metadata routing to the lower layer.
- ▶ ASCDCM-5509 Fixed known issue: The Dcm makes a session transition and signals an application update after a jump from ECUReset

Module version 5.0.6

2021-02-12



- ▶ ASCDCM-5391 Fixed known issue: The Dcm ignores further requests after positive response to EcuReset (0x11) even for subfunctions which should not cause an actual reset
- ▶ Performance optimization regarding Dcm Endianess Conversion.
- ▶ ASCDCM-5413 Fixed known issue: The Dcm transmits negative response code RCRRP (0x78) for OBD services not supporting RCRRP negative response code
- ▶ ASCDCM-5464 Fixed known issue: The Dcm module does not compile if a DcmDslProtocolRow has the DcmDslProtocolID set to DCM_SUPPLIER_XX and Rte usage is enabled
- ▶ ASCDCM-5403 Fixed known issue: The Dcm does not transmit a final response if at least one ManufacturerNotification/SupplierNotification operation returns DCM_E_REQUEST_NOT_ACCEPTED after a RCRRP response transmission
- ▶ ASCDCM-5408 Fixed known issue: The Dcm does not transmit a final response if the buffer for the diagnostic response transmission becomes busy after a RequestCorrectlyReceivedResponsePending response transmission

Module version 5.0.5

2021-01-22

- ▶ ASCDCM-5206 Fixed known issue: Dcm does not ensure consistent event-related data when reading freeze frames or extended data records from a non-Elektrorit Dem
- ▶ Changed multiplicity of DcmDemClientRef to 0..1

Module version 5.0.4

2020-12-17

- ▶ ASCDCM-3918 Fixed known issue: The Dcm might not send a RequestCorrectlyReceivedResponsePending (0x7F SID 0x78) NegativeResponseCode in time when Protocol Preemption occurs
- ▶ ASCDCM-5313 Fixed known issue: Undefined behavior might occur if ResponseOnEvent or periodic transmissions are used
- ▶ ASCDCM-5293 Fixed known issue: The Dcm does not postpone the processing of a ROE event on one protocol while the Dcm is executing a request on another protocol
- ▶ ASCDCM-5301 Fixed known issue: Undefined behavior might occur upon incoming requests on different connections of the same protocol
- ▶ ASCDCM-5208 Fixed known issue: The Dcm module becomes unresponsive if ResponseOnEvent (0x86) external sub-service handler returns E_NOT_OK
- ▶ Added configuration warning "The sum of the DcmDspRoutineSignalPos and the DsmDspRoutineSignalLength when aligned to the next byte boundary shall be less than the buffer size used for the reception channel of the RoutineControl service configured."



- ▶ ASCDCM-5330 Fixed known issue: The Dcm may become unresponsive when a low priority protocol is preempted by a higher priority protocol
- ▶ ASCDCM-5318 Fixed known issue: The Dcm transmits an incorrect response for the UDS service TransferData (0x36)
- ▶ ASCDCM-5212 Fixed known issue: Undefined ECU behavior might occur if the UDS service ReadDataByIdentifier (0x22) is requested for a DID containing signals with SenderReceiver access
- ▶ ASCDCM-5147 Fixed known issue: Dcm provides wrong message context data to the Dcm_ApplicationTransmissionConfirmation() callout function for negative response messages BusyRepeatRequest (0x21) and ConditionsNotCorrect (0x22)
- ▶ Added configuration warning: "Protocols which have different types (OBD/UDS) and which belong to different stacks should not reference service tables with mixed OBD/UDS services, or the same service table".

Module version 5.0.3

2020-10-23

- ▶ ASCDCM-5295 Fixed known issue: The Dcm might reset to the default diagnostic session while processing an application-injected request in a non-default diagnostic session

Module version 5.0.2

2020-09-25

Module version 5.0.1

2020-08-28

- ▶ ASCDCM-5164 Fixed known issue: The Dcm module might not compile if the OBD services \$03, \$07, and \$0A are configured
- ▶ ASCDCM-5197 Fixed known issue: The ECU might reset or block when a service is requested that requires a subservice but subservices are not enabled
- ▶ ASCDCM-5155 Fixed known issue: The Dcm sends an incorrect response when two ROE events are triggered one after another
- ▶ ASCDCM-5213 Fixed known issue: Dcm does not compile with OBD service Clear/Reset emission-related diagnostic information (\$04) or UDS service ClearDiagnosticInformation (0x14) enabled and postcondition assertion checks disabled
- ▶ ASCDCM-5192 Fixed known issue: Undefined Dcm behavior due to release of unlocked exclusive area



Module version 5.0.0

2020-06-26

- ▶ Implemented OBD service Request Vehicle Information (\$09) according to AUTOSAR 4.4.0 specification

Module version 4.16.2

2020-05-22

- ▶ Parallel service processing - DemFacade rework
- ▶ Parallel UDS and OBD processing/ASR4.3 Dcm/Dem interface with ClientID
- ▶ ASCDCM-5067 Fixed known issue: Service TransferData (0x36) ignores the maxNumberOfBlockLength returned by the previous RequestUpload (0x35) service response
- ▶ Implemented support for 0xFF00 DID according to ISO-14229/2013.
- ▶ Omission of DTR data in service response when Dem_DcmGetDTRData() returns E_NOT_OK
- ▶ ASCDCM-5063 Fixed known issue: Service ReadDTCInformation (0x19) related macro is incorrectly generated
- ▶ Implemented OBD service Request Vehicle Information (\$09) according to AUTOSAR 4.4.0 specification

Module version 4.16.1

2020-04-24

- ▶ ASCDCM-4775 Fixed known issue: The Dcm still accepts further service requests after answering positively to service ECURest (0x11)
- ▶ ASCDCM-4977 Fixed known issue: Session switch does not succeed after positive OBD response
- ▶ ASCDCM-5027 Fixed known issue: Dcm provide an incomplete protection against double type definition of type Dcm_DidSupportedType
- ▶ ASCDCM-4964 Fixed known issue: Dcm transmits an incorrect response for OBD service \$09 - Request vehicle information

Module version 4.16.0

2020-03-25

Module version 4.15.2

2020-02-21



- ▶ ASCDCM-4742 Fixed known issue: Dcm transmits not supported NRCs requestCorrectlyReceivedResponsePending (0x78) and generalReject (0x10) for OBD services \$03, \$07, and \$0A when its service processing needs additional time to finish
- ▶ ASCDCM-4640 Fixed known issue: Compilation error occurs when service InputOutputControlByIdentifier (0x2F) is configured and no DIDs are present
- ▶ ASCDCM-4641 Fixed known issue: The Dcm can enter a non-default session while processing an OBD request and perform another unexpected reset to the Default Session
- ▶ ASCDCM-4855 Fixed known issue: The DCM blocks the protocol on which a related diagnostic DID service is requested after a cancellation of an InputOutputControlByIdentifier (0x2F) service request

Module version 4.15.1

2020-01-24

- ▶ ASCDCM-4710 Fixed known issue: Compiler errors occur when no DcmDslSecurityLevels are configured
- ▶ ASCDCM-4709 Fixed known issue: The DCM is not able to receive requests after protocol preemption
- ▶ ASCDCM-4717 Fixed known issue: Compilation error occurs when OBD service Request Control of On-Board System, Test or Component (0x08) is configured and no Request Control configuration is present
- ▶ Restrict internal support for subservices 0x03 and 0x43 belonging to ResponseOnEvent (0x86) from configuration
- ▶ ASCDCM-4720 Fixed known issue: The Dcm does not reset the security level when starting a protocol.
- ▶ Implemented sub-services reportNumberOfDTCBySeverityMaskRecord(0x07), reportDTCBySeverityMaskRecord(0x08), reportSeverityInformationOfDTC(0x09) for UDS Service Read-DTCInformation (0x19) according to AUTOSAR 4.4.0 specifications.
- ▶ ASCDCM-4748 Fixed known issue: Dcm transmits an incorrect response for OBD service \$04 Clear/reset emission-related diagnostic information
- ▶ ASCDCM-4719 Fixed known issue: The Dcm may become unresponsive as a result of protocol preemption
- ▶ ASCDCM-4777 Fixed known issue: Out of bounds memory access occurs if the services WriteMemoryByAddress (0x3D), RequestDownload (0x34), and RequestUpload (0x35) are used

Module version 4.15.0

2019-12-13

- ▶ Parallel UDS and OBD processing/ASR4.3 Dcm/Dem interface with ClientID

Module version 4.14.23

2019-12-06



- ▶ ASCDCM-4667 Fixed known issue: Compilation error occurs when service ReadDataByPeriodicIdentifier(0x2A) is configured and no DIDs are present

Module version 4.14.22

2019-10-25

- ▶ ASCDCM-4577 Fixed known issue: The Dcm transmits an incorrect response for UDS service Transfer-Data (0x36) requests with repeated blockSequenceCounter
- ▶ ASCDCM-4626 Fixed known issue: Dcm module Generation fails when no Read Memory Ranges are defined and DcmDspMemoryRangeHighNotIncluded parameter is set to TRUE
- ▶ ASCDCM-3544 Implemented Generic End Of Line addon for Data Identifiers for Service ReadDataByIdentifier (0x22), Service WriteDataByIdentifier (0x2E) and InputOutputControlByIdentifier (0x2F)
- ▶ ASCDCM-4572 Fixed known issue: Dcm sends an incorrect positive response if paged-buffer mechanism is used
- ▶ ASCDCM-4716 Fixed known issue: Out of bounds memory access occurs for OBD services \$03, \$07 and \$0A

Module version 4.14.21

2019-09-06

- ▶ ASCDCM-4512 Fixed known issue: Signal data corruption occurs when processing a periodic transmission in parallel with a DID read operation
- ▶ ASCDCM-4514 Fixed known issue: Null pointer dereference occurs when the ReadDataByPeriodicIdentifier (0x2A) service is executed
- ▶ ASCDCM-4509 Fixed known issue: The Dcm transmits an incorrect response when the UDS service Write-DataByIdentifier (0x2E) is requested for not supported DIDs in the OBD range
- ▶ ASCDCM-4521 Fixed known issue: The Dcm calls the ReturnControlToEcu asynchronous operation with an invalid OpStatus

Module version 4.14.20

2019-08-09

- ▶ ASCDCM-4470 Fixed known issue: Requests received shortly after a positive response caused by a WARM_START condition may be rejected
- ▶ ASCDCM-4477 Fixed known issue: RequestTransferExit behaves unexpected when DcmDataTransfer-ServicesASRVersion is set to 'AUTOSAR_43' or 'AUTOSAR_422'



- ▶ ASCDCM-4478 Fixed known issue: Dcm does not generate if its default service API is different from Dem's default service API
- ▶ ASCDCM-4480 Fixed known issue: Dcm does not generate when DcmConfigSet default name (DcmConfigSet_0) is changed

Module version 4.14.19

2019-07-12

- ▶ ASCDCM-4149 Fixed known issue: Dcm does not generate A2L files
- ▶ ASCDCM-4469 Fixed known issue: Dem_DcmGetAvailableOBDMIDs() function is called also for non-availability OBDMIDs

Module version 4.14.18

2019-06-17

- ▶ Increased maximum number of RequestCorrectlyReceivedResponsePending (0x78) NRCs
- ▶ ASCDCM-4355 Fixed known issue: Reading DID ranges may overrun the response message buffer
- ▶ ASCDCM-4367 Fixed known issue: TYPE1 periodic responses keep the Dcm in a non-default session
- ▶ ASCDCM-4379 Fixed known issue: The Dcm stays in a non-default session indefinitely when ResponseOnEvent or PeriodicDID transmissions are used
- ▶ Implemented array type signals for the UDS service RoutineControl (0x31) according to AUTOSAR 4.3.0
- ▶ Implement Generic End Of Line handling for Routine Control service (0x31) when an unsupported (unconfigured / unused) routine is requested
- ▶ Implement Routine Info Byte handling for Routine Control service (0x31) when a byte from the application is requested to be added on the response
- ▶ ASCDCM-4401 Fixed known issue: In `Dcm_CommunicationServices.c` Incorrect response data is transmitted if buffers over 65535 bytes in length are used
- ▶ ASCDCM-4402 Fixed known issue: In `Dcm_CommunicationServices.c` Incorrect request data is received if buffers over 65535 bytes in length are used

Module version 4.14.17

2019-05-24

- ▶ Added support for configuration parameter `DcmDspDidControlMask`. Values 1, 2, 3 and 4 are supported for parameter `DcmDspDidControlMaskSize`. The value `DCM_CONTROLMASK_EXTERNAL` is supported for parameter `DcmDspDidControlMask`. Values configured for `DcmDspDidControlMask` and `DcmDspDidControlMaskSize` have effect only upon the interface signatures of `DcmDspData` elements refer-



enced by the `DcmDspDid` which references the `DcmDspDidInfo` containing the previously-mentioned `DcmDspDidControlMask` and `DcmDspDidControlMaskSize` parameters

- ▶ Implemented generic interface to inject service requests from application software

Module version 4.14.16

2019-05-16

- ▶ ASCDCM-4360 Fixed known issue: Compiler abstraction in declaration of `Xxx_WriteData` and `Xxx_ShortTermAdjustment` interface functions

Module version 4.14.15

2019-04-18

- ▶ ASCDCM-4217 Fixed known issue: Requests for the `ShortTermAdjustment` (0x03) operation of service `InputOutputControlByIdentifier` (0x2F) are not handled correctly
- ▶ ASCDCM-4241 Fixed known issue: Missing AUTOSAR compiler abstraction for pointers defined in type definitions or defined as function parameters

Module version 4.14.14

2019-04-11

- ▶ ASCDCM-4274 Fixed known issue: Invalid comments in parameter documentation
- ▶ ASCDCM-4229 Fixed known issue: Rte client/server interface operation call is executed from the context of the `Dcm_TpRxIndication()` API

Module version 4.14.13

2019-03-22

- ▶ ASCDCM-4232 Fixed known issue: Undefined behavior when using Sender/Receiver interfaces for `DcmDspData` DID signals
- ▶ Project specific service callbacks interfaces ASR 4.2.2 compliant for `WriteMemory` (0x27) and `ProcessRequestTransferExit` (0x32)

Module version 4.14.12

2019-02-18

- ▶ ASCDCM-4183 Fixed known issue: Compilation fails if `DcmDspMemoryRangeHighNotIncluded` is set to true



- ▶ ASCDCM-4204 Fixed known issue: Compiler errors occur when a DcmDspWriteMemoryRangeByLabelInfo is configured and no DcmDspWriteMemoryRangeInfo and no DcmDspReadMemoryRangeByLabelInfo are configured
- ▶ ASCDCM-4214 Fixed known issue: Undefined behavior might occur when a request for the UDS service TransferData (0x36) is received
- ▶ ASCDCM-4216 Fixed known issue: CONST variables of RoutineControl (0x31) service implementation are not defined in an appropriate MemMap section

Module version 4.14.11

2019-01-24

- ▶ ASCDCM-4098 Fixed known issue: The Dcm sends a positive response if subfunction clearDynamicallyDefinedDataIdentifier (0x03) of service DynamicallyDefineDataIdentifier (0x2C) is requested for an un-configured dynamically defined DID
- ▶ ASCDCM-4145 Fixed known issue: Dcm accesses wrong data when DynamicallyDefineDataIdentifier service is requested with a DID that is outside the dynamic range
- ▶ ASCDCM-4102 Fixed known issue: The Dcm may become unresponsive if the application requests a reset to the default session too often
- ▶ ASCDCM-4175 Fixed known issue: Periodic DIDs are not transmitted in the same order in which they were configured
- ▶ Enable DCM internal service processing in case of project specific service handling
- ▶ Implementation of selectable NRC behaviour for unsupported OBD services.

Module version 4.14.10

2018-11-23

- ▶ ASCDCM-4025 Fixed known issue: Compiler errors occur when a DcmDslRoeConnection is configured and referenced, and the internally-managed ResponseOnEvent (0x86) service handler is not used
- ▶ ASCDCM-4142 Fixed known issue: A code generation error occurs if DcmDspWriteMemoryRangeInfo or DcmDspWriteMemoryRangeByLabelInfo are configured and neither DcmDspReadMemoryRangeInfo nor DcmDspReadMemoryRangeByLabelInfo are configured

Module version 4.14.9

2018-10-26

- ▶ Implement support for configuration parameter DcmDspMemoryRangeHighNotIncluded to allow for more flexibility in the definition of memory ranges



- ▶ Implemented storage of output signals data between service handler calls and improved memory usage for UDS service RoutineControl (0x31)
- ▶ ASCDCM-4041 Fixed known issue: The ECU may reset during the transmission of a RequestCorrectlyReceivedResponsePending (0x78) NegativeResponseCode when the DiagnosticSessionControl (0x10) service is used with a session marked as either DCM_OEM_BOOT or DCM_SYS_BOOT
- ▶ ASCDCM-4039 Fixed known issue: Externally managed OBD service "Request on-board monitoring test results for specific monitored systems" (\$06) causes compile error
- ▶ TxConfirmation callout with information for positive response suppression (ID10)

Module version 4.14.8

2018-09-28

- ▶ ASCDCM-4021 Fixed known issue: Wrong mode declaration string RTE_MODE_DcmEcuReset_KEYOFFON causes compile error
- ▶ ASCDCM-3998 Fixed known issue: Missing AdjacentLayer.properties information for TYPE2 ROE connections
- ▶ ASCDCM-4010 Fixed known issue: When DiagnosticSessionControl (0x10) service is used with a boot session the message "0x50" is transmitted incorrectly instead of a positive response
- ▶ ASCDCM-4028 Fixed known issue: Dcm does not compile if service RequestVehicleInformation (\$09) is enabled without configuration data
- ▶ Notification callout on S3Timeout (ID7)

Module version 4.14.7

2018-08-24

- ▶ Add support for short response on sub-function 0x01 - onDTCStatusChange for the UDS service ResponseOnEvent (0x86)
- ▶ ASCDCM-3980 Fixed known issue: TesterPresent request with the suppressPosRspMsgIndicationBit set (0x3E 0x80) on a physical channel may result in a positive response
- ▶ ASCDCM-3950 Fixed known issue: The Dcm does not transmit a negative response for subsequent UDS service TransferData (0x36) requests which are not consistent with the UDS service RequestDownload (0x34) service requests

Module version 4.14.6

2018-07-27

- ▶ ASCDCM-3944 Fixed known issue: Undefined Dcm behavior when endianness conversion is enabled



- ▶ ASCDCM-3897 Fixed known issue: The Dcm becomes unresponsive when `Dcm_Init()` is called before `Rte_Start()`

Module version 4.14.5

2018-06-22

- ▶ ASCDCM-4188 Fixed known issue: Function declarations of TesterPresent (0x3E) service implementation are not inside a CODE section
- ▶ ASCDCM-4189 Fixed known issue: CONST variables of service request notification implementation are not declared in an appropriate MemMap section
- ▶ ASCDCM-4190 Fixed known issue: Definition of local pointers without AUTOSAR compiler abstraction
- ▶ ASCDCM-4191 Fixed known issue: Variable of RoutineControl (0x31) service implementation is not declared and defined in an appropriate MemMap section
- ▶ Implement usage of the Handheld wizard for `DcmDslProtocol` containers
- ▶ Support for BSW Distribution - Dcm and ComM on different partitions (cores)

Module version 4.14.4

2018-05-25

- ▶ ASCDCM-3867 Fixed known issue: Compilation errors are generated when DET is enabled and services `WriteMemoryByAddress` (0x3D) or `RequestDownload` (0x34) are configured, but `ReadMemoryByAddress` (0x23), `DynamicallyDefineDataIdentifier` (0x2C) or `RequestUpload` (0x35) are not configured

Module version 4.14.3

2018-05-07

- ▶ Implemented Sender/Receiver access for `DcmDspData` for UDS services `ReadDataByIdentifier` (0x22) and `WriteDataByIdentifier` (0x2E).
- ▶ Implemented endianness interpretation for `DcmDspData` with Sender/Receiver access for UDS services `ReadDataByIdentifier` (0x22) and `WriteDataByIdentifier` (0x2E) according to AUTOSAR 4.3 specifications
- ▶ ASCDCM-3665 Fixed known issue: Out-of-bounds memory access may occur when the UDS service `ReadDataByIdentifier` (0x22) is requested for a DID with signals with sender/receiver access

Module version 4.14.2

2018-04-20

- ▶ ASCDCM-3767 Fixed known issue: An already busy TYPE2 periodic transmission protocol accepts additional requests - processing timeout possible



- ▶ ASCDCM-3774 Fixed known issue: The TYPE2 periodic transmission protocol is blocked when the RduR rejects the periodic transmission

Module version 4.14.1

2018-03-19

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

Module version 4.14.0

2018-03-16

- ▶ ASCDCM-3693 Fixed known issue: Inconsistency between calls to ComM_DCM_ActiveDiagnostic() and ComM_DCM_InactiveDiagnostic() when more than one RxPduID of one protocol is used
- ▶ ASCDCM-3730 Fixed known issue: Memory section conflict between declaration and definition.
- ▶ ASCDCM-3686 Fixed known issue: Incorrect values may be sent for Periodic DIDs
- ▶ Memory range configuration of Services 0x23/0x3D with labels
- ▶ Added memory sections for the DCM software component description

Module version 4.13.2

2018-02-23

- ▶ ASCDCM-3715 Fixed known issue: Invalid type references generated when DID Ranges are used with Rte
- ▶ ASCDCM-3692 Fixed known issue: The ECU becomes unresponsive when the CommunicationControl (0x28) service is configured and the DiagnosticSessionControl (0x10) service is requested
- ▶ ASCDCM-3714 Fixed known issue: The Dcm might not compile if OBD services are enabled
- ▶ ASCDCM-3724 Fixed known issue: Dcm does not link if variable Dcm_BootloaderRequestToRespondTo is not assigned to RAM memory section
- ▶ ASCDCM-3694 Fixed known issue: The Dcm performs actions pertaining to a reset to the Default Session outside of normal reset-to-Default-Session circumstances
- ▶ ASCDCM-3721 Fixed known issue: Dcm wrongly resets the current communication status when the DiagnosticSessionControl (0x10) UDS service is requested
- ▶ ASCDCM-3725 Fixed known issue: DCM enable DTCSetting when the DiagnosticSessionControl (0x10) service is requested
- ▶ ASCDCM-3702 Fixed known issue: Rte generation error when asynchronous service execution is enabled for the RoutineControl (0x31) service



- ▶ ASCDCM-3706 Fixed known issue: The Dcm executes an unexpected reset to the Default Session
- ▶ ASCDCM-3708 Fixed known issue: The S3 timer is not restarted upon reception of a concurrent Tester-Present (0x3E 0x80) request
- ▶ ASCDCM-3709 Fixed known issue: A Precondition Assert is triggered in the conditions in which a BusyRepeatRequest (0x21) NegativeResponseCode is transmitted

Module version 4.13.1

2017-12-15

- ▶ ASCDCM-3618 Fixed known issue: TYPE1 Periodic DID transmissions may not be possible from a protocol that previously received a request which resulted in a suppressed transmission
- ▶ ASCDCM-3633 Fixed known issue: The XPath checker displays an incorrect warning message for the entries of DcmModeRule container
- ▶ ASCDCM-3626 Fixed known issue: Handle unexpected values returned by the Xxx_WriteDidData interface function.
- ▶ ASCDCM-3620 Fixed known issue: Return value E_NOT_OK of DID range read operation Xxx_ReadDidData() is not handled correctly
- ▶ ASCDCM-3631 Fixed known issue: Unsorted DID ranges are not correctly processed
- ▶ ASCDCM-3636 Fixed known issue: Incorrect application of the controlEnableMaskRecord in InputOutputControlByIdentifier (0x2F) requests
- ▶ ASCDCM-3635 Fixed known issue: DID Range Read operation Xxx_ReadDidData() is invoked with an incorrect value of the OpStatus parameter
- ▶ ASCDCM-3619 Fixed known issue: Reentrancy of the Xxx_ReadDataLength() and Xxx_ConditionCheckRead() operations if the InputOutputControlByIdentifier (0x2F) service is handled asynchronously and is requested at the same time as a periodic DID readout times out

Module version 4.13.0

2017-11-17

- ▶ ASCDCM-3540 Fixed known issue: TYPE1 Periodic DID transmissions are not possible from a protocol which has returned an NRC to a previous service request processing
- ▶ ASCDCM-3527 Fixed known issue: TYPE1 ROE transmissions on a different protocol than the configuring protocol do not change the session to the default session
- ▶ ASCDCM-3526 Fixed known issue: TYPE1 Periodic DID transmissions on a different protocol than the configuring protocol may occur and this does not change the session to the default session
- ▶ ASCDCM-1592 Fixed known issue: Endianness conversion is performed regardless of interface type for DcmDspData signals when reading, writing, or controlling DIDs. This is fulfilled by allowing the user to



select or deselect endianness conversion to be performed only on Routine signals, only on DID signals, or both.

- ▶ ASCDCM-3528 Fixed known issue: A precondition assert is reached when TYPE1 periodic DID transmissions or TYPE1 ResponseOnEvent (0x86) transmissions are used and XXX_StartProtocol() operations fail
- ▶ ASCDCM-3606 Fixed known issue: The Dcm transmits a wrong response when ReadMemoryByAddress (0x23) and WriteMemoryByAddress (0x3D) are requested with memoryAddress and memorySize equal to 0

Module version 4.12.19

2017-10-04

- ▶ Removed AUTOSAR 3.x compliant symbolic name value macros and updated the logic to only provide AUTOSAR 4.0.2 compliant macros if macro DCM_PROVIDE_LEGACY_SYMBOLIC_NAMES is defined
- ▶ ASCDCM-3488 Fixed known issue: The Dcm accepts requests on RxPduIDs belonging to a MainConnection on which a BusyRepeatRequest (0x21) transmission is ongoing
- ▶ Optimized speed for configuration verifier checks for DID and PID configuration.
- ▶ ASCDCM-3503 Fixed known issue: Dcm generation fails when a DID signal has variable length and does not have the last position in the DID's DcmDspDidSignal list
- ▶ ASCDCM-3500 Fixed known issue: NegativeResponseCodes are incorrectly sent instead of periodic DIDs when the `ReadDataByPeriodicIdentifier` (0x2A) service is used
- ▶ ASCDCM-3523 Fixed known issue: DCM does not generate a link to subfunction handler for subservice 0x00 for `TesterPresent` (0x3E) service
- ▶ ASCDCM-3535 Fixed known issue: A compiler warning "unused parameter" is generated if all DcmDslPeriodicTransmissions lack DcmDslPeriodicConnection entries
- ▶ ASCDCM-3502 Fixed known issue: Dcm generates ports for DID range operations that are not required
- ▶ ASCDCM-3515 Fixed known issue: Reentrancy of the `Xxx_ReadDataLength()` and `Xxx_ConditionCheckRead()` operations if the `ReadDataByIdentifier` (0x22) service is handled asynchronously and is requested at the same time as a periodic DID readout times out
- ▶ ASCDCM-3533 Fixed known issue: Incorrect responses when EndiannessConversion is used and a DID is read

Module version 4.12.18

2017-08-25

- ▶ Provided forward-declarations in `Dcm_Cbk.h` for `Dcm_ComM_NoComModeEntered()`, `Dcm_ComM_SilentComModeEntered()` and `Dcm_ComM_FullComModeEntered()` as per AUTOSAR 4.2 requirement SWS_Dcm_01066



- ▶ ASCDCM-3470 Fixed known issue: Incorrect signature of operation RequestResults of ClientServerInterface RoutineServices_RoutineName when an AUTOSAR 4.0.3 interface is used
- ▶ ASCDCM-3492 Fixed known issue: Compilation errors and warnings occur when attempt counter is set to reset after timeout but is not configured to be handled externally
- ▶ Allow configuration of external subfunction handlers for UDS service TesterPresent (0x3E). Adapted NRC sequence of UDS service TesterPresent (0x3E) to ISO 14229-1:2013 requirements.
- ▶ ASCDCM-3479 Fixed known issue: The Dcm may transmit an incorrect response to a SecurityAccess (0x27) request

Module version 4.12.17

2017-07-28

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

Module version 4.12.16

2017-07-21

- ▶ ASCDCM-3433 Fixed known issue: Malformed positive response to service ECURest (0x11) upon reset.
- ▶ Implemented support for security access delay timers according to AUTOSAR 4.3
- ▶ ASCDCM-4773 Fixed known issue: The AttemptCounter of a security level is reset and SecurityAccess(0x27) request sequences can be made without DcmDspSecurityDelayTime time-out to pass

Module version 4.12.15

2017-06-30

- ▶ Implemented subfunctions enableRxAndDisableTxWithEnhancedAddressInformation (0x04) and enableRxAndTxWithEnhancedAddressInformation (0x05) of service CommunicationControl (0x28).

Module version 4.12.14

2017-06-02

- ▶ ASCDCM-3372 Fixed known issue: DCM does not store the correct state of communication modes related to ComMChannels if service CommunicationControl (0x28) is called and the mode request is rejected by BswM



- ▶ ComM Channels referenced in DcmDslProtocolRx are reset to the default communication mode when the DcmDspComControlCommunicationReEnableModeRuleRef is not fulfilled anymore and no DcmDspComControlSpecificChannel and DcmDspComControlAllChannel are configured.
- ▶ Improved generation of swcd and template code when using a huge amount of DID Data. This leads to a drastic decrease of generation time.
- ▶ ASCDCM-3092 Fixed known issue: Dcm module might block or behave unexpectedly if concurrent requests are received from two or more clients under rare timing conditions.

Module version 4.12.13

2017-04-28

- ▶ ASCDCM-3295 Fixed known issue: Out-of-bounds memory access occurs when using EndiannessConversion and performing a ReadDataByIdentifier (0x22) request
- ▶ ASCDCM-3239 Fixed known issue: The ECU may block or reset if the UDS service CommunicationControl (0x28) is requested for a ComMChannel which is not referenced by any DcmDslProtocolRx-ComMChannelRef which belongs to any RxConnection
- ▶ ASCDCM-3307 Fixed known issue: Wrong responses or NvM Block data corruption when OBD services Request current powertrain diagnostic data (\$01) or Request powertrain freeze frame data (\$02) are used and DIDs which contain NvM BlockID signals are configured
- ▶ ASCDCM-3349 Fixed known issue: Inconsistency between advertised state and internal state of communication control when UDS service CommunicationControl (0x28) is processed asynchronously and the mode rule for communication control reset fails
- ▶ Allowed configuration of DIDs within the 0x0000 - 0x00FF range. Allowed the usage of services ReadDataByIdentifier (0x22) and WriteDataByIdentifier (0x2E) for DIDs in this range.

Module version 4.12.12

2017-03-31

- ▶ ASCDCM-3298 Fixed known issue: Dcm might not compile if DcmDspDids are configured but for none of them at least one signal DcmDspDidSignal is configured
- ▶ ASCDCM-3303 Fixed known issue: Undefined behavior occurs when the ComM module changes the mode of a ComM channel at the same time that a Periodic DID that was configured on an RxPduID which references the same ComM channel times out
- ▶ ASCDCM-3185 Fixed known issue: Dcm does not obey configured P2* timing behavior and ignores the maximum number of sent RequestCorrectlyReceivedResponsePending (0x78) NRCs
- ▶ Added new configuration parameter DcmDspDiagSesRespMaxNumRespPend. This offers more flexibility as it allows individual configuration of the maximum allowed number of RequestCorrectlyReceivedResponsePending (0x78) NRCs per request for each individual diagnostic session.



Module version 4.12.11

2017-03-22

- ▶ ASCDCM-3254 Fixed known issue: The Dcm does not compile if UDS service RequestUpload (0x35) is configured and internally-managed and neither UDS service RequestDownload (0x36) nor Write-MemoryByAddress (0x3D) are configured and internally-managed
- ▶ ASCDCM-3286 Fixed known issue: Dcm might not compile if the UDS service DynamicallyDefine-DataIdentifier (0x2C) is configured and internally-managed
- ▶ ASCDCM-3290 Fixed known issue: Undefined behavior if the callout Dcm_ReadMemory() returns DCM_-READ_FORCE_RCRRP when dynamically defined DIDs are read using the UDS service ReadDataByI-dentifier (0x22)
- ▶ ASCDCM-3310 Fixed known issue: Dcm might not compile if the UDS service RequestUpload (0x34) is configured and internally-managed
- ▶ ASCDCM-3287 Fixed known issue: Incoming receptions are cancelled upon completion of a reception which can be processed
- ▶ ASCDCM-3294 Fixed known issue: The Dcm may become unresponsive as a result of protocol preemption
- ▶ ASCDCM-3306 Fixed known issue: Requests for the ReadDTCInformation (0x19) service with subfunctions reportFirstTestFailedDTC (0x0B), reportFirstConfirmedDTC (0x0C), reportMostRecentTest-FailedDTC (0x0D), or reportMostRecentConfirmedDTC (0x0E) with an incorrect length are accepted

Module version 4.12.10

2017-03-03

- ▶ ASCDCM-3257 Fixed known issue: NullPointerException when a request for an externally-managed DiagnosticSessionControl (0x10) service handler which contains just the ServiceID byte is re-ceived
- ▶ Enabled the UDS service WriteDataByIdentifier (0x22) to write DIDs in the OBD DIDs interval 0xF400-0xF8FF
- ▶ ASCDCM-3282 Fixed known issue: Service Read Generic Information (0xAF) produces wrong NRC when request exceeds required length
- ▶ ASCDCM-3281 Fixed known issue: Dcm might not stop the S3 timer timeout even though diagnostic processing is ongoing
- ▶ ASCDCM-3256 Fixed known issue: Compiler warning or error due to signature mismatch of Dcm_RxIndi-cation() API
- ▶ Implemented the following services according to AUTOSAR 4.3.0.
 - ▶ RequestDownload (0x34)
 - ▶ RequestUpload (0x35)
 - ▶ TransferData (0x36)



- ▶ RequestTransferExit (0x37)
- ▶ ReadMemoryByAddress (0x23)
- ▶ WriteMemoryByAddress (0x3D)

Module version 4.12.9

2017-02-03

- ▶ ASCDCM-3186 Fixed known issue: A compilation error occurs due to mismatching memory mapping sections on the definition and the declaration of a data structure on the DCM
- ▶ ASCDCM-3214 Fixed known issue: Compiler errors are reported if service `ReadDataByPeriodicIdentifier` (0x2A) is configured, `DcmDspDDIDCheckPerSourceDID` equals "true" and service `DynamicallyDefineDataIdentifier` (0x2C) is not configured
- ▶ ASCDCM-3226 Fixed known issue: A code generation error occurs if `DcmModeDeclarationSupport` is set to true and the `DcmDspSession` container contains more than one entry
- ▶ For periodic DID (0xF200..0xF2FF) allow the access types 'write' and 'I/O control', additional to the existing 'read'.

Module version 4.12.8

2017-01-05

- ▶ ASCDCM-3180 Fixed known issue: Dcm writes out of bounds and rejects requests after a jump from bootloader with `ResponseRequired` set to FALSE

Module version 4.12.7

2016-12-14

- ▶ ASCDCM-3129 Fixed known issue: An unreachable code assert occurs if OBD services 0x03, 0x07 or 0x0A are used
- ▶ The NRC handling for `RoutineControl(0x31)` service is made according with selected ISO(ISO_14229-2006/ISO_14229-2006).
- ▶ ASCDCM-3141 Fixed known issue: Events set with service `ResponseOnEvent` (0x86) to be persistently in the Stopped State with sub-service `stopResponseOnEvent` and the `StorageState` set to "storeEvent" (0x40), are erroneously re-started after an ECU reset and a Session Change.
- ▶ ASCDCM-3124 Fixed known issue: Incorrect responses to requests of service `ReadDTCInformation` (0x19) as a result of multiple calls to the `Dem_GetNumberOfFilteredDTC()` API
- ▶ ASCDCM-3155 Fixed known issue: Periodic transmission packets may be lost after removing or reconfiguring periodic DIDs



- ▶ ASCDCM-3164 Fixed known issue: The DCM produces redundant DET error notifications for channels which are not configured for the DCM

Module version 4.12.6

2016-11-04

- ▶ Changed all defensive programming configuration and functionality to `Dcm_DefProg_Cfg.h`
- ▶ ASCDCM-3064 Fixed known issue: Incorrectly named ports are generated for the `ServiceRequestNotification ClientServerInterface`
- ▶ Implemented subfunction `reportFirstTestFailedDTC (0x0B)` of UDS service `ReadDTCInformation (0x19)`
- ▶ Implemented subfunction `reportFirstConfirmedDTC (0x0C)` of UDS service `ReadDTCInformation (0x19)`
- ▶ Implemented subfunction `reportMostRecentTestFailedDTC (0x0D)` of UDS service `ReadDTCInformation (0x19)`
- ▶ Implemented subfunction `reportMostRecentConfirmedDTC (0x0E)` of UDS service `ReadDTCInformation (0x19)`
- ▶ ASCDCM-3085 Fixed known issue: If a request of service `WriteDataByIdentifier (0x2E)` is performed and `DcmDslProtocolEndiannessConvEnabled` is set to true, write data in the DID may be corrupted
- ▶ ASCDCM-3079 Fixed known issue: Asynchronous `ReadDataLength` operations are always called with a `DCM_INITIAL OpStatus` when an AUTOSAR 4.2.1 interface is used
- ▶ ASCDCM-3080 Fixed known issue: Compiler error if OBD service `RequestOnboardMonitoringTestResults (0x06)` is enabled and `DcmOBDMIDSupportType` is configured to `DCM_OBD-MID_SUPPORT_DEM`
- ▶ Implemented TYPE2 transmission for UDS service `ResponseOnEvent (0x86)`.
- ▶ ASCDCM-3090 Fixed known issue: Unexpected behavior due to reentrancy of `Dem_EnableDTCSetting`
- ▶ ASCDCM-3051 Fixed known issue: The `Dem_EnableDTCSetting()` API is called on diagnostic session transition / referenced ModeRule failure even if DTCSetting is already enabled

Module version 4.12.5

2016-10-07

- ▶ Adapted resource file for the scheduling of main functions to the split of `IpduM_MainFunction()` into `IpduM_MainFunctionRx()` and `IpduM_MainFunctionTx()`.
- ▶ ASCDCM-3036 Fixed known issue: UDS service `ReadDataByIdentifier (0x22)` does not handle requests for OBD availability MIDs/PIDs/VehInfos correctly



- ▶ ASCDCM-3005 Fixed known issue: Compiler errors and warnings are issued when service `ReadGenericInformation` (0xAF) is configured and service `ReadDTCInformation` (0x19) is not configured.
- ▶ Implemented subfunction `OnDTCStatusChange` (0x01) of service `ResponseOnEvent` (0x86) and related functionality.
- ▶ ASCDCM-3054 Fixed known issue: Dcm does not respond correctly when OBD Service 0x01 is enabled and PID 0x00 is requested using `ReadDataByIdentifier` (0x22) service.

Module version 4.12.4

2016-09-09

- ▶ ASCDCM-2998 Fixed known issue: An out-of-bounds memory access occurs if the same `DcmDspSubnetNumber` is configured twice and is used in a request for service `CommunicationControl` (0x28)
- ▶ Implemented subfunction `ClearResponseOnevent` (0x06) of service `ResponseOnEvent` (0x86)
- ▶ ASCDCM-2989 Fixed known issue: Incorrect security level checks and calls to application functions are performed upon protocol preemption
- ▶ ASCDCM-3010 Fixed known issue: Service handlers may be called with a `DCM_CANCEL` OpStatus after completion of operation
- ▶ Improved NRC handling for `RequestDownload` (0x34) service.
- ▶ Improved NRC handling for `RequestUpload` (0x35) service.
- ▶ Improved NRC handling for `TransferData` (0x36) service.
- ▶ Improved NRC handling for `RequestTransferExit` (0x37) service.
- ▶ ASCDCM-3014 Fixed known issue: Buffer corruption and undefined behavior may occur as a consequence of the transmitting protocol being stopped prematurely in the case of TYPE2 periodic transmissions
- ▶ ASCDCM-3005 Fixed known issue: The `ComMChannel` on which a diagnostic request is received might not go to or remain in `FullCommunicationMode`

Module version 4.12.3

2016-08-05

- ▶ ASCDCM-2407 Fixed known issue: Cancellation of pending receptions is not done correctly
- ▶ Configuration Signature
- ▶ Implemented full support for `eventWindowTime` for service `ResponseOnEvent` (0x86)
- ▶ Diagnostic session checks are performed on subfunctions of service `DiagnosticSessionControl` too

Module version 4.12.2

2016-07-22



- ▶ ASCDCM-2876 Fixed known issue: `Dcm_Cfg.h` include many spaces in one line
- ▶ ASCDCM-2870 Fixed known issue: ROE preconfiguration data is not initialized correctly and/or memory corruption occurs when persistent ROE settings are used
- ▶ ASCDCM-2837 Fixed known issue: Compiler errors occur when the EB Dcm is integrated with other vendor's basic software modules
- ▶ ASCDCM-2867 Fixed known issue: ROE events are wrongly re-activated when re-entering the default session when they are stopped outside the default session
- ▶ ASCDCM-2959 Fixed known issue: ROE events triggered outside FullCommunicationMode are wrongly queued and processed later for transmission

Module version 4.12.1

2016-07-01

- ▶ Implemented a subset of UDS service 0x86 (ResponseOnEvent) according to AUTOSAR 4.2 specification. The implemented subfunctions are: 0x00 stopResponseOnEvent, 0x04 reportActivatedEvents, 0x05 startResponseOnEvent for preconfigured events of OnChangeOfDataIdentifier type with Infinite(0x02) eventWindowTime.
- ▶ Implemented separate ROE Service subfunctions taking in consideration the StorageState bit (sub-function bit 6). The implementation is based on Bugzilla RfC #72061.
- ▶ ASCDCM-2717 Fixed known issue: The Dcm might not correctly respond with NRC 0x31 when reading multiple DIDs
- ▶ ASCDCM-2737 Fixed known issue: The Dcm maps the wrong security level to ReadMemoryRangeInfo and WriteMemoryRangeInfo elements in case of one of them references the same security level multiple times
- ▶ ASCDCM-2747 Fixed known issue: Unresolved possible error reference in `Dcm_swc_interfaces.arxml` lead to warnings during import
- ▶ ASCDCM-2756 Fixed known issue: Session checks could be incorrect for subservices of the service functions with an ID greater than 0x3E, if any session is configured for any subservice for the UDS service TesterPresent (0x3E)
- ▶ ASCDCM-2782 Fixed known issue: Final negative responses may not be produced after issuing a RequestCorrectlyReceivedResponsePending (0x78) NRC for requests received on functional RxPduIDs
- ▶ All combinations of signal configurations for all RoutineControl (0x31) operations are now available

Module version 4.12.0

2016-06-03

- ▶ The dynamically defined DID source table NvM block is marked to be stored if it has been restored to default after read block error during init.



- ▶ Add support for disabling ECU keep awake through diagnosis after ignition was turned off.
- ▶ ASCDCM-2701 Fixed known issue: ComM_DCM_ActiveDiagnostic(NetworkId) is not invoked when the diagnostic session is changed into a session different than the default session
- ▶ ASCDCM-2675 Fixed known issue: Generation errors if ClientServer interfaces are used for DcmDspRoutines, RequestResults IN-signals have variable length, and StopRoutine IN-signals have fixed length or do not exist.
- ▶ ASCDCM-2689 Fixed known issue: The Dcm might trigger a Det error when dynamically defined DIDs from NvM are initialized.
- ▶ ClientServer interfaces for RequestControlServices are now generated only when DcmDspRequestControl.DcmDspDidRangeUsePort is TRUE
- ▶ NRC 0x36 (exceededNumberOfAttempts) for SecurityAccess service (0x27) is transmitted one time, when SendKey subfunction is requested with an invalid key and the maximum number of failed access attempts is reached.
- ▶ ASCDCM-2674 Fixed known issue: The Dcm might trigger an Os error or block the Ecu when reading a DID with a memory source
- ▶ ASCDCM-2681 Fixed known issue: The Dcm might violate the order of calling memory callouts when multiple requests are processed
- ▶ ASCDCM-2738 Fixed known issue: OBD services Request current powertrain diagnostic data (\$01) and Request powertrain freeze frame data (\$02) are disabled when they are configured in more than one diagnostic services table
- ▶ ASCDCM-2743 Fixed known issue: UDS service DynamicallyDefineDataIdentifier (0x2C) may not transmit NRC RequestOutOfRange (0x31) for invalid (not supported) source DIDs in the OBD range

Module version 4.11.2

2016-04-29

- ▶ ASCDCM-2581 Fixed known issue: An incorrect warning message is issued when OBD DIDs are configured
- ▶ Add selectable removal from generated ClientServerInterface operations of unused IOControl operations
- ▶ Preserve data at buffer pointers provided to DID operations across asynchronous operation call sequences
- ▶ ASCDCM-2521 Fixed known issue: Signal data corruption due to race condition when processing periodic DID transmission in parallel with any DID read/write/control operation
- ▶ Implemented optimization of DID data configuration code
- ▶ ASCDCM-2439 Fixed known issue: The ECU may become unresponsive if a DTC-related OBD service is executed immediately after executing the ReadDTCInformation (0x19) or ReadGenericInformation (0xAF) service



- ▶ ASCDCM-3018 Dcm compilation fails when defining external reportDTCSnapshotRecordByDTCNumber (0x04) or reportDTCExtDataRecordByDTCNumber (0x06) sub-functions for the `ReadGenericInformation` (0xAF) service

Module version 4.11.1

2016-04-01

- ▶ Updated #defines for symbolic name values related to ROE according to AUTOSAR 4.0.3 naming schema
- ▶ ASCDCM-2508 Fixed known issue: Compiler error about exclusive areas declared implicitly in `Dcm_Dsp_MemoryServices.c`
- ▶ ASCDCM-2516 Fixed known issue: ServiceNeedsWizard generates NvM block NVM_BLOCK_DCM_DDDID independent of `DcmDD DIDStorage`
- ▶ ASCDCM-2551 Fixed known issue: Unconfigured dynamically defined data identifiers are not ignored by UDS service `ReadDataByPeriodicIdentifier` (0x2A)
- ▶ ASCDCM-2520 Fixed known issue: Compiler errors and warnings when no `DcmDspMemoryIdInfo` elements and no read or write memory ranges are configured

Module version 4.11.0

2016-03-04

- ▶ Updated #defines for symbolic name values according to AUTOSAR 4.0.3 naming schema
- ▶ ASCDCM-2432 Fixed known issue: Service `WriteMemoryByAddress` (0x3D) does not generate a valid positive response
- ▶ Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeader-File`
- ▶ ASCDCM-2447 Fixed known issue: The Dcm does not provide the definitions of macros which represent values of the `Dcm_SesCtrlType` type if the Rte is disabled
- ▶ ASCDCM-2452 Fixed known issue: NULL-pointer dereference occurs when the `InputOutputControl-ByIdentifier` (0x2F) service is executed if read-access for the DID to be controlled is disabled

Module version 4.10.6

2016-02-10

- ▶ Added the option of dummy configuration of Subfunction 0x00 for service `TesterPresent` (0x3e)
- ▶ ASCDCM-2331 Fixed known issue: DET errors are reported when the StopProtocol operation of the Call-backDCMRequestServices ClientServer Interface returns `E_NOT_OK`



- ▶ Improved transmission latency by triggering transmission of available PDID samples in the same cycle previous transmissions are confirmed
- ▶ ASCDCM-2379 Fixed known issue: Periodic UUDT transmissions occur on wrong TxPdulds when DcmDslPeriodicConnections from multiple protocols are sorted incorrectly
- ▶ ASCDCM-2384 Fixed known issue: The Dcm may become unresponsive if a page-buffered transmission for service ReadDTCInformation (0x19) is processed
- ▶ ASCDCM-2388 Fixed known issue: The service RequestPowertrainFreezeFrameData (0x02) returns wrongly a positive response if the request has incorrect length.
- ▶ ASCDCM-2403 Fixed known issue: Negative response when trying to schedule unconfigured DIDs along with existing and valid ones using UDS service ReadDataByPeriodicIdentifier (0x2A)
- ▶ ASCDCM-2398 Fixed known issue: The service RequestPowertrainFreezeFrameData (0x02) service returns a too long positive response under certain conditions
- ▶ ASCDCM-1850 Fixed known issue: The Dcm module does not compile if a DcmDslProtocolRow has the DcmDslProtocolID set to DCM_SUPPLIER_XX and Rte usage is disabled
- ▶ ASCDCM-2389 Fixed known issue: If service ReadGenericInformation (0xAF) is requested, an NRC 0x10 (general reject) can be wrongly generated
- ▶ The DcmDslProtocolTransType configuration parameter is made deprecated. The decision on which TxPdulds to transmit is now based on the presence of DcmDslPeriodicTransmissionConRef or DcmDslROEConnectionRef in the DcmDslMainConnection on which a periodic transmission is configured and on the presence of TxPdulds on the channels referenced by those two configuration parameters.
- ▶ ASCDCM-2415 Fixed known issue: The ECU resets if a DID with a smaller identifier than any configured DID is used in a request to any of the InputOutputControlByIdentifier (0x2F), ReadDataByIdentifier (0x22), ReadDataByPeriodicIdentifier (0x2A) or WriteDataByIdentifier (0x2E) services
- ▶ Improve Design and fix Design issues regarding service 0x2a

Module version 4.10.5

2016-01-15

- ▶ ASCDCM-2291 Fixed known issue: Generator and/or compiler errors occur when DID F186 is not configured with a signal with synchronous callout access
- ▶ ASCDCM-2314 Fixed known issue: The Dcm may become unable to transmit on a TxPduld in case of protocol cancellation
- ▶ ASCDCM-2315 Fixed known issue: Incorrect security level transitions and response messages when using SecurityAccess (0x27) for an invalid security level
- ▶ ASCDCM-2207 Fixed known issue: Asynchronous 'returnControlToECU' operations are always called with a DCM_INITIAL OpStatus when switching to the default session



- ▶ ASCDCM-2285 Fixed known issue: Memory corruption when service `ReadGenericInformation` (0xAF) is requested for all DTCs
- ▶ ASCDCM-2339 Fixed known issue: A compile error occurs if only start routine out signal has variable length
- ▶ ASCDCM-2306 Fixed known issue: The Dcm may become unresponsive as a result of executing a request for the `ReadDTCInformation` (0x19) service with the `suppressPosRspMsgIndicationBit` set
- ▶ ASCDCM-2128 Fixed known issue: Incorrect positive responses are reported for the `reportUserDefMemoryDTCExtDataRecordByDTCNumber` (0x19) subfunction of the `ReadDTCInformation` (0x19) service
- ▶ ASCDCM-2252 Fixed known issue: Periodic transmissions may be disabled if a periodic transmission event cannot be dispatched
- ▶ ASCDCM-2321 Fixed known issue: Erroneous interpretation of Dem error codes provokes unexpected behavior of the Dcm
- ▶ ASCDCM-2214 Fixed known issue: The Dcm returns a NRC when trying to read a dynamically defined DID containing an OBD DID as source
- ▶ ASCDCM-2362 Fixed known issue: Dcm might block or timeout after transmission when multiple periodic connections are configured per periodic diagnostic channel
- ▶ ASCDCM-2340 Fixed known issue: Service 0xAF with subservice 0x04 is erroneously providing a positive response under circumstances where a negative response "Conditions Not Correct" would apply
- ▶ ASCDCM-2358 Fixed known issue: Dcm module does not compile if services 0x07 or 0x0A and parameters `DcmDevErrorDetect`, `DcmDefProgEnabled`, and `DcmUnreachAssertEnabled` are enabled
- ▶ ASCDCM-2360 Fixed known issue: Service `ReadDTCInformation` (0x19) runs into timeout if used with a Dem module other than the EB Dem.
- ▶ ASCDCM-2311 Fixed known issue: UDS service `DynamicallyDefineDataIdentifier` (0x2C) allows configuring oversized dynamically defined DIDs
- ▶ ASCDCM-2334 Fixed known issue: Compiler errors occur if the `InputOutputControlByIdentifier` (0x2F) service is enabled, but no DID has control access.
- ▶ ASCDCM-1653 Fixed known issue: Wrong mode switches will be executed for ComM channels when the `DcmDspComControlCommunicationReEnableModeRuleRef` mode rule fails. Note: This bugfix changes the numerical values of the mode declarations.
- ▶ ASCDCM-2368 Fixed known issue: Service `ReadGenericInformation` (0xAF) sub-service 0x04 response contains too much data when no event data is present for a particular DTC.
- ▶ ASCDCM-1759 Fixed known issue: DTC Setting is reenabled on session change only when the internal `DiagnosticSessionControl` (0x10) service handler is used
- ▶ ASCDCM-2222 Fixed known issue: ComM Channels are not reset to the default communication mode when the `DcmDspComControlCommunicationReEnableModeRuleRef` ModeRule is not fulfilled if no `DcmDspComControlSpecificChannel` is configured



- ▶ ASCDCM-2375 Fixed known issue: Port interfaces for service `RoutineControl` might be defined following the wrong AUTOSAR specification version if vendor-specific parameter `DcmDspRoutineUsePortASRVersion` is disabled

Module version 4.10.4

2015-12-04

- ▶ ASCDCM-2255 Fixed known issue: The protocol on which a TYPE1 periodic request was received gets blocked in active state if the periodic request is processed on a different protocol
- ▶ ASCDCM-2242 Fixed known issue: Dcm does not correctly handle three or more parallel requests
- ▶ ASCDCM-2280 Fixed known issue: The ECU resets after it removes periodic DIDs from the scheduler with the `stopSending` (0x04) subfunction of the UDS service `ReadDataByPeriodicIdentifier` (0x2A)
- ▶ ASCDCM-2284 Fixed known issue: Incorrect answer for service `ReadGenericInformation` (0xAF) when shared reception and transmission buffers are used
- ▶ ASCDCM-2293 Fixed known issue: The Dcm becomes unresponsive if an external service handler returns an `E_NOT_OK` error

Module version 4.10.3

2015-11-06

- ▶ ASCDCM-2136 Fixed known issue: Compile errors occur when `DcmDspRoutines` have fixed length OUT signals and variable length IN signals
- ▶ ASCDCM-2187 Fixed known issue: Response contains IDs of unsupported OBD DIDs if service `ReadDataByIdentifier` (0x22) is used
- ▶ ASCDCM-2054 Fixed known issue: The service `Dynamically Define Data Identifier` (0x2C) could misbehave if DIDs from OBD Range are used as source
- ▶ ASCDCM-2206 Fixed known issue: Incorrect responses to the `RoutineControl` (0x31) service when `DcmDspRoutines` have fixed length OUT signals and variable length IN signals
- ▶ ASCDCM-2237 Fixed known issue: The space allocated for sampling periodic DIDs can only be used to max elements minus one
- ▶ Implemented support for reading the data identifier 0xF186 (`ActiveDiagnosticSessionDataIdentifier`)

Module version 4.10.2

2015-10-22

- ▶ ASCDCM-2215 Fixed known issue: Integrating the Dcm with the EB PduR results in errors



- ▶ ASCDCM-2213 Fixed known issue: Dcm transmission may block if a lower layer module calls the `Dcm_CopyTxData()` or `Dcm_TpTxConfirmation()` APIs from an interrupt context or from the context of the `PduR_DcmTransmit()` API

Module version 4.10.1

2015-10-09

- ▶ ASCDCM-2071 Fixed known issue: The NRC set from the application is ignored if the `InputOutputControlByIdentifier (0x2F)` service is handled asynchronously and development error detection is enabled.
- ▶ ASCDCM-2186 Fixed known issue: Interfaces responsible for reading DID signals may be erroneously called with the `DCM_PENDING OpStatus`
- ▶ ASCDCM-2113 Fixed known issue: The ECU may block if one of the configured DIDs has just `returnControlToEcu` access enabled
- ▶ Add new vendor specific configuration parameter for user configurable Addressing of User-Defined Memory `DEM_DTC_ORIGIN_SECONDARY_MEMORY`
- ▶ ASCDCM-2147 Fixed known issue: If an OBD MID is requested with service `ReadDataByIdentifier (0x22)`, the response is too long

Module version 4.10.0

2015-10-02

- ▶ ASCDCM-2078 Fixed known issue: The service `DiagnosticSessionControl (0x10)` can block the ECU with incorrect configuration
- ▶ ASCDCM-2022 Fixed known issue: The service `Dynamically Define Data Identifier (0x2C)` could misbehave if DIDs from DID Ranges are used as source
- ▶ ASCDCM-2085 Fixed known issue: Wrong response if a request of service `ReadMemoryByAddress (0x23)` is made and `Dcm_DspInternalReadMemoryAddress_Execute()` returns `DCM_E_READ_FORCE_RCRRP`
- ▶ Changed name of file `Dcm_Dsp_SvcH_OBDMode6.c/.h` to `Dcm_Dsp_SvcH_RequestControlOfOn-BoardSysTstComp.c/.h`
- ▶ ASCDCM-2083 Fixed known issue: Generator errors occur when DcmDspRoutines have fixed length OUT signals and variable length IN signals
- ▶ ASCDCM-2095 Fixed known issue: `Dcm_DsdServiceUsed` may deactivate several services in different `DcmDsdServiceTable`
- ▶ ASCDCM-2117 Fixed known issue: Incorrect NRC is sent for UDS service `InputOutputControlByIdentifier (0x2F)` with `inputOutputControlParameter` out of range



- ▶ Implemented UUDT transmission and AUTOSAR 4.2.1 - compatible handling for UDS service `ReadDataByPeriodicIdentifier` (0x2A)
- ▶ ASCDCM-2152 Fixed known issue: The service `ReadGenericInformation` (0xAF) runs into an endless loop under certain conditions
- ▶ ASCDCM-2155 Fixed known issue: If an OBD availability PID is requested with service `ReadDataByIdentifier` (0x22), the response is invalid
- ▶ Added config checks according service 0x06 and OBDMID
- ▶ ASCDCM-2172 Fixed known issue: If service `ReadDataByIdentifier` (0x22) is an OBD availability VehicleInfoType e.g. by requesting DID 0xF800, SW may crash

Module version 4.9.1

2015-09-07

- ▶ ASCDCM-3700 Fixed known issue: Unexpected behavior when the `CommunicationControl` (0x28) service is configured and the `DiagnosticSessionControl` (0x10) service is requested
- ▶ ASCDCM-2060 Fixed known issue: `Dcm_DemFacade_GetEventData()` broken, leads to not working `ReportDtcSnapshotRecordByDtcNumber`
- ▶ ASCDCM-2089 Fixed known issue: `MamMap` include issue

Module version 4.9.0

2015-08-28

- ▶ Support of DIDs in system supplier specific range
- ▶ Introduced NRC 0x14 (`responseTooLong`) for UDS service `ReadDataByIdentifier` 0x22
- ▶ ASCDCM-2074 Fixed known issue: Compiler warnings are reported when OBD services \$06 and \$08 are both used
- ▶ Implemented support for protocol preemption

Module version 4.8.2

2015-07-31

- ▶ Added a limitation for the service `Dynamically Define Data Identifier` (0x2C). The service responds with NRC 0x31 in case of adding a Dynamic DID as source.
- ▶ ASCDCM-1970 Fixed known issue: An illegal memory access may occur if the subfunction `defineByIdentifier` (0x01) of service `DynamicallyDefineDataIdentifier` (0x2C) is used
- ▶ ASCDCM-2034 Fixed known issue: Wrong data is written for a DID if service `WriteDataByIdentifier` (0x2E) is used with a DID range



- ▶ ASCDCM-2026 Fixed known issue: Description of operations of interface `DataServices_DIDRange` does not comply to AUTOSAR 4.2.1
- ▶ Implemented support for configuring diagnostic sessions for sub-functions of UDS service `SecurityAccess` 0x27
- ▶ ASCDCM-2001 Fixed known issue: In OBD service 0x02 (Request powertrain freeze frame data) PID 0x00 does not show PID 0x02 as supported

Module version 4.8.1

2015-07-10

- ▶ ASCDCM-1954 Fixed known issue: The ECU may block or reset if the number of signals of a DID is higher than 254
- ▶ ASCDCM-1938 Fixed known issue: If `xxx_GetInfotypeValueData()` returns pending, an unreachable code error may be displayed (OBD service 0x09)
- ▶ ASCDCM-1933 Fixed known issue: Compiler or generator errors may occur when AUTOSAR 4.2.1 APIs are used by `DcmDspRoutines` which contain `VARIABLE_LENGTH` signals
- ▶ ASCDCM-1953 Fixed known issue: Code generation fails when a DID is configured for 'Control' access without explicit 'Read' access and uses `ClientServerInterfaces` for data access
- ▶ ASCDCM-1998 Fixed known issue: Incomplete writing of signals may occur using service `WriteDataByIdentifier(0x2E)` with DID including multiple signals with `USE_BLOCK_ID` interface
- ▶ ASCDCM-2002 Fixed known issue: Dcm triggers a DET error if service `ReadDataByIdentifier` (0x22) is canceled
- ▶ ASCDCM-1748 Fixed known issue: The Dcm does no longer respond to requests if the communication channel is switched off while it dispatches an ROE event
- ▶ ASCDCM-1978 Fixed known issue: Wrong response messages are sent for service `SecurityAccess` (0x27)
- ▶ ASCDCM-1936 Fixed known issue: The service `Dynamically Define Data Identifier` (0x2C) allows misconfiguration which results in incorrect run-time behavior
- ▶ ASCDCM-1641 Fixed known issue: 'Unused variable' compiler warning under certain configuration
- ▶ Implemented support for reading OBD data identifiers using service `ReadDataByIdentifier` (\$22)

Module version 4.8.0

2015-06-19

- ▶ ASCDCM-1783 Fixed known issue: Service handlers act as if the response transmission was successful if the response cannot be sent before the P2 timeout expires



- ▶ ASCDCM-1913 Fixed known issue: No DTCs are reported after ClearDTC in OBD service (0x0A) "Request emission-related diagnostic trouble codes with permanent status"
- ▶ ASCDCM-1615 Fixed known issue: DTC setting may be enabled on every session change if DCM_ALL_SESSION_LEVEL is configured for service ControlDTCsetting (0x85)
- ▶ ASCDCM-949 Fixed known issue: The Dcm relies on the fact that the Dem interfaces never return a "PENDING" status
- ▶ ASCDCM-1747 Fixed known issue: Unnecessary check of the size of all extended data records for a DTC
- ▶ Implemented Calibration of Dcm Services
- ▶ Adapted EB-internal assert handling
- ▶ ASCDCM-1447 Fixed known issue: ECU resets too early if a jump to the bootloader is executed by using the service DiagnosticSessionControl (0x10)
- ▶ Implemented support for DTCSettingControlOptionRecord and LengthCheck for ControlDTCSetting service request according to AUTOSAR 4.2.1.
- ▶ ASCDCM-1879 Fixed known issue: Compiler errors are reported if the DynamicallyDefineDataIdentifier (0x2C) service is used and none of the RequestDownload (0x34), RequestUpload (0x35), ReadMemoryByAddress (0x23) or WriteMemoryByAddress (0x3D) services are used
- ▶ Implemented customer-specific diagnostic service ReadGenericInformation (0xAF).
- ▶ ASCDCM-1915 Fixed known issue: Request type (reqType) in Dcm_MsgAddInfoType is incorrectly defined
- ▶ Implemented UDS service ReadDTCInformation (0x19), subservice ReportUserDefMemory-DTCSnapshotRecordByDTCNumber (0x18).

Module version 4.7.0

2015-04-29

- ▶ Implemented support for OBD Service \$06 - Request On-Board Monitoring Test-results for Specific Monitored Systems API according to both AUTOSAR 4.0.3 and AUTOSAR 4.2.1 releases.
- ▶ ASCDCM-1836 Fixed known issue: Rte interface for the RoutineControl service is not correctly generated
- ▶ ASCDCM-1868 Fixed known issue: If AUTOSAR 4.2.1 DataServices interfaces are used, a compile error might occur in case when RTE is not configured
- ▶ ASCDCM-1866 Fixed known issue: Compiler errors are reported when the RoutineControl (0x31) service is configured and the default Dcm interface version is set to Autosar 4.2.1
- ▶ ASCDCM-1876 Fixed known issue: Application call-back function IsDidAvailable of feature DidRanges is not using Dcm_DidSupportedType



- ▶ ASCDCM-1776 Fixed known issue: Compiler errors are reported if the `Dcm_ReadMemory()` or `Dcm_WriteMemory()` callbacks are not defined
- ▶ Implemented support for suppressed NRC of functional requests according to both ISO14229-2006 and ISO14229-2013 releases.
- ▶ Implemented support for the `DCM_Roe ClientServerInterface` according to AUTOSAR 4.2.1 by:
 1. having the `RoeEventId` parameter defined as a Portdefined argument value;
 2. changing the name of the generated P-port to `DCM_Roe_{RoeName}`;as a configuration selectable alternative to the existing `DCM_Roe ClientServerInterface` implementation according to AUTOSAR 4.0.3
- ▶ Implemented support for calibration of configuration parameters: `DcmDspPidUsed`, `DcmDspTestResultObdmidUsed`, `DcmDspTestResultTid`, `DcmDspVehInfoUsed` and `DcmDspRequestControlUsed`.
- ▶ Implemented support for the OBD service \$09 Request Vehicle Information.
- ▶ Implemented support for OBD Service \$08 Request Control of On-Board System, Test or Component
- ▶ ASCDCM-1903 Fixed known issue: Response transmission may not be possible if TYPE2 periodic responses are used and DET checks are enabled
- ▶ ASCDCM-1897 Fixed known issue: The `reportUserDefMemoryDTCByStatusMask(0x17)` sub-function of service `ReadDTCInformation (0x19)` is not configurable in Tresos.

Module version 4.6.4

2015-04-02

- ▶ Fixed usage of Dem API interfaces `Dem_GetDTCOfOBDFreezeFrame` and `Dem_ReadDataOfOBDFreezeFrame`
- ▶ ASCDCM-1865 Fixed known issue: If a DidRange is configured without any service using Dids, a compiler error occurs

Module version 4.6.3

2015-03-27

- ▶ Improved the asynchronous processing of service handlers by breaking the continuous processing within a loop inside a separate task into discreet calls to the service handler function. Each call is executed within a separate task, thus allowing tasks of a lower priority than the asynchronous service processor task to also be scheduled, as they will no longer be blocked until the service handling is completed.
- ▶ Added software units for diagnostic services `0x22`, `0x2A` and `0x2E`. Added `DidServices` software unit and improved diagnostic service handlers `0x22` and `0x2E`.
- ▶ Implemented support for the OBD services:



- ▶ \$03 Request emission-related diagnostic trouble codes
- ▶ \$07 Request emission-related diagnostic trouble codes detected during current or last completed driving cycle
- ▶ \$0A Request emission-related diagnostic trouble codes with permanent status
- ▶ ASCDCM-1787 Fixed known issue: The ECU becomes unresponsive or memory corruption occurs if a DID is requested which is unsupported and outside the `ReadDID Range`
- ▶ Implemented support for the `SecurityAccess` API according to AUTOSAR 4.2.1 as a configuration selectable alternative to the existing `SecurityAccess` API implementation according to AUTOSAR 4.0.3
- ▶ ASCDCM-1837 Fixed known issue: The ECU becomes unresponsive, restarts or performs faulty TYPE2 periodic response transmissions
- ▶ Implemented support for the `DataServices` API according to AUTOSAR 4.2.1 as a configuration selectable alternative to the existing `DataServices` API implementation according to AUTOSAR 4.0.3
- ▶ ASCDCM-1843 Fixed known issue: File `AdjacentLayer.properties` contains wrong x-path expressions which leads to errors during generation of PduR
- ▶ Implemented support for the OBD service: \$04 Clear/reset emission-related diagnostic information

Module version 4.6.2

2015-03-06

- ▶ ASCDCM-1771 Fixed known issue: The ECU may reset or become unresponsive if service `InputOutputControlByIdentifier` (0x2F) is requested
- ▶ ASCDCM-1674 Fixed known issue: Undefined identifier errors are reported when Rte usage is enabled in Dcm and Dcm internal services are not configured to use a Client/Server interface
- ▶ Added improvements to the the `DynamicallyDefineDataIdentifier` (0x2C), `ReadDataByPeriodicIdentifier` (0x2A) and `ReadDataByIdentifier` (0x22) service handlers
- ▶ Changed datatype of the index used to search the DID table in `Dcm_DspInternal_OnChangeOfDataIdentifier_SvcStart` to `uint16_least` in order to avoid possible problems on 8-bit architectures.
- ▶ ASCDCM-1777 Fixed known issue: Compiler errors are reported if the `DynamicallyDefineDataIdentifier` (0x2C) service is configured, but the `ReadMemoryByAddress` (0x23) service is not configured
- ▶ ASCDCM-1779 Fixed known issue: Compiler errors are reported if the `ReadMemoryByAddress` (0x23), the `WriteMemoryByAddress` (0x3D), or the `DynamicallyDefineDataIdentifier` (0x2C) service is configured, but the `DcmDspMemory` container is disabled
- ▶ ASCDCM-1671 Implemented UDS service `ReadDTCInformation` (0x19), subservice `ReportUserDefMemoryDTCExtDataRecordByDTCNumber` (0x19)
- ▶ ASCDCM-1803 Fixed known issue: The ECU becomes unresponsive or memory corruption occurs when the `RoutineControl` (0x31) service is requested without a `routineControlType` sub-function



- ▶ ASCDCM-1498 Fixed known issue: Dcm communication may fail in the event of a paged buffering transmission timeout
- ▶ Implemented ISO14229-1:2013 Response Message Format (UUDT Identifiers) for the diagnostic service `ReadDataByPeriodicIdentifier` (0x2A) responses.
- ▶ Implemented support for TYPE2 periodic responses.
- ▶ Implemented support for the `RoutineServices` API according to AUTOSAR 4.2.1 as a per routine selectable alternative to the existing `RoutineServices` API implementation according to AUTOSAR 4.0.-3 The implementation is based on AUTOSAR Bugzilla RFC #57860

Module version 4.6.1

2015-02-13

- ▶ ASCDCM-1586 Fixed known issue: Incorrect response to UDS service 0x19 0x06 if the DTC isn't stored in error memory
- ▶ ASCDCM-1535 Fixed known issue: `ResponseOnEvent` transmissions may free buffers held by preempted normal diagnostic transmissions
- ▶ ASCDCM-1562 Fixed known issue: Corrupt responses for service `ReadDataByIdentifier` (0x22) and unreliable behavior for `onChangeOfDataIdentifier`-triggered ROE events when asynchronous service processing is used
- ▶ Implemented support for the `Dem_DcmClearDTC()` API according to AUTOSAR 4.2.1 as a configuration selectable alternative to the existing `Dem_ClearDTC()` AUTOSAR 4.0.3 API

Module version 4.6.0

2015-01-21

- ▶ Implemented negative response with NRC 0x10 instead of 0x72 upon `Dcm_ReadMemory` failing with `DCM_READ_FAILED`, considering both AUTOSAR 4.2 rev1 and Bugzilla issue https://www.autosar.org/bugzilla/show_bug.cgi?id=57196
- ▶ ASCDCM-1565 Fixed known issue: Executing the `EcuReset` (0x11) service may fail if a previous request to the same service timed-out
- ▶ ASCDCM-1564 Fixed known issue: Service `DiagnosticSessionControl` (0x10) may fail to execute after a timed-out jump to the bootloader
- ▶ ASCDCM-1488 Fixed known issue: The Dcm does not reset the `CommunicationControl` when switching to the default session
- ▶ ASCDCM-1340 Fixed known issue: Generated symbol names for `DcmDslProtocolRx`, `DcmDslProtocolID`, `DcmDspSecurityLevel` and `DcmDspSessionLevel` result in name clashes
- ▶ ASCDCM-1241 Fixed known issue: External subfunction handlers for the service `ReadDTCinformation` (0x19) are only called with the `DCM_INITIAL` `OpStatus`.



- ▶ Implemented UDS service `ReadDTCInformation` (0x19), subservice `ReportDTCFaultDetectionCounter` (0x14)
- ▶ Implemented call of `SchM_Switch_Dcm_DcmDiagnosticSessionControl()` at initialization time if the initialization is a result of a warm-start condition.
- ▶ Implemented rejection of diagnostic requests after a positive response to the `EcuReset` (0x11) service, considering both AUTOSAR 4.1 rev3 and Bugzilla issue https://www.autosar.org/bugzilla/show_bug.cgi?id=53154.
- ▶ Implemented requirement Dcm768: Call of `BswM_Dcm_ApplicationUpdated` after jump from bootloader
- ▶ Implemented support of OBD DIDs via regular UDS-style DID access.
- ▶ The services `SecurityAccess` (0x27), `CommunicationControl` (0x28), `ControlDTCSetting` (0x85), `ReadDataByPeriodicIdentifier` (0x2A), and `InputOutputControlByIdentifier` (0x2F) can now be configured to be available in the default session
- ▶ Implemented support for the configuration parameter `DcmDspDataConditionCheckReadFncUsed`, as described in Autosar Bugzilla RfC #53669
- ▶ ASCDCM-1753 Fixed known issue: If the `ResponseOnEvent` (0x86) service is used with an `OnChangeOfDataIdentifier` (0x03) event for an internally managed DID which does not have Read-access enabled, the ECU may reset or become unresponsive
- ▶ Implemented support for the `DEM_NUMBER_PENDING` return value for the `Dem_GetNumberOfFilteredDTC()` API.
- ▶ Implemented support for the `DEM_FILTERED_PENDING` return value for the `Dem_GetNextFilteredDTC()` API.
- ▶ Implemented support for the `DEM_FILTERED_PENDING` return value for the `Dem_GetNextFilteredRecord()` API.

Module version 4.5.0

2014-10-22

- ▶ ASCDCM-1526 Fixed known issue: The Dcm may report errors during the generation phase if the default session's `DcmDspSession` entry is not named `DCM_DEFAULT_SESSION` and `DcmModeDeclarationSupport` is enabled
- ▶ ASCDCM-1529 Fixed known issue: Compiler errors are reported if the `ResponseOnEvent` (0x86) service is configured without configuring at least one `DcmDslResponseOnEvent` connection
- ▶ ASCDCM-1567 Fixed known issue: Jumping to the boot loader fails if the delay in setting the `ProgrammingConditions` causes a P2/P2* server timeout



- ▶ ASCDCM-1416 Fixed known issue: The Dcm may time out or perform the jump to the bootloader very late if the `DiagnosticSessionControl (0x10)` is configured to be run asynchronously
- ▶ ASCDCM-1552 Fixed known issue: Compiler error when a single protocol is configured and the number of `RequestCorrectlyReceivedResponsePending` NRCs that the Dcm can send is unlimited
- ▶ ASCDCM-1483 Fixed known issue: The Dcm does not enable DTC setting when transitioning to the default diagnostic session outside of an S3 timeout scenario
- ▶ ASCDCM-1566 Fixed known issue: Jumping to the bootloader may fail if a previous request to the `EcuReset (0x11)` service timed-out
- ▶ ASCPD-189 Implemented subfunction `reportDTCWithPermanentStatus (0x15)` of UDS service `ReadDTCInformation (0x19)`
- ▶ ASCPD-189 Implemented subfunction `reportEmissionsRelatedOBDDTCByStatusMask (0x13)` of UDS service `ReadDTCInformation (0x19)`
- ▶ ASCDCM-1540 Fixed known issue: The Dcm may become unresponsive when executing service `RoutineControl (0x31)`
- ▶ ASCDCM-1557 Fixed known issue: The Dcm may corrupt the memory when ComM channels are re-enabled when the `DcmDspComControlCommunicationReEnableModeRuleRef` mode rule fails
- ▶ ASCDCM-1558 Fixed known issue: The wrong ComM channels are re-enabled when the `DcmDspComControlCommunicationReEnableModeRuleRef` mode rule fails
- ▶ ASCDCM-1597 Fixed known issue: The Dcm becomes unresponsive if a bootloader jump is executed with incorrect programming conditions
- ▶ ASCDCM-1484 Fixed known issue: The Dcm does not enable DTC setting correctly after a successful `ECUReset (0x11)` request
- ▶ ASCDCM-1444 Fixed known issue: `ComM_DCM_ActiveDiagnostic()` is not always called before an NRC is sent
- ▶ ASCDCM-1585 Fixed known issue: Incorrect behavior of service `ReadDTCInformation (0x19)` with subservices `reportDTCSnapshotRecordByDTCNumber (0x04)` and `reportDTCExtendedDataRecordByDTCNumber (0x06)` if paged buffering is used
- ▶ Implemented configuration parameter support for the `DynamicallyDefineDataIdentifier (2C hex)` service
- ▶ Implemented UDS service `ReadDataByPeriodicIdentifier (2A hex)`
- ▶ Implemented UDS service `DynamicallyDefineDataIdentifier (2C hex)`
- ▶ ASCDCM-2122 Fixed known issue: Buffer corruption occurs if the maximum size of data is requested that may fit in the transmission buffer with the `ReadMemoryByAddress (0x23)` service

Module version 4.4.4

2014-08-08



- ▶ ASCDCM-1463 Fixed known issue: Out of bounds memory access when ROE events are not configured and started
- ▶ ASCDCM-1489 Fixed known issue: Compiler warnings are reported if the same symbol name is used more than once for RoutineControl operations
- ▶ ASCDCM-1413 Fixed known issue: Programming conditions are corrupted when the DiagnosticSessionControl (0x10) service and asynchronous processing are used
- ▶ ASCDCM-1412 Fixed known issue: The Dcm may attempt a jump to the bootloader while a previous call of `Dcm_SetProgConditions()` is still underway
- ▶ Implemented `DcmModeRule` support for reading or writing individual `DcmDspMemory` ranges
- ▶ Implemented `DcmModeRule` support for reading, writing, and controlling individual DIDs
- ▶ Implemented `DcmModeRule` support for re-enabling of DTC setting
- ▶ Implemented `DcmModeRule` support for resetting the state of the `CommunicationControl` service
- ▶ ASCDCM-1475 Fixed known issue: The Dcm performs service and subfunction diagnostic session and security level checks incorrectly
- ▶ ASCDCM-1491 Fixed known issue: `OnChangeOfDataIdentifier` ROE event generation fails for DIDs containing variable length signals
- ▶ ASCDCM-1556 Fixed known issue: Compiler errors are reported if Mode Declaration support is enabled and no specific channels for the `CommunicationControl` (0x28) service are configured
- ▶ ASCDCM-1555 Fixed known issue: Compiler warnings are reported for certain configuration options for diagnostic session handling
- ▶ ASCDCM-1438 Fixed known issue: Unexpected concurrencies and possible memory corruption occur if asynchronous service processing is used
- ▶ Added the possibility to configure input signals for the `requestRoutineResults` (0x01) sub-function of the `RoutineControl` (0x31) diagnostic service
- ▶ ASCDCM-1524 Fixed known issue: The Dcm reports warnings during the generation phase

Module version 4.4.3

2014-04-25

- ▶ ASCDCM-1311 Fixed known issue: The Dcm may send responses for service `ReadDTCInformation` (0x19) even if the communication channel is in silent communication mode
- ▶ ASCDCM-1329 Fixed known issue: The `requestSeed` / `compareKey` subfunction sequence of the `SecurityAccess` (0x27) service is not reset correctly
- ▶ ASCDCM-1339 Fixed known issue: The Dcm transmits service responses with the wrong `TxDuld` when only one protocol (`DcmDslProtocolRow`) is configured



- ▶ ASCDCM-1315 Fixed known issue: Duplicate CurHsmInfo variable in the `Dcm_DsdInternal_AllocateTxBuffer()` function
- ▶ Improvement to make sure all periodic DIDs are sampled and sent under the condition that more samples are generated per time unit as can be send
- ▶ ASCDCM-1336 Fixed known issue: Data may be corrupted when writing DID signals to NVRAM using the service `WriteDataByIdentifier (0x2E)`
- ▶ ASCDCM-1139 Fixed known issue: Configuration of a DID referencing other DIDs but containing no signals is not possible
- ▶ ASCDCM-1357 Fixed known issue: Service `ResponseOnEvent (0x86)` sends wrongly addressed messages, corrupts buffers and locks the Dcm
- ▶ ASCDCM-1361 Fixed known issue: The response to the `DiagnosticSessionControl (0x10)` that causes a jump/return from the bootloader is sent on an invalid txPdul
- ▶ ASCDCM-1346 Fixed known issue: Parameters `DcmDspDataUsePort` and `DcmDspSecurityUsePort` use values different from the AUTOSAR-specified ones
- ▶ ASCDCM-1366 Fixed known issue: Subfunction `ReportDtcSnapshotRecordByDtcNumber (0x04)` of service `ReadDTCInformation (0x19)` does not allow the request of specific data records with values 0x00 or 0xF0-0xFE
- ▶ ASCDCM-1396 Fixed known issue: Usage of `Dcm_GetActiveProtocol()` may cause an invalid memory access
- ▶ ASCDCM-1394 Fixed known issue: Dcm communication modes may not be set correctly by the ComM
- ▶ ASCDCM-1406 Fixed known issue: Dcm communication modes may not be interpreted correctly
- ▶ ASCDCM-1395 Fixed known issue: Use of `Dcm_ResetToDefaultSession()` may result in an invalid memory access, corrupt diagnostic session settings and communication failure
- ▶ ASCDCM-1414 Fixed known issue: Usage of `Dcm_GetSesCtrlType()` may result in an invalid memory access or the wrong session identifier being reported
- ▶ ASCDCM-1398 Fixed known issue: The Dcm may send an extra `requestCorrectlyReceivedResponsePending` NRC when executing a jump to the Bootloader via an asynchronous `DiagnosticSessionControl (0x10)` request
- ▶ ASCDCM-1401 Fixed known issue: Buffer corruption and/or blocked Dcm when ROE final responses are transmitted during a request reception
- ▶ ASCDCM-1429 Fixed known issue: If the service `ECUReset (0x11)` is configured to use an internal service handler, but only user-implemented subfunction handlers are used, a compiler warning is reported
- ▶ ASCDCM-1421 Fixed known issue: Invalid memory access when unused `DcmDspRoutineInfo` members are configured
- ▶ ASCDCM-1403 Fixed known issue: Use of the `Dcm_ResetToDefaultSession()` API function may cause inconsistent or faulty behaviour in the Dcm



- ▶ ASCDCM-1332 Fixed known issue: Jumping to the bootloader is not executed correctly if a `requestCorrectlyReceivedResponsePending(0x78)` NRC is to be sent in the same Dcm cycle
- ▶ ASCDCM-1415 Fixed known issue: If asynchronous service handlers time out, the wrong NRC may be transmitted
- ▶ ASCDCM-1220 Fixed known issue: DID data may get corrupted as a result of a DID NvM Block read/write operation taking too much time to execute
- ▶ ASCDCM-1430 Fixed known issue: ROE final responses cause the Dcm to become unresponsive
- ▶ ASCDCM-1397 Fixed known issue: The Dcm may fail to respect S3 timing requirements
- ▶ ASCDCM-1439 Fixed known issue: The Dcm may block if asynchronously processed services time out
- ▶ ASCDCM-1158 Fixed known issue: Overflows because of values assigned to `DcmDspSecurityAddress`, `DcmDspSecurityKeySize` and `DcmDspSecuritySeedSize`
- ▶ Implemented `DcmModeRule` support for controlling individual Routines
- ▶ ASCDCM-1469 Fixed known issue: Generator reports a warning if Rte usage is enabled and externally managed DIDs are not configured

Module version 4.4.2

2013-11-27

- ▶ ASCDCM-1103 Fixed known issue: The Dcm may send corrupt responses as a result of a race condition between transmission and reception
- ▶ ASCDCM-1162 Fixed known issue: Generator errors when `DcmDspRoutineUsed` is FALSE and `DcmDspRoutineUsePort` is TRUE
- ▶ ASCDCM-1314 Fixed known issue: MemMap section for uninitialized variables of unspecified length and 32 bit variable in Dcm.c.m4 is not correct
- ▶ ASCDCM-1226 Fixed known issue: The Dcm accepts requests before being fully initialized
- ▶ ASCDCM-1217 Fixed known issue: Wrong `ProtocolID` written in `ProgConditions`
- ▶ ASCDCM-1292 Fixed known issue: The Dcm may block during processing of ROE events
- ▶ Implemented the possibility to have mixed (Variable length and Fixed length) signal types present in the `RoutineControl (0x31)` service operation interfaces
- ▶ Implemented the possibility to call `Dem_DcmCancelOperation()` in case the service `ClearDiagnosticInformation (0x14)` times out as a result of too many `REQUESTCORRECTLYRECEIVEDRESPONSEPENDING` NRCs having been sent
- ▶ ASCDCM-1168 Fixed known issue: The Dcm fails to respond to service requests received when the communication channel used is in Silent Communication Mode
- ▶ ASCDCM-1090 Fixed known issue: Incorrect service handler return values dependent on user-implemented code (callouts and Rte interface calls) may block DCM



Module version 4.4.1

2013-10-18

- ▶ ASCDCM-1182 Fixed known issue: The handling of statuses for a routine ("started" or "stopped") is not done correctly
- ▶ ASCDCM-1120 Fixed known issue: Asynchronous DID interfaces/operations are called with an OpStatus of DCM_INITIAL just for the first signal belonging to the DID
- ▶ ASCDCM-1113 Fixed known issue: A protocol processing an ROE event may corrupt the transmit buffers of another currently executing protocol
- ▶ ASCDCM-1250 Fixed known issue: DTC record updating may remain disabled if the subfunction ReportDTCsnapshotRecordbyDTCNumber of the service ReadDTCInformation (0x19) fails
- ▶ ASCDCM-1295 Fixed known issue: Asynchronous service processing of external service handlers can cause unwanted behavior

Module version 4.4.0

2013-09-17

- ▶ ASCDCM-1206 Fixed known issue: Page-buffered transmission may fail if the lower layer only inquires about the remaining amount of data in the Tx buffer when all data has been sent from the current page, instead of requesting more data
- ▶ ASCDCM-927 Fixed known issue: Asynchronous RoutineControl and DataServices service handlers are not called with OpStatus DCM_CANCEL
- ▶ Implemented support for multiple DcmDslMainConnections per DcmDslProtocolRow
- ▶ Introduced a configuration parameter DcmDspRoutineVariableLengthInBytes in the DcmDsp container to indicate whether the VARIABLE_LENGTH signals in the Routine Control configurations are expressed in bytes or bits
- ▶ ASCDCM-882 Fixed known issue: Range of values for DcmDspSecurityDelayTime needs to be extended
- ▶ ASCDCM-1236 Fixed known issue: The Dcm reports a DET error when Dcm_GetProgConditions() returns DCM_COLD_START at startup
- ▶ ASCDCM-1230 Fixed known issue: Compiler errors due to too large shift counts on 16-bit architectures
- ▶ ASCDCM-1009 Fixed known issue: User-configured subfunction handlers for EB-implemented services CommunicationControl (0x28), EcuReset (0x11), LinkControl (0x87) or ControlDTCSetting (0x85) are only called once

Module version 4.3.0

2013-08-05



- ▶ ASCDCM-1088 Fixed known issue: The ECU may crash when executing service `InputOutputControlByIdentifier` (0x2F)
- ▶ ASCDCM-1074 Fixed known issue: In case `DcmDslDiagRespOnSecondDeclinedRequest` is imported as TRUE, the response buffer may get corrupted
- ▶ ASCDCM-1075 Fixed known issue: The parameter in the function prototype for `Dcm_Init()` needs to be declared as type "const"
- ▶ ASCDCM-1099 Fixed known issue: Service `WriteDataByIdentifier` (0x2E) may corrupt DID data as a result of a race condition
- ▶ ASCDCM-761 Fixed known issue: Current variable sizes are insufficient for parameters dealing with timing values
- ▶ ASCDCM-1079 Fixed known issue: Compiler error when no security level is configured (no need for security level checks) and Rte usage is enabled
- ▶ ASCDCM-1112 Fixed known issue: The Dcm Dsl uses the wrong `TxPduId` and `ConfirmationTxPduId`

Module version 4.2.0

2013-06-26

- ▶ ASCDCM-1019 Fixed known issue: The Dcm may initialize its ROE persistent data structure with corrupt values
- ▶ ASCDCM-870 Fixed known issue: `ImplementationDatatype` of type `TYPE_REFERENCE` references an `InvalidValue` when it is not allowed
- ▶ ASCDCM-1068 Fixed known issue: Mode declaration groups and prototypes are generated conditionally based on service availability
- ▶ ASCDCM-1034 Fixed known issue: Undefined identifier errors are reported when Rte usage is enabled in Dcm, the `ResponseOnEvent` (0x86) service is configured and none of the Dcm internal services is configured to use a Client/Server interface
- ▶ ASCDCM-1059 Fixed known issue: Compiler errors may be reported when the `ShortTermAdjustment` operation is enabled for a DID and one of the signals configured for this DID is of variable length and uses a ClientServer interface
- ▶ ASCDCM-1018 Fixed known issue: Optional parameter `DcmDslProtocolSessionRef` causes a code generation error if it is disabled
- ▶ ASCDCM-1023 Fixed known issue: Configuration for routines is generated incorrectly
- ▶ ASCDCM-983 Fixed known issue: Compiler errors when configuring `DcmDspRoutineSignalType` of a `DcmDspRoutineInfo` to a length other than `VARIABLE_LENGTH` when `DcmDspRoutineFixedLength` is set to FALSE in a routine which references this `DcmDspRoutineInfo`
- ▶ ASCDCM-1047 Fixed known issue: The `RoutineControl` (0x31) service fails if the complete data length configured for input signals is not a multiple of 8 bits



- ▶ ASCDCM-1026 Fixed known issue: Compilation fails if the same subfunction ID `DcmDsdSidTabServiceId` is configured for the same service `DcmDsdSidTabServiceId` inside different service tables (`DcmDsdServiceTable`)
- ▶ ASCDCM-1051 Fixed known issue: The persistent ROE initialization must be modified to reinitialize the internal variables with defaults in case of failure
- ▶ ASCDCM-884 Fixed known issue: `REQUESTCORRECTLYRECEIVEDRESPONSEPENDING` (0x78) NRCs are transmitted too late
- ▶ ASCDCM-1063 Fixed known issue: `Dcm_CopyTxData()` returns `BUFREQ_E_NOT_OK` if called with a `PduInfoPtr->SduLength` of 0
- ▶ ASCDCM-1086 Fixed known issue: The SecurityAccess (0x27) service may fail if subfunction availability (`DcmDsdSidTabSubfuncAvail`) is disabled
- ▶ Implemented `DcmModeRule` support for general service and subfunction handling
- ▶ ASCDCM-1061 Fixed known issue: Asynchronous service processing can execute forever

Module version 4.1.4

2013-05-22

- ▶ ASCDCM-1017 Fixed known issue: BswM APIs are called even though the Mode Declaration Support is used
- ▶ Added support for containers `DcmDspComControlAllChannel` and `DcmDspComControlSpecificChannel`
- ▶ ASCDCM-995 Fixed known issue: The Dcm performs Diagnostic Session checks on subfunction level correctly only for the first subfunction configured in the `DcmDsdSubService` table
- ▶ ASCDCM-1005 Fixed known issue: The Dcm performs Security Level checks on subfunction level correctly only for the first subfunction configured in the `DcmDsdSubService` table
- ▶ ASCDCM-1032 Fixed known issue: Mode Declarations are generated conditionally within Mode Declaration Groups based on subfunction configuration
- ▶ Implemented `ModeDeclarationGroupPrototypes` for service `CommunicationControl` (0x28) and invoked SchM Switch APIs for mode switch notifications
- ▶ ASCDCM-994 Fixed known issue: Configuration for Diagnostic Session and Security Level permission checks on subfunction level is not correctly generated
- ▶ Updated asynchronous service execution of services and related test cases
- ▶ Implemented usage of `BswM_Dcm_CommunicationMode_CurrentState()` for notifying ComMode to BswM within the `CommunicationControl` (0x28) service



Module version 4.1.3

2013-04-12

- ▶ ASCDCM-972 Fixed known issue: Suppress Positive Response Bit is not reset correctly
- ▶ ASCDCM-854 Fixed known issue: Service `ResponseOnEvent` returns illegal Negative Response Codes
- ▶ Implemented `ModeDeclarationGroupPrototypes` for service `ControlDTCSetting` (0x85) and invoked SchM Switch APIs for mode switch notifications

Module version 4.1.2

2013-03-19

- ▶ ASCDCM-941 Fixed known issue: Compiler errors when configuring `DcmDspRoutineInfo` containers which are not referenced from any `DcmDspRoutineInfoRef`
- ▶ ASCDCM-963 Fixed known issue: Compilation fails if asynchronous service processing is enabled and Det usage is disabled
- ▶ ASCDCM-852 Fixed known issue: The ROE persistency NV block is not initialized at first start-up of the ECU, leading to usage of corrupt data stored in the non-volatile memory
- ▶ ASCDCM-944 Fixed known issue: Active security level is set to an incorrect value
- ▶ Added DataLength parameter to interface `Xxx_ShortTermAdjustment` in case the length of the DID signal is not fixed (RfC #58643)
- ▶ ASCDCM-958 Fixed known issue: The Dcm locks after a request is rejected from the Manufacturer/Supplier Indication without any NRC being sent, if the requested service is configured to be handled asynchronously
- ▶ Changed the type of the configuration parameter `DcmDslProtocolRxComMChannelRef` from SYMBOL-IC-NAME-REFERENCE to CHOICE-REFERENCE
- ▶ ASCDCM-935 Fixed known issue: `DcmDspDidRoeActivateFnc` configuration parameter must always be configured if the `DcmDspDidExtRoe` container is enabled

Module version 4.1.1

2013-02-25

- ▶ ASCDCM-813 Fixed known issue: The operation `TriggerOnEvent` of Client-Server interface `DCM_Roe` uses the wrong `ImplementationDataType`
- ▶ ASCDCM-839 Fixed known issue: Code generation fails when Dcm is added to PduR/PduRBswModules and parameter `PduRBswModuleIsEnabled` is enabled
- ▶ ASCDCM-849 Fixed known issue: The Dcm may accept new requests while processing an ROE Type 1 event



- ▶ ASCDCM-842 Fixed known issue: Undefined identifier errors are reported when Rte usage is enabled in Dcm and none of the Dcm internal services is configured to use a Client/Server interface
- ▶ ASCDCM-848 Fixed known issue: Dcm ceases to respond to service requests if the transmission of the response to service `DiagnosticSessionControl (0x10)` fails
- ▶ ASCDCM-857 Fixed known issue: `DcmDspDataReadDataLengthFnc` always has to be configured, even if the data of the DID has a fixed length
- ▶ ASCDCM-878 Fixed known issue: Compiler error when only one protocol is used, paged buffer handling is enabled and UDS service `ReadDtcInformation (0x19)` is configured
- ▶ Added Session and Security-checks on subfunction level
- ▶ Implemented UDS service `ReadMemoryByAddress (0x23)`
- ▶ Implemented UDS service `WriteMemoryByAddress (0x3D)`
- ▶ ASCDCM-891 Fixed known issue: Warnings reported when service `ResponseOnEvent (0x86)` is not configured
- ▶ ASCDCM-892 Fixed known issue: Service `ControlDTCSetting (0x85)` can be configured without any subfunctions
- ▶ ASCDCM-872 Fixed known issue: Compiler errors may occur due to invalid code if a protocol with no `DcmDslMainConnection` is configured
- ▶ ASCDCM-907 Fixed known issue: Warnings reported when importing the Rte configuration because of an invalid reference value in `DcmIntBeh`
- ▶ ASCDCM-880 Fixed known issue: The Dcm may get blocked if it returns from the Bootloader (a Warm Start procedure) and the ComM module's Main Function is scheduled before the Dcm Main Function
- ▶ ASCDCM-916 Fixed known issue: Compiler errors due to missing declaration of the name of the multiple configuration container in `BswM_UserCallouts.c`
- ▶ Added support for container `DcmDspMemory/AddressAndLengthFormatIdentifier` as described in AUTOSAR Bugzilla RfC #53661
- ▶ Implemented `ModeDeclarationGroupPrototypes` for services `DiagnosticSessionControl (0x10)` and `EcuReset (0x11)` and invoked SchM Switch APIs for mode switch notifications
- ▶ Implemented `InputOutputControlByIdentifier (2F hex)` service
- ▶ Updated handling of service `ReadDTCInformation`, subfunction `reportDTCExtendedDataRecord-ByDTCNumber (0x19-0x06)` based on configuration parameter `DcmGetSizeOfExtended-DataRecordByDTCOptimization`, considering both AUTOSAR 4.0 rev3 and Bugzilla issue http://www.autosar.org/bugzilla/show_bug.cgi?id=52426

Module version 4.1.0

2012-10-17



- ▶ ASCDCM-708 Fixed known issue: The definition of `Dcm_NegativeResponseCodeType` is missing several response code types
- ▶ ASCDCM-709 Fixed known issue: Rte Editor reports an error if the Dcm has no Security Levels configured
- ▶ ASCDCM-720 Fixed known issue: Compiler errors occur when using certain values for `DcmDspSecurityLevel` and Rte usage is enabled
- ▶ ASCDCM-735 Fixed known issue: Compiler error in Dem because of undefined identifier `DCM_INITIAL` when Rte usage is enabled
- ▶ ASCDCM-731 Fixed known issue: Callout function `Dcm_Confirmation()` always needs to be provided, even if it is not used
- ▶ ASCDCM-710 Fixed known issue: Corruption of the response may occur when using the service `ReadDataByIdentifier`
- ▶ ASCDCM-732 Fixed known issue: `DcmDspRoutineSignalType` can be configured to `VARIABLE_LENGTH` even if `DcmDspRoutineFixedLength` is set to `TRUE`, resulting in errors during code generation
- ▶ ASCDCM-759 Fixed known issue: The parameter `DcmTaskTime` can be configured to 0 seconds
- ▶ ASCDCM-679 Fixed known issue: Positive response from UDS 0x10 is incorrect
- ▶ Implemented Migration to ASR 4.0 ComStack HandleId Policy
- ▶ ASCDCM-755 Fixed known issue: Verification of the Diagnostic Session and Security Level is also done for services `DiagnosticSessionControl (0x10)` and `SecurityAccess (0x27)`
- ▶ ASCDCM-777 Fixed known issue: Paged handling of service requests is not handled properly in case some subfunctions of `ReadDTCInformation (0x19)` are not configured
- ▶ ASCDCM-763 Fixed known issue: Wrong handling of the return status of the `Dcm_SetProgConditions()` function when returning from bootloader
- ▶ ASCDCM-780 Fixed known issue: The signature of the `Dcm_Init()` function is different from that specified in the SWS
- ▶ ASCDCM-753 Fixed known issue: If usage of the `DcmDspDataConditionCheckReadFnc()` is not enabled, the ECU may crash due to a NULL pointer access
- ▶ Implemented Asynchronous handling of diagnostic services
- ▶ The top-level structure of the software-component description in the ARXML files changed from `/AUTOSAR/Dcm` to `/AUTOSAR_Dcm`
- ▶ Updated Dcm-Dem interface to R4.0.3
- ▶ ASCDCM-811 Fixed known issue: Code generation error when external subfunction handlers are configured
- ▶ ASCDCM-778 Fixed known issue: The `DiagnosticSessionControl (0x10)` service cannot switch to Diagnostic Session Identifiers configured within the `DcmDspSession` container
- ▶ ASCDCM-794 Fixed known issue: If the configuration parameter `DcmDspDataType` is disabled, compilation fails because invalid data types are generated



- ▶ Implemented Migration to ASR 4.0 TP PDU API
- ▶ ASCDCM-819 Fixed known issue: Illegal memory access when entering Full Communication mode

Module version 4.0.15

2012-06-21

- ▶ ASCDEM-663 Fixed known issue: Dcm does not free locked NvM blocks always
- ▶ ASCDCM-683 Fixed known issue: `SwcBswMapping` is located at the wrong location in BSWMD
- ▶ ASCDCM-694 Fixed known issue: Generation of `Dcm_RoutineSesType` array not synchronized with `DcmDspRoutineInfoConfig`
- ▶ ASCDCM-682 Fixed known issue: Handling NvM Read operation with `NvM_GetErrorStatus` does not consider all positive cases
- ▶ ASCDCM-677 Fixed known issue: Inconsistent SWC-D generation when `DcmDspSecurityUsePort` is configured with certain options
- ▶ ASCDCM-654 Fixed known issue: `pMsgContext` can be overwritten before processing last service
- ▶ ASCDCM-692 Fixed known issue: Cannot configure `StopRoutine` without `DcmDspRoutineStopIn` and `DcmDspRoutineStopOut`
- ▶ ASCDCM-678 Fixed known issue: If only input or output signals are configured for routines, but not both, un compilable source code is generated
- ▶ ASCDCM-643 Fixed known issue: `ClearDiagnosticInformation (0x14)` service execution malfunctions when Rte usage is enabled
- ▶ ASCDCM-486 Fixed known issue: Importing a corrected `Dcm_swc_interface.arxml` file into Tresos Studio may results in un compilable code after Rte generation
- ▶ ASCDCM-707 Fixed known issue: Inconsistent SWC-D generation when `DcmDspSecurityUsePort` is configured with certain options

Module version 4.0.14

2012-05-16

- ▶ Initial AUTOSAR 4.0.3 version
- ▶ ASCDCM-689 Fixed known issue: Dcm code generation will fail with ERROR 1806 if a DID data configuration has, as Data Access Interface, `DcmDspDataUsePort = 'USE_DATA_SENDER_RECEIVER'`

3.3.1.2. New features

- ▶ Implemented `DcmModeRule` support for general service and subfunction handling



- ▶ Implemented `DcmModeRule` support
- ▶ Implemented `DcmModeRule` support for controlling individual routines
- ▶ Implemented `DcmModeRule` support for reading or writing individual `DcmDspMemory` ranges
- ▶ Implemented `DcmModeRule` support for reading, writing and controlling individual DIDs
- ▶ Implemented `DcmModeRule` support for re-enabling of DTC setting
- ▶ Implemented `DcmModeRule` support for resetting the state of the `CommunicationControl` service
- ▶ Implemented configuration parameter support for the `DynamicallyDefineDataIdentifier` (2C hex) service
- ▶ Implemented UDS service `ReadDataByPeriodicIdentifier` (2A hex)
- ▶ Implemented UDS service `DynamicallyDefineDataIdentifier` (2C hex)
- ▶ Implemented support for the configuration parameter `DcmDspDataConditionCheckReadFncUsed`, as described in Autosar Bugzilla RfC #53669
- ▶ Implemented ISO14229-1:2013 Response Message Format (UUDT Identifiers) for the diagnostic service `ReadDataByPeriodicIdentifier` (0x2A) responses.
- ▶ Implemented support for TYPE2 periodic responses.
- ▶ Implemented support for the `RoutineServices` API according to AUTOSAR 4.2.1 as a per routine selectable alternative to the existing `RoutineServices` API implementation according to AUTOSAR 4.0.-3 The implementation is based on Autosar Bugzilla RFC #57860
- ▶ Implemented support for configuration parameter `DcmDspDidRange`.
- ▶ Implemented support for the OBD services \$03 (Request emission-related diagnostic trouble codes), \$06 (Request on-board monitoring test results for specific monitored systems), \$07 (Request emission-related diagnostic trouble codes detected during current or last completed driving cycle) and \$0A (Request emission-related diagnostic trouble codes with permanent status).
- ▶ Implemented support for the `SecurityAccess` API according to AUTOSAR 4.2.1 as a configuration selectable alternative to the existing `SecurityAccess` API implementation according to AUTOSAR 4.0.3
- ▶ Implemented support for the `DataServices` API according to AUTOSAR 4.2.1 as a configuration selectable alternative to the existing `DataServices` API implementation according to AUTOSAR 4.0.3
- ▶ Implemented support for the OBD services \$04 (Clear/reset emission-related diagnostic information), according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented support for the `DCM_Roe ClientServerInterface` according to AUTOSAR 4.2.1 by
 1. having the `RoeEventId` parameter defined as a Portdefined argument value
 2. changing the name of the generated P-port to `DCM_Roe_{RoeName}`

as a configuration selectable alternative to the existing `DCM_Roe ClientServerInterface` implementation according to AUTOSAR 4.0.3



- ▶ Implemented support for suppressed NRC of functional requests according to ISO14229-2006 and ISO14229-2013 releases.
- ▶ Implemented support for the OBD service \$09 (Request Vehicle Information)
- ▶ Implemented support for OBD Service \$08 - Request Control of On-Board System, Test or Component
- ▶ Implemented Calibration of Dcm Services
- ▶ Implemented support for DTCSettingControlOptionRecord and LengthCheck for ControlDTCSetting service request according to AUTOSAR 4.2.1.
- ▶ Implemented support for the ReadGenericInformation (0xAF) service according to AUTOSAR 4.4.0, depending on the existence of a valid license.
- ▶ Implemented support for reading OBD data identifiers using service ReadDataByIdentifier (\$22):

For range F400..F4FF, the data for the DID is obtained as defined for OBD Service \$01 (PID)

For range F600..F6FF, the data for the DID is obtained as defined for OBD Service \$06 (OBD MID)

For range F800..F8FF, the data for the DID is obtained as defined for OBD Service \$09 (InfoType)

- ▶ Implemented support for protocol preemption:

Parameter DcmDslProtocolPreemptTimeout is now supported and required to be higher than 0

Parameter DcmDslDiagRespOnSecondDeclinedRequest is now supported

- ▶ Implemented UUDT transmission and AUTOSAR 4.2.1 - compatible handling for UDS service ReadData-ByPeriodicIdentifier (0x2A)
- ▶ Updated #defines for symbolic name values according to AUTOSAR 4.0.3 naming schema
- ▶ Implemented support for disabling ECU keep awake through active diagnosis after ignition was turned off by introducing the interface `Xxx_SetActiveDiagnostic()` that can be used by the application to inform the Dcm about the desired `ActiveDiagnostic` status. The status `DCM_COMM_ACTIVE` will keep the ECU awake by diagnosis while the status `DCM_COMM_NOT_ACTIVE` will prevent the ECU from being kept awake by diagnosis. This functionality can be turned on or off by setting the configuration item `DcmDslEnableSetActiveDiagnosis` to true or false.
- ▶ Implemented optimization of DID data configuration code

DID data configuration code now contains only the available operations function pointers

- ▶ SecurityAccess service (0x27) returns NRC 0x36 (exceededNumberOfAttempts) one time when SendKey subfunction is requested with an invalid key and the maximum number of failed access attempts is reached.

The number of failed access attempts is reset after a correct SecurityAccess sequence (successful Re-questeSeed followed by successful SendKey)

- ▶ Implemented separate ROE Service subfunctions taking in consideration the StorageState bit (sub-function bit 6). The implementation is based on Bugzilla RfC #72061



- ▶ Implemented a subset of UDS service 0x86 (ResponseOnEvent) according to AUTOSAR 4.2 specification. The implemented subfunctions are: 0x00 stopResponseOnEvent, 0x04 reportActivatedEvents, 0x05 startResponseOnEvent for preconfigured events of OnChangeOfDataIdentifier type with Infinite(0x02) eventWindowTime.
- ▶ Implemented separate ROE Service subfunctions taking in consideration the StorageState bit (sub-function bit 6). The implementation is based on Bugzilla RfC #72061
- ▶ Implemented full support for the eventWindowTime parameter for the UDS service 0x86 (ResponseOnEvent): Infinity(0x02), CurrentCycle(0x03) and CurrentAndFollowingCycle(0x04)
- ▶ Implemented subfunction ClearResponseOnevent(0x06) of service ResponseOnEvent(0x86)
- ▶ Implemented RequestDownload (0x34) Service NRC handling according to ISO14229-1:2013.
- ▶ Implemented RequestUpload (0x35) Service NRC handling according to ISO14229-1:2013.
- ▶ Implemented TransferData (0x36) Service NRC handling according to ISO14229-1:2013.
- ▶ Implemented RequestTransferExit (0x37) Service NRC handling according to ISO14229-1:2013.
- ▶ Implemented subfunction onDTCStatusChange(0x01) of UDS service ResponseOnEvent(0x86) according to AUTOSAR 4.2 specification.
- ▶ Implemented TYPE2 transmission for UDS service ResponseOnEvent(0x86).
- ▶ Implemented services RequestDownload (0x34), RequestUpload (0x35), TransferData (0x36), RequestTransferExit (0x37), ReadMemoryByAddress (0x23) and WriteMemoryByAddress (0x3D) according to AUTOSAR 4.3.0.
- ▶ Implemented subfunctions enableRxAndDisableTxWithEnhancedAddressInformation (0x04) and enableRxAndTxWithEnhancedAddressInformation (0x05) of service CommunicationControl(0x28).
- ▶ Implemented support for security access delay timers according to AUTOSAR 4.3.0
- ▶ Allow configuration of external subfunction handlers for UDS service TesterPresent (0x3E). Adapted NRC sequence of UDS service TesterPresent (0x3E) to ISO 14229-1:2013 requirements
- ▶ Implemented the capability to configure Memory ranges of Services 0x23/0x3D to be defined with labels
- ▶ Implemented Sender/Receiver access for DcmDspData for UDS services ReadDataByIdentifier (0x22) and WriteDataByIdentifier (0x2E)
- ▶ Implemented endianness interpretation for DcmDspData with Sender/Receiver access for UDS services ReadDataByIdentifier (0x22) and WriteDataByIdentifier (0x2E) according to AUTOSAR 4.3 specifications
- ▶ Implemented support for BSW distribution, when Dcm and ComM are on different partitions (cores)
- ▶ Add support for short response on sub-function 0x01 - onDTCStatusChange for the UDS service ResponseOnEvent (0x86)

That should provide the following format :

byte 1: ResponseOnEvent Response SID



byte 2: eventType

byte 3: numberOfldentifiedEvents

byte 4: eventWindowTime

- ▶ Implemented support to enable notification callout for S3 timeout.
- ▶ Implemented support for configuration parameter `DcmDspMemoryRangeHighNotIncluded` to allow for more flexibility in the definition of memory ranges
- ▶ Implemented support for notification callout on each `Dcm_TpTxConfirmation` call.
- ▶ Implemented support for configuration parameter `DcmDsdDisableGenericServiceImplementation` to allow DCM internal service processing in case of project specific service handling.
- ▶ Implemented parameter `DcmDslObdNRCResponseSupression` to allow for selectable NRC behaviour when OBD services are requested but not enabled
- ▶ Implemented service callbacks interfaces according to AUTOSAR 4.2.2 for `WriteMemory` and `ProcessRequestTransferExit`
- ▶ Added generic interface to inject service requests from application software.
- ▶ Implemented array type signals for the UDS service `RoutineControl` (0x31) according to AUTOSAR 4.3.0
- ▶ Implemented Generic End Of Line addon for Routine Control service (0x31). If `DcmDspRoutineEnableDefaultInterfaces` configuration parameter is set to TRUE, for every routine which is not configured or not used, Dcm module shall call the Default Interfaces for Start, Stop and RequestResults sub-services from application.
- ▶ Implemented Routine Info Byte handler for Routine Control service (0x31). If `DcmDspRoutineEnableRoutineInfoByte` configuration parameter is set to TRUE and if the requested routine returns `E_OK`, a byte can be added into the response by calling the interface `Dcm_AddRoutineInfoByte()` defined in Application.
- ▶ Implemented Generic End Of Line addon for Data Identifiers for Service `ReadDataByIdentifier` (0x22), Service `WriteDataByIdentifier` (0x2E) and `InputOutputControlByIdentifier` (0x2F). Behaviour to be activated via vendor-specific configuration flag. If `DcmDspDidEnableDefaultInterfaces` configuration parameter is set to TRUE, for every DID which is not configured or not used, nor in the range of a configured "DcmDspDidRange", the Dcm module shall call the Default Interfaces for the specific request.
- ▶ Implemented Parallel UDS and OBD processing/ASR4.3 Dcm/Dem interface with ClientID
- ▶ Implemented sub-services `reportNumberOfDTCBySeverityMaskRecord(0x07)`, `reportDTCBySeverityMaskRecord(0x08)`, `reportSeverityInformationOfDTC(0x09)` for UDS Service `ReadDTCInformation` (0x19), used to: - retrieve the number of DTCs matching a client defined severity status mask record `reportNumberOfDTCBySeverityMaskRecord(0x07)`; - retrieve a list of DTC severity and functional unit information, which satisfy a client defined severity mask record `reportDTCBySeverityMaskRecord(0x08)`; - retrieve severity and functional unit information for a client de-



fined DTCMaskRecord reportSeverityInformationOfDTC(0x09). The mentioned above subservices are implemented according to AUTOSAR 4.4.0 specifications.

- ▶ Implemented sub-service reportNumberOfDTCByStatusMask(0x01) for UDS Service ReadDTCInformation (0x19), used to retrieve the number of DTCs matching a client defined status mask according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented sub-service reportDTCByStatusMask(0x02) for UDS Service ReadDTCInformation (0x19), used to retrieve a list of DTCs matching a client defined status mask according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented sub-service reportDTCSnapshotIdentification(0x03) for UDS Service ReadDTCInformation (0x19), used to retrieve the DTCSnapshot record identification according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented sub-service reportDTCSnapshotRecordByDTCNumber(0x04) for UDS Service ReadDTCInformation (0x19), used to retrieve the DTCSnapshot record data for a client defined DTC mask according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented sub-service reportDTCExtDataRecordByDTCNumber(0x06) for UDS Service ReadDTCInformation (0x19), used to retrieve the DTCExtendedData record data for a client defined DTC mask and a client defined DTCExtendedData record number according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented sub-service reportSupportedDTC(0x0A) for UDS Service ReadDTCInformation (0x19), used to retrieve the status of all DTCs supported by the server according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented sub-service reportFirstTestFailedDTC (0x0B) for UDS service ReadDTCInformation (0x19), used to retrieve the first failed DTC according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented sub-service reportFirstConfirmedDTC (0x0C) for UDS service ReadDTCInformation (0x19), used to retrieve the first detected confirmed DTC according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented sub-service reportMostRecentTestFailedDTC (0x0D) for UDS service ReadDTCInformation (0x19), used to retrieve the most recent failed DTC according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented sub-service reportMostRecentConfirmedDTC (0x0E) for UDS service ReadDTCInformation (0x19), used to retrieve the most recently detected confirmed DTC according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented sub-service reportEmissionsOBDDTCByStatusMask(0x13) for UDS Service ReadDTCInformation (0x19), used to retrieve the list of "only emissions-related OBD" DTCs that match a client defined status mask according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented sub-service reportDTCFaultDetectionCounter(0x14) for UDS Service ReadDTCInformation (0x19), used to retrieve a list of "prefailed" DTC status according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented sub-service reportDTCWithPermanentStatus(0x15) for UDS Service ReadDTCInformation (0x19), used to retrieve a list of DTCs with "permanent DTC" status according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented sub-service reportUserDefMemoryDTCByStatusMask(0x17) for UDS Service ReadDTCInformation (0x19), used to retrieve the list of DTCs out of the server's user defined DTC memory that match a client defined DTC status mask according to AUTOSAR 4.4.0 specifications.



- ▶ Implemented sub-service reportUserDefMemoryDTCSnapshotRecordByDTCNumber(0x18) for UDS Service ReadDTCInformation (0x19), used to retrieve user defined memory DTCSnapshot record data for a client defined DTC mask and a client defined DTCSnapshotNumber out of the DTC user defined memory according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented sub-service reportUserDefMemoryDTCExtDataRecordByDTCNumber(0x19) for UDS Service ReadDTCInformation (0x19), used to retrieve user defined memory DTCExtendedData record data for a client defined DTC mask and a client defined DTCExtendedData record number out of the DTC memory according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented support for the UDS Service ClearDiagnosticInformation (0x14), used to clear diagnostic information in one or multiple server's memory according to AUTOSAR 4.4.0 specifications.
- ▶ Implemented support for the Request Vehicle Information (0x09) according to AUTOSAR 4.4.0 as a configuration selectable alternative to the existing Request Vehicle Information (0x09) implementation according to AUTOSAR 4.0.3.
- ▶ Implemented support for parallel processing of multiple diagnostic requests, the EB Dcm solution provides the possibility of configuring protocol stack number. A protocol stack number in the EB Dcm solution is used in order to define the behavior of processing two protocols. Protocols on the same stack will obey the preemption mechanism. Two protocols on different stacks may run in parallel, unless they are of the same kind (OBD/UDS). In order for two protocols to run in parallel, different stack numbers and different types for the two protocols shall be configured for them. Each protocol has a configured DcmDemClientRef defining the Dem client interacting with the Dem module. This client Id allows the Dem module to distinguish between concurrent calls of the Dcm module of the same function or set of functions to process a certain request. Parallel processing is currently limited to an OBD diagnostic request with an UDS diagnostic request.
- ▶ Implemented support for Pdu metadata. Metadata information is retrieved using the EB EcuC APIs: - EcuC_GetMetaDataSourceAddr() - EcuC_GetMetaDataTableAddr() Metadata information is provided to the lower layer using the EB EcuC APIs: - EcuC_SetMetaDataSourceAddr() - EcuC_SetMetaDataTableAddr() and also provided to the Xxx_Indication/Xxx_Confirmation operations of the ServiceRequestNotification interface.
- ▶ Implemented support for the UDS service ReadScalingDataByIdentifier (0x24) according to AUTOSAR 4.0.3 and R20-11.

3.3.1.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- ▶ Diagnostic application access without Rte

Description:

Diagnostic applications which do not use the AUTOSAR Rte interface are supported. Dcm allows the configuration of user defined handlers for diagnostic services.

**Rationale:**

Compatibility with legacy diagnostic applications.

- ▶ Addition of parameter `DcmDslProtocolRxPduId`

Description:

Parameter `DcmDslProtocolRxPduId` is added as a Vendor Specific parameter. It is not present in the Dcm SWS of AUTOSAR R3.0 and R3.1.

Rationale:

This parameter is essential for the functionality of Dcm.

- ▶ Provision for handling all session and security levels

Description:

The configuration of `DcmDspSecurityRow` with the parameter `DcmDspSecurityLevel` value 255 shall be used for configurations which support all security levels. The configuration of `DcmDspSessionRow` with the parameter `DcmDspSessionLevel` value 255 shall be used for configurations which can be executed in all sessions.

Rationale:

This makes the configuration more convenient for the user.

- ▶ Extension of `Dcm_NegativeResponseCodeType`

Description:

The type definition of `Dcm_NegativeResponseCodeType` has been extended to include these values also:

- ▶ `DCM_E_SERVICENOTSUPPORTED` (0x11)
- ▶ `DCM_E_SUBFUNCTIONNOTSUPPORTED` (0x12)
- ▶ `DCM_E_INCORRECTMESSAGELENGTHORINVALIDFORMAT` (0x13)
- ▶ `DCM_E_RESPONSETOOLONG` (0x14)
- ▶ `DCM_E_INVALIDKEY` (0x35)
- ▶ `DCM_E_EXCEEDNUMBEROFTTEMPTS` (0x36)
- ▶ `DCM_E_REQUIREDTIMEDELAYNOTEEXPIRED` (0x37)
- ▶ `DCM_E_UPLOADDOWNLOADNOTACCEPTED` (0x70)
- ▶ `DCM_E_TRANSFERDATASUSPENDED` (0x71)
- ▶ `DCM_E_WRONGBLOCKSEQUENCECOUNTER` (0x73)



- ▶ DCM_E_REQUESTCORRECTLYRECEIVEDRESPONSEPENDING (0x78)
- ▶ DCM_E_SERVICENOTSUPPORTEDINACTIVESESSION (0x7F)

Rationale:

AUTOSAR does not specify all the NRCs possible within `Dcm_NegativeResponseCodeType`. It is not present in the Dcm SWS of AUTOSAR R3.0 and R3.1.

- ▶ Usage of `DCM_FORCE_RCRRP_OK` in external service handlers

Description:

As the `opStatus` parameter `DCM_FORCE_RCRRP_OK` shall not be passed to External service handlers, `DCM_PENDING` shall be passed instead.

Rationale:

The Dcm cannot use the `opStatus` parameter `DCM_FORCE_RCRRP_OK` in the API `<Module>_<DiagnosticService>`.

- ▶ Asynchronous handling of diagnostic services.

Description:

The Dcm will be able to handle service execution asynchronously, within another task separate from the Dcm task.

Rationale:

The Dcm module can now execute services which contain routines that may take longer to run than the Dcm task time and may not be interruptable in the way that they can be executed in steps, each step returning a PENDING status.

- ▶ Multiple `DcmDslMainConnections` per `DcmDslProtocolRow`.

Description:

The Dcm will be able to handle multiple `DcmDslMainConnections` per `DcmDslProtocolRow`.

Rationale:

Eliminating the need to duplicate protocol information when multiple connections require the use of the same protocol

- ▶ Routine Control variable length signals.

Description:

Configuration parameter `DcmDspRoutineVariableLengthInBytes` is added, by which the user can specify whether the variable length signals for the Routine Control are expressed in bits or bytes. This



parameter set to 'true' indicates that the variable length signals are expressed in Bytes and the value 'false' indicates signals in bits.

Rationale:

Bugzilla entry http://www.autosar.org/bugzilla/show_bug.cgi?id=55779.

- ▶ Mixed signal types in RoutineControl (0x31) operations

Description:

The Dcm will be able to handle mixed signal types (Variable and Fixed length) for input and output signals of the RoutineControl (0x31) service operations. The DcmDspRoutineFixedLength parameter for a routine shall only control whether the currentDataLength INOUT parameter is used in the interfaces generated for the RoutineControl operations.

Rationale:

Allowing more flexibility in configuring signals used in the interfaces for this service.

- ▶ Dem_DcmCancelOperation() is called if service ClearDiagnosticInformation (0x14) times out

Description:

Dem_DcmCancelOperation() is now called in case the service ClearDiagnosticInformation (0x14) times out as a result of too many REQUESTCORRECTLYRECEIVEDRESPONSEPENDING NRCs having been sent, in order to cancel pending operations of the Dem.

Rationale:

Allowing an asynchronous Dem_ClearDTC() job to be canceled.

- ▶ Parameter DcmDspRoePreconfRxConnectionRef is now available for preconfiguration of ROE events

Description:

The DcmDs1ProtocolRx referenced by DcmDspRoePreconfRxConnectionRef shall be considered as the reception connection on which configuring messages for the preconfigured ROE event would have arrived. This is necessary as the following shall be taken from this connection:

- ▶ The RxPduId for the preconfigured ROE event that is used to determine the TxPduId for sending during Type1 transmissions.
- ▶ The ComM channel ID for this transmission.

Rationale:

Ensuring a correct and consistent ROE event preconfiguration.

- ▶ Input signals are now allowed for the requestRoutineResults (0x03) sub-function of the RoutineControl (0x31) diagnostic service

**Description:**

Input signals are now available for the `requestRoutineResults` (0x03) sub-function of the `RoutineControl` (0x31) diagnostic service, by configuring the `DcmDspRoutineRequestResIn` container within the `DcmDspRoutineInfo` referenced by the routine's `DcmDspRoutine` configuration. The configuration process is identical to the configuration of input signals for the other sub-functions of the service, and results in similar interface signatures.

Rationale:

Allowing greater flexibility in configuring operations for the `RoutineControl` (0x31) service.

- ▶ Availability of services in the default session

Description:

The services `SecurityAccess` (0x27), `CommunicationControl` (0x28), `ControlDTCSetting` (0x85), `ReadDataByPeriodicIdentifier` (0x2A) and `InputOutputControlByIdentifier` (0x2F) can now be configured to be available in the default session.

Rationale:

Allowing this configuration in order to provide maximum flexibility. Should an ISO 14229-1 compatible behavior be required, explicitly configure these services to be unavailable in the default session.

- ▶ Only one execution at a time of the service handler function is carried out within the context of a separate task when using asynchronous service handling

Description:

Improved the asynchronous processing of service handlers by breaking the continuous processing within a loop inside a separate task into discreet calls to the service handler function. Each call is executed within a separate task instead of the entire service processing being run in one continuous loop.

Rationale:

The asynchronous service handler task is now allowed to yield after each service handler execution cycle , thus allowing tasks of a lower priority than the asynchronous service processor task to also be scheduled, as they will no longer be blocked until the service handling is completed.

- ▶ The DIDs in System Supplier Specific range are accessible by `WriteDataByIdentifier` (0x2E) service.

Description:

The DIDs in System Supplier Specific range (0xFD00..0xFEFF) are accessible by `WriteDataByIdentifier` (0x2E) service. AUTOSAR does not specify any special handling for this range.



Rationale:

AUTOSAR does not restrict the access to this DID range although the handling is not explicitly stated for the DIDs in this range.

- ▶ Transmission latency is reduced by triggering transmission of periodic DID samples as early as possible.

Description:

Improved transmission latency by triggering transmission of available PDID samples in the same cycle previous transmissions are confirmed

Rationale:

To avoid wasting bandwidth the transmission of samples that are waiting in the transmission ring buffers are triggered as soon as possible. This is done as soon as all pending confirmations of previous transmissions are received.

- ▶ The protocol for processing a periodic transmission is no longer chosen based on deprecated configuration item `DcmDslProtocolTransType`.

Description:

The protocol for processing periodic transmissions is now selected based on the presence of `DcmDslPeriodicTransmissionConRef` or `DcmDslROEConnectionRef` in the `DcmDslMainConnection` on which a periodic transmission is configured. The `TxPdulds` and `TxConfirmationPdulds` for periodic transmissions too are chosen based on those connection references and the availability of the `Pdulds` (`DcmDslPeriodicTxConfirmationPduld` and `DcmDslPeriodicTxPduRef`) in the referenced `DcmDslPeriodicTransmission` connection, or the equivalent `DcmDslResponseOnEvent`. for ROE

Rationale:

`DcmDslProtocolTransType` is deprecated as it is redundant. Additionally the new method of selecting the periodic transmission protocol provides more flexibility.

- ▶ Addition of parameter `DcmDspGenerateOnlyNeededIOControlOperations`

Description:

Parameter `DcmDspGenerateOnlyNeededIOControlOperations` allows to generate only Client-Server Interface of used IOControl operations.

- ▶ If `DcmDspGenerateOnlyNeededIOControlOperations` is set to TRUE and any DID that contains a `DcmDspDidSignal` which has client server data access interface and also references a `DcmDspData` element that in turn references a `DcmDspDidInfo` which has the `DcmDspDidControl` container enabled, only necessary Client-Server Interfaces of used IOControl operations are generated.
- ▶ If `DcmDspGenerateOnlyNeededIOControlOperations` is set to FALSE and any DID that contains a `DcmDspDidSignal` which has client server data access interface and also references a `DcmD-`



`spData` element that in turn references a `DcmDspDidInfo` which has the `DcmDspDidControl` container enabled, all Client-Server Interfaces of IOControl operations are generated.

Parameter `DcmDspGenerateOnlyNeededIOControlOperations` is not present in the Dcm SWS of AUTOSAR.

Rationale:

To avoid generating unrelated Client-Server Interfaces of IOControl operations.

- ▶ Preserve data at buffer pointers provided to DID operations across asynchronous operation call sequences

Description:

The signal buffer for input/output operations for DIDs is maintained consistent throughout the sequence of calls constituting an asynchronous operation: Every new call to the interface which is part of the same operation (subsequent to the interface having returned a `DCM_E_PENDING` error status), the `ControlOptionRecord` pointer in the case of `Xxx_ShortTermAdjustment`, or the `Data` pointer in the case of `Xxx_ReadData` or `Xxx_WriteData`, shall have the same address and point to the same data it had when the previous invocation of the operation finished. UDS services `ReadDataByIdentifier` (0x22), `WriteDataByIdentifier` (0x2E), `InputOutputControlByIdentifier` (0x2F) and the periodic DID sampling as a result of service `ReadDataByPeriodicIdentifier` (0x2A) usage are affected.

Rationale:

Autosar 4.2.2 specifies this behaviour, which is developed in RfC #62898.

- ▶ Transmit NRC 0x36 (`exceededNumberOfAttempts`) for `SecurityAccess` service (0x27) one time, when `SendKey` subfunction is requested with an invalid key and the maximum number of failed access attempts is reached

Description:

NRC 0x36 (`exceededNumberOfAttempts`) for `SecurityAccess` service (0x27) is transmitted one time, when `SendKey` subfunction is requested with an invalid key and the maximum number of failed access attempts is reached. The next requests will receive NRC `RequiredTimeDelayNotExpired` until the required delay time passes. The number of failed access attempts is reset after a correct `SecurityAccess` sequence (successful `RequesteSeed` followed by successful `SendKey`).

Rationale:

ISO 14229-1 specifies this behaviour.

- ▶ When requesting UDS service `ReadDataByIdentifier` (0x22), if for a DID the `DcmDspDataConditionCheck-ReadFnc` function or the `ConditionCheckRead` operation returns `DCM_E_NOT_OK` and a NRC equal to 0x00, the DID is treated as an unsupported DID.

Description:



When requesting UDS service ReadDataByIdentifier (0x22), if the DcmDspDataConditionCheckReadFnc function or the ConditionCheckRead operation for a DID returns DCM_E_NOT_OK and a NRC equal to 0x00, the DID is treated as an unsupported DID. This means that if none of the requested DIDs is supported, the DCM module will transmit a negative response with NRC 0x31 (Request out of range). If at least one DID is supported, the DCM module will transmit a positive response and the DIDs which received DCM_E_NOT_OK will not be part of the response.

Rationale:

This is an addition to the Autosar specifications to specify the behavior of the DCM module in case the DcmDspDataConditionCheckReadFnc function or the ConditionCheckRead operation for a DID returns DCM_E_NOT_OK and a NRC equal to 0x00.

- ▶ All combinations of signal configurations for all RoutineControl (0x31) operations are now available

Description:

This means that all RoutineControl (0x31) operations for any DcmDspRoutine can have any number of (including zero) signals configured for input and/or output.

Rationale:

This is an addition to the Autosar specifications to specify the behavior of the DCM module for all combinations of signals for all RoutineControl operations, this allows maximum flexibility in configuring routines.

- ▶ Diagnostic session checks are performed on subfunctions of service DiagnosticSessionControl too

Description:

Although not required by AUTOSAR, to allow for more flexibility the subfunction of UDS service DiagnosticSessionControl can now be assigned diagnostic sessions and they can only be called from those diagnostic sessions.

Rationale:

Added flexibility for allowed session transitions.

- ▶ DIDs may contain a variable length signal as the last element even if there are more than one signals in that DID.

Description:

Older versions of AUTOSAR specified that a DID only may contain a variable-length signal if that signal was the only one contained in that DID. Latest AUTOSAR versions relax that restriction and allow more than one fixed-length signals before the variable-length signal which must be the last one.

Rationale:



Offer more flexibility to the user and support the latest Autosar specification.

- ▶ The NRC handling for RoutineControl(0x31) service is made according with selected ISO(ISO_-14229-2006/ISO_14229-2006).

Description:

If ISO_14229-2006 is used and UDS service RoutineControl(0x31) is requested the NRC requestSequenceError(0x24) is return by DCM module according with ISO_14229-2006. If ISO_-14229-2013 is used and UDS service RoutineControl(0x31) is requested the next checks needs to be handle by application "routineControlOptionRecord contains valid data for the requested RID", "Condition check" and "Request sequence respected for the RID?" to be according with ISO 14229-1:2013 and AUTOSAR 4.2.1.

Rationale:

Offer more flexibility to the user and support the latest AUTOSAR specification.

- ▶ The NRC handling for EcuReset(0x11) service improves to produce NRC Conditions Not Correct (0x22) in the event that the result of the switch of the DcmEcuReset mode declaration group is different than SCHM_E_OK.

Description:

AUTOSAR 4.2.2 SWS does not consider the result of the switch of the DcmEcuReset mode declaration group before triggering the positive response. This improvement introduces the posiblity of producing a negative response whenever conditions are met to do so, and so provide information to the user about the working conditions of the system.

Rationale:

Offer more flexibility to the user and support the latest AUTOSAR specification.

- ▶ Introduced a new configuration parameter `DcmDspDiagSesRespMaxNumRespPend`. This offers more flexibility in configuring the maximum amount of `RequestCorrectlyReceivedResponsePending` (0x78) NRCs allowed in a given diagnostic session.

Description:

A new configuration parameter `DcmDspDiagSesRespMaxNumRespPend` is introduced for each element of the `DcmDspSessionRow` container. If the parameter is not present (is disabled) the functionality of the Dcm module is unchanged. If the parameter is enabled for a certain session and a request in that session is being processed, the maximum number of the consecutive responses with the NRC `RequestCorrectlyReceivedResponsePending`(0x78) will depend on the value of the new parameter, i.-e. this parameter overrides the value configured in the `DcmDslDiagRespMaxNumRespPend` parameter while the ECU is in the session for which the `DcmDspDiagSesRespMaxNumRespPend` parameter is configured.

**Rationale:**

Offers more flexibility to the user.

- ▶ Allowed configuration of DIDs within the 0x0000 - 0x00FF range. Allowed the usage of services ReadDataByIdentifier (0x22) and WriteDataByIdentifier (0x2E) for DIDs in this range.

Description:

Some limitations placed upon the readable/writeable DID ranges were quite restrictive for the user. This is why DIDs within the 0x0000 - 0x00FF range are now allowed for configuration and read/write operations even though they are defined as reserved by ISO_14229-2006. The responsibility of making sure that the DIDs are allowed to be accessed in compliance to the ISO specification passes to the user.

Rationale:

Offers more flexibility to the user.

- ▶ Introduced usage of HandleId wizard for RxPdulds and TxConfirmationPdulds.

Description:

Allows the user to generate zero based, ascending and consecutive values for the DcmDslProtocolRx-Pduld configuration nodes. Allows the user to generate zero based, ascending and consecutive values for the DcmDslTxConfirmationPduld, DcmDslPeriodicTxConfirmationPduld, DcmDslRoeTxConfirmation-Pduld configuration nodes that are handled as a group with unique values. This could help the user to prevent entering values manually for the above mentioned configuration parameters. This is especially helpful for large configurations.

Rationale:

Simplify the module configuration for the user.

- ▶ Introduce usage of a notification callout for S3 timeout.

Description:

Allows the user to enable a callout, to be notified when S3 timeout occurs.

Rationale:

The user application is able to track S3 timeout.

- ▶ Introduce usage of a notification callout for every transmission confirmation .

Description:

Allows the user to enable a callout, to be notified of Dcm_TpTxConfirmation.

Rationale:



The user application is able to track order and variability of transmissions.

- ▶ Introduce parameter `DcmDsdDisableGenericServiceImplementation`.

Description:

This parameter allows to control the availability of the DCM Internal Service Handler for a Service Handler in which the External Service Handler option has been enabled.

Rationale:

The user can take advantage of the already available functionality in the context of developing a proprietary implementation of a particular service, by properly making use of this new parameter.

- ▶ Introduce new configuration parameter `DcmDslDiagRespMaxNumRespPendInfinityLimit`.

Description:

- ▶ If the configuration parameter `DcmDslDiagRespMaxNumRespPendInfinityLimit` is set to value 255, then the parameter `DcmDslDiagRespMaxNumRespPend` and `DcmDspDiagSesRespMaxNumRespPend` can have values from 0 to 255 (inclusive), where the value 255 is treated as the infinity limit for transmitting NRC 0x78 by the Dcm.
- ▶ If the configuration parameter `DcmDslDiagRespMaxNumRespPendInfinityLimit` is set to "65535", then the parameter `DcmDslDiagRespMaxNumRespPend` and `DcmDspDiagSesRespMaxNumRespPend` can have values from 0 to 65535 (inclusive), where the value 65535 is treated as the infinity limit for transmitting NRC 0x78 by the Dcm.

Rationale:

To increase the maximum number of requestCorrectlyReceived-ResponsePending (0x78) NRCs to be sent by the Dcm from 254 to 65534.

- ▶ Expected behavior of `Dem_DcmGetDTRData()` for OBD service \$06 Request On-Board Monitoring Test-results for Specific Monitored Systems

Description:

When configuration parameter `DcmOBDMIDSupportType` is set to `DCM_OBDMID_SUPPORT_DEM`:

- ▶ The configuration of OBDMID, TID and UaSID of a DTR is part of Dem configuration.
- ▶ Dcm retrieves DTR data from DEM using `Dem_DcmGetDTRData()`.

`Dem_DcmGetDTRData()` shall return `E_NOT_OK` and Dcm shall not append the reported data to the response message, if:

- ▶ Report of DTR result fails.
- ▶ DTR status is invisible.



`Dem_DcmGetDTRData()` shall return `E_OK` and Dcm shall append the reported data to the response message, if report of DTR result is successful:

- ▶ If DTR data is available, Dem shall report DTR data along with TID and UaSID and test result with lower and upper limit.
- ▶ If DTR data is not available, Dem shall report DTR data along with TID and UaSID and shall set test result with lower and upper limit to `0x00`.

Rationale:

ISO-15031-5: Test result with lower and upper limit set to `0x00`, when no data is available.

▶ S3 server handling

Description:

Restrict S3 server handling to requests, which are received on the same `DcmDslMainConnection` as the one on which the `DiagnosticSessionControl` (0x10) request was received, which has changed the session to the non-default session.

Rationale:

- ▶ AR-94821: Clarify whether requests on other protocols than the one which set the default session have any effect upon the S3 timer
 - ▶ AR-94194: Clarify Multiple client handling
- ▶ Dcm Endianess Conversion

Description:

Total signal buffer length argument in `Dcm_Dsp_DidSignals_Endianness_Conversion()` changed to signal data length (`DCM_DATA_MAX_SIZE_BYTE_ARRAY` is replaced by `ReadContext->SignalLength`).

Rationale:

- ▶ Performance optimization
- ▶ DcmDemClientRef multiplicity

Description:

Multiplicity of `DcmDemClientRef` changed from 1 to 0..1

Rationale:

- ▶ The Dcm should be able to generate and compile with no Dem module/stub present regardless of the availability of UDS Services 0x19, 0x14, 0x85 or OBD Services.



3.3.1.4. Deviations

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

- ▶ Parameter DcmDslProtocolRxBufferID

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Parameter DcmDslProtocolRxBufferID is moved to the Rx connection configuration level in the schema tree.

Rationale:

The location as given by AUTOSAR specifies one Rx buffer for all Rx connections of one protocol. This makes it complex to configure the Dcm. Instead, a separate buffer can be configured for each Rx connection.

Requirements:

Dcm701_Conf

- ▶ Unsupported AUTOSAR parameters

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Not all configuration parameters are supported by the Dcm implementation. Unsupported parameters are not editable and a note is added to their description.

Requirements:

Dcm936_Conf, Dcm825_Conf, Dcm600_Conf

- ▶ Symbolic port names used instead of numeric names

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Rte ports are named by their symbolic short name taken from the configuration.

Rationale:



Symbolic port names do not change when ports are deleted or inserted. Also the symbolic name can be chosen to reflect the purpose which makes the port connection process easier and less error prone. But numeric names get renumbered and need to be re-connected.

Requirements:

Section 8.8

- ▶ Paged buffered transmission is supported only for service `ReadDTCInformation` (0x19) (reference to product description: ASCPD-41)

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Paged buffer response transmissions are supported only for 0x19 service.

Requirements:

Dcm028

- ▶ No support of RTE Sender-Receiver interfaces for services `InputOutputControlByIdentifier` (0x2E) and `RequestCurrentPowertrainDiagnosticData` (0x01) (reference to product description: ASCPD-110)

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3
- ▶ R4.3.0

Description:

- ▶ RTE Sender-Receiver interface for service `InputOutputControlByIdentifier` (0x2F) is not supported.
- ▶ RTE Sender-Receiver interface for service `RequestCurrentPowertrainDiagnosticData` (\$01) is not supported.

Requirements:

Dcm639, Dcm638, Dcm640, Dcm718, SWS_Dcm_01121, SWS_Dcm_01122, SWS_Dcm_01123, SWS_Dcm_01124, SWS_Dcm_01125, SWS_Dcm_00687

- ▶ DID data access using ECU Signal interfaces (reference to product description: ASCPD-69)

Affected AUTOSAR releases:



- ▶ R4.0 Rev 3

Description:

Interaction with IoHwAb module for data services is not supported.

Rationale:

A workaround is to use the function callouts generated in case the configured interface is `USE_DATA_SYNCH_FNC`. In this case, the differences is:

- ▶ For service `InputOutputControlByIdentifier` (0x2F), instead of a call to API: `void IoHwAb_Dcm_<EcuSignalName>(uint8 action, <EcuSignalDataType> signal)`, with 'action' set to the operation desired, and in case the 'action' is `shortTermAdjustment`, the 'signal' set to the desired value, a call to one of the User-defined callouts (depending on the desired operation):
 - ▶ `Std_ReturnType User_ShortTermAdjustment(uint8* ControlOptionRecord, Dcm_NegativeResponseCodeType* ErrorCode)`
 - ▶ `Std_ReturnType User_FreezeCurrentState(Dcm_NegativeResponseCodeType* ErrorCode)`
 - ▶ `Std_ReturnType User_ResetToDefault(Dcm_NegativeResponseCodeType* ErrorCode)`
 - ▶ `Std_ReturnType User_ReturnControlToEcu(Dcm_NegativeResponseCodeType* ErrorCode)`
- ▶ For service `ReadDataByIdentifier` (0x22), instead of a call to API: `void IoHwAb_Dcm_<EcuSignalName>(<EcuSignalDataType>* signal)` with the current signal value to be stored at pointer 'signal', a call to the user-defined callout: `Std_ReturnType User_ReadData(uint8* Data)`

Since all these user-configurable function callouts either share the same signature with the APIs in the IoHwAb module, or the API from the IoHwAb module is refined into four callouts, depending on the 'action' desired, these mechanisms are equivalent and may be used as a valid workaround. The specification (of both Dcm and IoHwAb modules) is also inconsistent in regard to this behavior, as it is nowhere specified where the type `EcuSignalDataType` is to be obtained from.

Requirements:

`Dcm578, Dcm580, Dcm626, Dcm627, Dcm628`

- ▶ Values of `Dcm_StatusType` errors changed for consistency

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:



The error values defined by the type `Dcm_StatusType` deviate from the ones found in the AUTOSAR R4.0 Rev 3 SWS in Section 8.2.1: Value of error `DCM_E_NOT_OK` is changed to 0x01 from 0x0B. Value of error `DCM_E_COMPARE_KEY_FAILED` is changed to 0x0B from 0x01.

Rationale:

To keep the error values consistent between `Dcm_StatusType` and the Rte interface.

Requirements:

Section 8.2.1

- ▶ Default session needs to be configured always

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

The `Default Session` needs to be always configured with a value of 1.

Rationale:

Every ECU is initialized to the `Default Session` after power-on. Also the default values of `P2` time and `P2*` time for the ECU are specified along with `Default Session` configuration.

Requirements:

`Dcm769_Conf`, `Dcm767_Conf`, `Dcm765_Conf`

- ▶ Renamed `DCM_DISABLE_RX_TX_NORMAL` to `DCM_DISABLE_RX_TX_NORM`

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Enum `DCM_DISABLE_RX_TX_NORMAL` of type `Dcm_CommunicationModeType` was renamed and exported as `DCM_DISABLE_RX_TX_NORM`.

Rationale:

To keep uniformity between the names of enums of type `Dcm_CommunicationModeType`. See also AUTOSAR Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=50341.

Requirements:

Section 8.2.6



- ▶ Dcm does not handle read access for `dataIdentifier 0xF186` completely internally.

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Dcm085 specifies that the read access of the `dataIdentifier 0xF186 (ActiveDiagnosticSessionDataIdentifier)` shall be handled internally within DSP. This Dcm implementation contains a recommended configuration which enables the DID 0xF186 as a normal DID and passes it to an internal function providing this information.

Rationale:

Providing this information using regular DID reading mechanisms is the most transparent and flexible solution for the user. Furthermore, there might be use cases in which `dataIdentifier 0xF186` needs to be handled by the application using `Dcm_GetSesCtrlType()`.

In order to configure DID 0xF186 for internal read access follow the steps below:

- ▶ Add DID 0xf186
- ▶ Add a `DcmDspData` with the following configuration:
 - ▶ Enable `DcmDspDataReadFnc` with dedicated function name `Dcm_Internalf186Readout`
 - ▶ Set `DcmDspDataUsePort` to `USE_DATA_SYNCH_FNC`
 - ▶ Configure for `DcmDspDataInfoRef` a "fix data length signal"
 - ▶ Configure `DcmDspDataSize` to "8 bit" (one byte) and `DcmDspDataType` to `UINT8`
- ▶ Configure a `DcmDspDataInfo` with `DcmDspDataFixedLength` dedicated for configured `DcmDspData`
- ▶ Configure a `DcmDspDidInfo` with `DcmDspDidRead` enabled
- ▶ Configure `DcmDspDidInfoRef` from DID 0xf186 to the above configured `DcmDspDidInfo`
- ▶ Configure only one `DcmDspDidSignal` with `DcmDspDidDataRef` to the above configured `DcmDspData`

Requirements:

Dcm085

- ▶ Mandatory configuration of parameter `DcmDspMaxDidToDelete`

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:



Multiplicity of parameter `DcmDspMaxDidToRead` is changed to 1, so this parameter always has to be configured if the service `ReadDataByIdentifier` (0x22) is used. Also, the range of values for this parameter has been changed to 1 .. 65535.

Rationale:

For implementation of Dcm module, the maximum number of Dids that can be received in a single request of service 0x22 has to be available. See http://www.autosar.org/bugzilla/show_bug.cgi?id=52991.

Requirements:

`Dcm638_Conf`

- ▶ Mandatory configuration of parameter `DcmDspMaxDidToRead`

Affected AUTOSAR releases:

- ▶ R4.2 Rev 1

Description:

Multiplicity of parameter `DcmDspMaxDidToRead` is changed to 1, so this parameter always has to be configured if the service `ReadDataByIdentifier` (0x22) is used.

Rationale:

For implementation of Dcm module, the maximum number of Dids that can be received in a single request of service 0x22 has to be available. See http://www.autosar.org/bugzilla/show_bug.cgi?id=52991.

Requirements:

`ECUC_Dcm_00638`

- ▶ Return value of `E_NOT_OK` for `Indication()` operation

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

If the Manufacturer/Supplier `Indication()` operation returns `E_NOT_OK`, with ErrorCode `DCM_E_OK`, the Dcm module shall send a generalReject (0x10) negative response code. See configuration parameters `DcmDslServiceRequestManufacturerNotification` and `DcmDslServiceRequestSupplierNotification`.

Rationale:



Requirements Dcm463 and Dcm518 specify that if `E_NOT_OK` is returned from `Indication()`, the Dcm shall send a negative response with NRC value equal to `ErrorCode` parameter value. But if an error code is not set from the function, the Dcm shall send the `generalReject` (0x10) NRC. For all interfaces having `ErrorCode` as out parameter, if interfaces return positive return values, the Dcm does not verify the `ErrorCode`.

Requirements:

Dcm463, Dcm518

- ▶ Return value of `DCM_E_REQUEST_NOT_ACCEPTED` for `Indication()` operation

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

If the Manufacturer/Supplier `Indication()` operation returns `DCM_E_REQUEST_NOT_ACCEPTED`, and at least one `RequestCorrectlyReceivedResponsePending` (0x78) NRC has been sent for the UDS service request, the Dcm module shall send a `ErrorCode ConditionsNotCorrect` (0x22) negative response code.

Rationale:

Requirements Dcm462 and Dcm517 specify that if `E_REQUEST_NOT_ACCEPTED` is returned from `Indication()`, the Dcm shall give no response. But if at least one `RequestCorrectlyReceivedResponsePending` (0x78) NRC has been sent for the service request, according with ISO 14229-1, it needs to be followed by a final response.

Requirements:

Dcm462, Dcm517

- ▶ Subfunction `reportDTCSnapShotRecordByRecordNumber` (0x05) of service `ReadDTCInformation` (0x19) (reference to product description: ASCPD-81)

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

`reportDTCSnapShotRecordByRecordNumber` (0x05) subfunction of UDS service `ReadDTCInformation` (0x19) is not supported.

Requirements:



Dcm632, Dcm574, Dcm388, Dcm389

- ▶ **Subfunction reportMirrorMemoryDTCByStatusMask (0x0F) of service ReadDTCInformation (0x19)** (reference to product description: ASCPD-81)

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

reportMirrorMemoryDTCByStatusMask (0x0F) subfunction of UDS service ReadDTCInformation (0x19) is not supported.

Requirements:

Dcm377, Dcm008, Dcm378

- ▶ **Subfunction reportMirrorMemoryDTCExtendedDataRecordByDTCNumber (0x10) of service ReadDTCInformation (0x19)** (reference to product description: ASCPD-81)

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

reportMirrorMemoryDTCExtendedDataRecordByDTCNumber (0x10) subfunction of UDS service ReadDTCInformation (0x19) is not supported.

Requirements:

Dcm297, Dcm295, Dcm296, Dcm478, Dcm386, Dcm382

- ▶ **Subfunction reportNumberOfMirrorMemoryDTCByStatusMask (0x11) of service ReadDTCInformation (0x19)** (reference to product description: ASCPD-81)

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

reportNumberOfMirrorMemoryDTCByStatusMask (0x11) subfunction of UDS service ReadDTCInformation (0x19) is not supported.

Requirements:

Dcm376, Dcm293



- ▶ Subfunction `reportNumberOfEmissionsRelatedOBDDTCByStatusMask` (0x12) of service `ReadDTCInformation` (0x19) (reference to product description: ASCPD-81)

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

`reportNumberOfEmissionsRelatedOBDDTCByStatusMask` (0x12) subfunction of UDS service `ReadDTCInformation` (0x19) is not supported.

Requirements:

Dcm376, Dcm293

- ▶ DcmDsp Container Multiplicity

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Multiplicity of `DcmDsp` is changed from 0..1 to 1..1. `DcmDsp` must be configured explicitly.

Rationale:

The explicit configuration of `DcmDsp` allows simpler code generation.

Requirements:

Dcm712_Conf

- ▶ Behavior related to `ModeDeclarationgroupPrototype` for UDS service `LinkControl` (0x87) (reference to product description: ASCPD-70)

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Features related to `ModeDeclarationgroupPrototype` for UDS service `LinkControl` (0x87) are not supported.

Requirements:

Dcm533



- ▶ Service `LinkControl` (0x87) functionality implemented as callout (reference to product description: ASCPD-81)

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Dcm invokes a callout after receiving service request for Link control service. The integrator will provide the implementation part of the callout.

Rationale:

Functionality supported as a callout.

Requirements:

Dcm744, Dcm531

- ▶ BswM interface is not compliant to ASR 4.0.3

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

BswM interfaces used in Dcm module are conformant to ASR release R4.0.2

Requirements:

Section 8.7.2

- ▶ Structure of the `Dcm_ProgConditionsType` data type is changed

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

`Dcm_ProgConditionsType` has been changed to:

Name: `Dcm_ProgConditionsType`

Type: structure

Element:

`uint16 TesterSourceAddr`: Tester source address configured per protocol

`uint8 ProtocolId`: Id of the protocol on which the request has been received



uint8 Sid: Service identifier of the received request
 uint8 SubFncId: Identifier of the received subfunction
 boolean ReprogrammingRequest: Set/Clear the ReprogrammingRequest flag on ECU reset
 boolean ApplUpdated: Set/Clear the ApplUpdated flag on ECU reset
 boolean ResponseRequired: Set/Clear the ResponseRequired flag on ECU reset

Description:

Used in `Dcm_SetProgConditions()` to allow the integrator to store relevant information prior to jumping to bootloader.

Rationale:

The `DcmDs1ProtocolRxTesterSourceAddr` is a 16-bit integer, therefore an 8-bit integer is insufficient to hold this information. This is the solution for the ongoing Bugzilla #55594 discussion.

Requirements:

Section 8.2.14

- ▶ Interface DataServices_DIDRange_{Range}

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Interface `DataServices_DIDRange_{Range}` is supported according to AUTOSAR 4.2.-1: ClientServerInterface Operation `IsDidAvailable` receives a parameter `OpStatus` and can be asynchronous. DataServices callout function `Xxx_IsDidAvailable` receives a parameter `OpStatus` and can be asynchronous. DataServices callout functions `Xxx_ReadDidData` and `Xxx_WriteDidData` can be asynchronous. DID Range checks are not supported in this implementation.

Rationale:

The specification has been reworked / refined.

Requirements:

Section 8.7.3.3, Dcm769, Dcm803, Dcm804, Dcm805

- ▶ Support of Dids configured via container `DcmDspDidRange`

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:



Dids configured via container `DcmDspDidRange` are not supported in all diagnostic services.

The following diagnostic services return NRC 0x31 REQUEST OUT OF RANGE if a requested Did is not configured via `DcmDspDid`:

- ▶ 0x2F IOControl
- ▶ 0x2A ReadDataByPeriodicIdentifier
- ▶ 0x2C DynamicallyDefineDataIdentifier

Rationale:

These internal service handlers do not support the asynchronous execution of the availability check, yet.

Requirements:

Section 8.7.3.3, Dcm769, Dcm803, Dcm804, Dcm805

- ▶ Multiplicity of `DcmDspMemoryIdInfo`

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Multiplicity of the container `DcmDspMemoryIdInfo` is changed from 1..* to 0..*, as the parameter and all the sub-containers are optional.

Rationale:

To optimize the configuration, multiplicity is changed. Refer to AUTOSAR Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=56265.

Requirements:

Section 10.2.39, Dcm911_Conf

- ▶ Configuration parameter `DcmDslProtocolRxComMChannelRef` is implemented as CHOICE-REFERENCE

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:



The configuration parameter `DcmDslProtocolRxComMChannelRef` is implemented as CHOICE-REFERENCE instead of SYMBOLIC-NAME-REFERENCE. This is required by the parameter's entry in AUTOSAR_MOD_ECUConfigurationParameters.arxml

Rationale:

This parameter was changed to a CHOICE-REFERENCE type in order to allow the ACG6.4.0 Dcm module to be used with the ACG6.3.0 ComM module. This allows the user to switch between the ACG6.3.0 or the ACG6.4.0 versions of the paths for the `ComMChannel` parameter in the ComM module. As one of these paths is always valid, the Dcm is compatible with both versions of the ComM module.

Requirements:

Dcm906_Conf

- ▶ Configuration parameters `DcmDspAllComMChannelRef` and `DcmDspSpecificComMChannelRef` and `DcmDspComControlSubNodeComMChannelRef` are implemented as CHOICE-REFERENCE.

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

The configuration parameters `DcmDspAllComMChannelRef` and `DcmDspSpecificComMChannelRef` and `DcmDspComControlSubNodeComMChannelRef` are implemented as CHOICE-REFERENCE instead of SYMBOLIC-NAME-REFERENCE. This is required by the parameter's entry in AUTOSAR_MOD_ECUConfigurationParameters.arxml.

Rationale:

This parameter was changed to a CHOICE-REFERENCE type in order to allow the ACG6.4.0 Dcm module to be used with the ACG6.3.0 ComM module. This allows the user to switch between the ACG6.3.0 or the ACG6.4.0 versions of the paths for the `ComMChannel` parameter in the ComM module. As one of these paths is always valid, the Dcm is compatible with both versions of the ComM module.

Requirements:

Dcm902_Conf, Dcm904_Conf, ECUC_Dcm_01030

- ▶ Functionality of services `ControlDTCSetting` (0x85) is executed before the response is sent

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:



The AUTOSAR standard R4.0 Rev 3 defines that the functionality related to the services `ControlDTCSetting` (0x85) is executed if the send confirmation function for service responses is invoked. The current Dcm has the following implemented:

- ▶ The Dcm module does not invoke `Dem_EnableDTCSetting()` if the DSD calls the send confirmation function for the response to a request for the service `ControlDTCSetting` with `DTCSettingType = on`.
- ▶ The Dcm module does not invoke `Dem_DisableDTCSetting()` if the DSD calls the send confirmation function for the response to a request for the service `ControlDTCSetting` with `DTCSettingType = off`.
- ▶ The Dcm module does not invoke a switch of the `ModeDeclarationGroupPrototype DcmControlDTCSetting` to `ModeDeclaration ENABLEDTCSETTING` if the DSD calls the send confirmation function for the response to a request for the service `ControlDTCSetting` (0x85) with `DTCSettingType = on`.
- ▶ The Dcm module does not invoke a switch of the `ModeDeclarationGroupPrototype DcmControlDTCSetting` to `ModeDeclaration DISABLEDTCSETTING` if the DSD calls the send confirmation function for the response to a request for the service `ControlDTCSetting` (0x85) with `DTCSettingType = off`.

Instead, this functionality is executed before the responses to the requests are transmitted.

Rationale:

In this Dcm implementation, the above mentioned functionality is caused by a bug.

Requirements:

`Dcm304, Dcm406, Dcm783, Dcm784`

- ▶ Configuration parameter `CONTEXT` within `DcmBswModeRef` and `DcmSwcModeRef` is disabled

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

The configuration parameter `CONTEXT` within `DcmBswModeRef` and `DcmSwcModeRef` is omitted and not used to get the `ModeDeclarationGroupPrototype`.

Rationale:

Configuration parameters `DcmBswModeRef` and `DcmSwcModeRef` refer to the `ModeDeclaration` path within their `TARGET` and `ModeDeclarationGroupPrototype` path within `CONTEXT`. `ModeDeclarationGroupPrototype` can be obtained from the path provided by `TARGET` itself so that `CONTEXT` is not required.



Requirements:

Dcm931_Conf, Dcm930_Conf

- ▶ **Dcm_Init() parameter ConfigPtr updated with const qualifier**

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

A const qualifier is missing for the input parameter `ConfigPtr` of `Dcm_Init()`. This parameter is used during post-build time.

Rationale:

The missing const qualifier for the `ConfigPtr` is a bug in the AUTOSAR R4.0 Rev 3 Dcm SWS that has been corrected in the AUTOSAR R4.1 Rev 1 Dcm SWS. Refer to AUTOSAR Bugzilla http://www-autosar.org/bugzilla/show_bug.cgi?id=52382.

Requirements:

Dcm037

- ▶ **Dcm_Init() need not be called before Dcm_GetVersionInfo()**

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

If the development error detection is enabled, for a request of `Dcm_GetVersionInfo`, it is not necessary that the Dcm module is already initialized.

Rationale:

Version check of the module could be performed prior to a module initialization. Refer to BswM_C_207.

Requirements:

Dcm043

- ▶ **Parameter `DcmTxPduId` in `Dcm_CopyTxData()` is not the same as in `PduR_DcmTransmit()`**

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3



Description:

Dcm_CopyTxData() uses parameter DcmTxPduId which is the Tx Pduld configured within the Dcm DcmDslTxConfirmationPduId.

Rationale:

The PduR shall call Dcm_CopyTxData() through a handle ID defined by the Dcm, according to the handle ID concept in AUTOSAR_TPS_ECUConfiguration.pdf. The PduR shall not call Dcm_CopyTxData() through a handle ID defined by the PduR. Refer to AUTOSAR Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=57991.

Requirements:

Dcm350

- ▶ Internal service handlers do not call Dcm_ExternalProcessingDone

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Internal service handlers do not call Dcm_ExternalProcessingDone(). It is called only from external service handlers.

Rationale:

This does not effect the functionality of service handlers.

Requirements:

Dcm269, Dcm733, Dcm225

- ▶ No AUTOSAR debugging support

Description:

AUTOSAR debugging support is not available.

Requirements:

Dcm484, Dcm485, Dcm486, Dcm487, Dcm506, Dcm507, Dcm508, Dcm509

- ▶ Link to service table not done during run-time at protocol initialization

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

**Description:**

For each configured protocol, a link to the corresponding service table shall be done statically in the code generation step.

Requirements:

Dcm035

- ▶ Endianness conversion supported for all interface types

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Endianness conversion is performed on data that belongs to `DcmdspData` elements with `DcmDsp-DataUsePort` configured to `USE_DATA_SYNCH_FNC`, `USE_DATA_ASYNCH_FNC`, `USE_DATA_SYNCH_CLIENT_SERVER`, `USE_DATA_ASYNCH_CLIENT_SERVER`. Also the conversion is not only performed on data that belongs to `DcmdspData` elements with `DcmDspDataUsePort` configured to `USE_SENDER_RECEIVER` or `USE_ECU_SIGNAL`. Note: For non-support of Sender-Receiver and IoHwAb interfaces themselves, refer to the related deviation entry 'No support of RTE Sender-Receiver interface for services `ReadDataByIdentifier (0x22)` and `WriteDataByIdentifier (0x2E)`' and 'DID data access using ECU Signal interfaces'.

Rationale:

Endian conversion is valid for all interface types.

Requirements:

Dcm638, Dcm639, Dcm640, Dcm718, Dcm716

- ▶ Parameter `DcmRespondAllRequest`

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Parameter `DcmRespondAllRequest` is not supported in this implementation.

Requirements:

Dcm084



- ▶ Return value of `E_NOT_OK` for `Confirmation()` operation that belongs to the `ServiceRequestNotification` interface

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

The `Confirmation()` operation is called after sending the NRC or a positive response. Thus it does not set any `ErrorCode`.

Rationale:

The `ErrorCode` parameter is not present in the `Xxx_Confirmation()` operation.

Requirements:

Dcm678

- ▶ Security level, session type and active protocol interfaces return `E_NOT_OK`

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

For the following interfaces the `E_NOT_OK` error code can be returned in case the DET is enabled and the interfaces are called before a Dcm initialization or are called with invalid parameters:

- ▶ `Dcm_GetSecurityLevel`
- ▶ `Dcm_GetSesCtrlType`
- ▶ `Dcm_GetActiveProtocol`

In case the Dcm is started after the application, the Det error-checking needs to be enabled to ensure run-time integrity. This applies for the production code.

Rationale:

Requires clarification from Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=54135

Requirements:

Dcm338, Dcm339, Dcm340

- ▶ No jump to bootloader if transmission of NRC 0x78 fails

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3



Description:

If the jump to bootloader is requested with the configuration parameter `DcmSendRespPendOnTransToBoot` set to 'true' but the NRC 0x78 (Response Pending) response is not sent successfully, the Dcm shall cancel the current request. A jump to the bootloader shall not be performed.

Rationale:

The jump should only be performed when the transmission is successful and not only confirmed. Refer to AUTOSAR Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=57856.

Requirements:

Dcm535

- ▶ RxBufferSizePtr of `Dcm_StartOfReception` and `Dcm_CopyRxData()` if `SduLength` from `PduInfoPtr` equals 0

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

If `Dcm_StartOfReception` is invoked with `TpSduLength` equal to 0, the value `BUFREQ_OK` shall be returned and `RxBufferSizePtr` shall be set to the configured size of the allocated Rx buffer. If `Dcm_CopyRxData` is invoked with `SduLength` from `PduInfoPtr` equal to 0, the value `BUFREQ_OK` shall be returned and `RxBufferSizePtr` shall be filled with the remaining size of the Rx buffer.

Rationale:

Dcm642 does not mention that the size to be returned should be the remaining size of the Rx buffer and not the allocated size itself in case of `Dcm_CopyRxData` if `SduLength` equals 0. Refer to AUTOSAR Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=56890.

Requirements:

Dcm642

- ▶ Additional configuration parameters added to the container `DcmDspSecurityRow`

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

`DcmDspSecurityUsePort`, `DcmDspSecurityGetSeedFnc`, and `DcmDspSecurityCompareKeyFnc` are added to the `DcmDspSecurityRow` container. `DcmDspSecurityUsePort` defines the inter-



face which shall be used to access the data, either USE_SECURITY_ASYNC_FNC, USE_SECURITY_SYNCH_FNC, USE_SECURITY_SYNCH_CLIENT_SERVER, or USE_SECURITY_ASYNC_CLIENT_SERVER for the service SecurityAccess (0x27). Based on the value of DcmDspSecurityUsePort (USE_SECURITY_ASYNC_FNC, USE_SECURITY_SYNCH_FNC), additional parameters DcmDspSecurityGetSeedFnc and DcmDspSecurityCompareKeyFnc are introduced.

Rationale:

Refer to AUTOSAR Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=53066.

Requirements:

Dcm759_Conf

- ▶ Additional configuration parameters added to the container DcmDspMemory

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Container DcmDspAddressAndLengthFormatIdentifier and list DcmDspSupportedAddressAndLengthFormatIdentifier are added in the DcmDspMemory container.

Rationale:

Refer to AUTOSAR Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=53661.

Requirements:

Dcm784_Conf

- ▶ Function interface not specified

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

The operations mentioned in the requirements Dcm793, Dcm794, Dcm796, Dcm797, Dcm799, Dcm800, Dcm801 and Dcm802 should use synchronous or asynchronous function interface operations instead of a client server interface. The signatures of operations for a client server interface is already mentioned in Dcm686.

Rationale:



This implements the solution for the issue reported in http://www.autosar.org/bugzilla/show_bug.cgi?id=61514. The issue is introduced in http://www.autosar.org/bugzilla/show_bug.cgi?id=54767, as in the implementation Callout interfaces and ClientServer interfaces share the same signature.

Requirements:

Dcm793, Dcm794, Dcm796, Dcm797, Dcm798, Dcm799, Dcm800, Dcm801, Dcm802

- ▶ Dcm_GetProgConditions() called in Dcm_Init()

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

In order for the Dcm to know if its initialization is the consequence of a jump from the bootloader, it shall call Dcm_GetProgConditions() in Dcm_Init(). Dcm_GetProgConditions() shall not be called during the first call to Dcm_MainFunction(), as mentioned in the requirement Dcm536.

Rationale:

Refer to AUTOSAR Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=61670

Requirements:

Dcm536

- ▶ Support for RoutineControl services with fixed signals if DcmDspRoutineFixedLength is set to false

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

In contrast to the definitions for the Start, Stop, and RequestResults interfaces mentioned in the requirement Dcm690 (AUTOSAR SWS R4.0 Rev 3) it is not mandatory for their signatures to contain VARIABLE_LENGTH input and output signals if DcmDspRoutineFixedLength is set to false.

Requirements:

Dcm690

- ▶ Signature of expected interfaces for the RequestRoutineResults operation of service RoutineControl (0x31)

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3



Description:

In addition to the definition for the `RequestResults` interface mentioned in the requirement Dcm690 (AUTOSAR SWS R4.0 Rev 3), the `RequestRoutineResults` operation can also accept input signals. Therefore, the signature for this interface may additionally have `IN <datatype> dataIn` or `IN uint8 dataInN[(<DcmDspRoutineSignalLength of DcmDspStartRoutineInSignal> +7) /8]` input parameters similar to the `Start` and `Stop` operations.

Requirements:

Dcm690

- ▶ If the configuration parameter `DcmDataTransferServicesASRVersion` is set to the value `DATA_TRANSFER_SERVICES_AUTOSAR_403`, the NRC `GeneralReject` (0x10) is triggered if `Dcm_ReadMemory()` returns `DCM_READ_FAILED`.

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

If the configuration parameter `DcmDataTransferServicesASRVersion` is set to the value `DATA_TRANSFER_SERVICES_AUTOSAR_403` and if the call to `Dcm_ReadMemory` returns `DCM_READ_FAILED`, the DCM module shall trigger a negative response with NRC `GeneralReject` (0x10), instead of `GeneralProgrammingFailure` (0x72) according with RFC #57196.

Requirements:

Dcm644

- ▶ The `Dem_SetDTCFilter()` API is not called from the context of the `Dcm_MainFunction()` when the service `ReadDTCInformation` (0x19) is requested with the subfunction `ReportDTCFaultDetectionCounter` (0x14)

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

In case asynchronous service processing is enabled for service `ReadDTCInformation` (0x19) (parameter `DcmAsyncServiceExecution` is set to 'true'), and the subfunction `ReportDTCFaultDetectionCounter` (0x14) is enabled and configured to use the EB-supplied implementation, then the `Dem_SetDTCFilter()` API is not called from the context of the `Dcm_MainFunction()`.

Requirements:



Dcm.ReportFaultDetectionCounter.Async

- ▶ Size calculation of ImplementationDataType DataArrayTypeUint8_{Data}

Affected AUTOSAR releases:

- ▶ R4.2 Rev 1

Description:

The size of the Array type is calculated wrong. The right size should be:

- ▶ $\{\text{ecuc}(\text{Dcm}/\text{DcmConfigSet}/\text{DcmDsp}/\text{DcmDspData}/\text{DcmDspDataSize})\}+7)/8$ respective
- ▶ $\{\text{ecuc}(\text{Dcm}/\text{DcmConfigSet}/\text{DcmDsp}/\text{DcmDspPid}/\text{DcmDspPidData}/\text{DcmDspPidDataSize})\}+7)/8$

Rationale:

The type DataArrayTypeUint8_{Data} shall store a count of bits as specified in either of the following:

- ▶ 'count of bits' = $\{\text{ecuc}(\text{Dcm}/\text{DcmConfigSet}/\text{DcmDsp}/\text{DcmDspData}/\text{DcmDspDataSize})\}$
- ▶ 'count of bits' = $\{\text{ecuc}(\text{Dcm}/\text{DcmConfigSet}/\text{DcmDsp}/\text{DcmDspPid}/\text{DcmDspPidData}/\text{DcmDspPidDataSize})\}$

For this a count of bytes is required. It is calculated as follows: 'count of bytes' = ('count of bits'+7)+8

Requirements:

SWS_Dcm_01121

- ▶ The parameter `DcmDspTestResultObdmidTidRef` is not supported

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Instead a vendor specific container `DcmDspTestResultObdmidTidRefs` was introduced. See AutoCore_Generic_Diagnostic_Stack_documentation.pdf chapter `DcmDspTestResultObdmidTidRefs`.

Requirements:

Dcm685_Conf

- ▶ Signature of expected interfaces for the `RequestRoutineResults` operation of service `RoutineControl` (0x31)

Affected AUTOSAR releases:

- ▶ R4.2 Rev 1



- ▶ CP Release 4.3.0

Description:

In addition to the definition for the `RequestResults` interface mentioned in the requirement `SWS_Dcm_00690` (AUTOSAR SWS R4.2 Rev 1 and AUTOSAR SWS CP Release 4.3.0), the `RequestRoutineResults` operation can also accept input signals. Therefore, the signature for this interface may additionally have `IN <datatype> dataIn` or `IN uint8 dataInN[(<DcmDspRoutineSignalLength of DcmDspStartRoutineInSignal> + 7) / 8]` input parameters similar to the `Start` and `Stop` operations.

Requirements:

`SWS_Dcm_00690`

- ▶ Storing the dynamic length of periodicDIDs signals at the time of configuration through UDS service `ReadDataByPeriodicIdentifier` (0x2A)

Affected AUTOSAR releases:

- ▶ R4.2 Rev 1

Description:

Requirement `SWS_Dcm_01099` (AUTOSAR SWS R4.2 Rev 1) states that upon configuring a periodicDID, with a configured dynamic length the Dcm module shall invoke the `ReadDataLength` operation (or the respective C-Function) to retrieve the length of the periodicDID and then use and consider that length valid for each `ReadData` operation till the periodicDID is removed from the scheduler or updated via a new request. That length shall further be used to check against the UUDT size.

The current implementation does not support this. The length of the Data is read each time the data is retrieved and not just once at configuration time. The length is also not used to check against UUDT size.

Requirements:

`SWS_Dcm_01099`

- ▶ Value of IN-parameters of interfaces when being called with an `OpStatus` equal to `DCM_PENDING`, `DCM_CANCEL` or `DCM_FORCE_RCRRP_OK`

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

The Dcm does not clear all the input parameters of the following interfaces:

- ▶ `<Module>_<DiagnosticService>()`



- ▶ <Module> _<DiagnosticService> _<SubService>()
 - ▶ Dcm_WriteMemory()
- on calls with a DCM_PENDING, DCM_CANCEL or DCM_FORCE_RCRRP_OK OpStatus.

Rationale:

For <Module> _<DiagnosticService>() and <Module> _<DiagnosticService> _<SubService>(), this would not allow the pMsgContext input data (containing the response buffers) to be used by subsequent calls of the service or subfunction handler (and would require it to be memorized by the service handler in the first call).

For Dcm_WriteMemory(), this allows the Application to read data from the *MemoryData input buffer, thus making it not necessary to copy all this data into an own buffer. This improves memory usage.

The fact that these parameters are not cleared should not actually cause any problems. The Application should assume that they are cleared as per the SWS if the SWS-compliant behavior is required, and therefore ignore their values on subsequent calls with OpStatus values of DCM_PENDING, DCM_CANCEL or DCM_FORCE_RCRRP_OK.

Requirements:

Dcm763, Dcm764, Dcm540

- ▶ Updated name of AUTOSAR parameter

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

In the AUTOSAR release R4.0 Rev 3, the configuration parameter DcmDslPeriodicTransmissionConRef has a typo. This is changed in AUTOSAR release R4.1 Rev 1 based on Rfc 53221 into DcmDslPeriodicTransmissionConRef. This typo fix is included in this implementation.

Requirements:

Dcm707_Conf,

- ▶ Dcm_DemTriggerOnDTCStatus interfaces return E_NOT_OK

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:



For the Dcm_DemTriggerOnDTCStatus interface the `E_NOT_OK` error code can be returned in case the DET is enabled and the interfaces are called before a Dcm initialization. In case the Dcm is started after the application, the Det error-checking needs to be enabled to ensure run-time integrity. This applies for the production code.

Rationale:

All DCM interfaces return `E_NOT_OK` if DCM module is not initialized and Det error-checking is enabled.

Requirements:

SWS_Dcm_00614

- ▶ Dcm_ResetToDefaultSession interfaces return `E_NOT_OK`

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

For the Dcm_ResetToDefaultSession interface the `E_NOT_OK` error code can be returned in case the DET is enabled and the interfaces are called before a Dcm initialization. In case the Dcm is started after the application, the Det error-checking needs to be enabled to ensure run-time integrity. This applies for the production code.

Rationale:

All DCM interfaces return `E_NOT_OK` if DCM module is not initialized and Det error-checking is enabled.

Requirements:

Dcm520

- ▶ Dcm_CopyTxData interfaces return `BUFREQ_E_BUSY`

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

For the Dcm_CopyTxData interface the `BUFREQ_E_BUSY` error code can be returned in case the request could not be fulfilled, because the required amount of Tx data is not available.

Requirements:

Dcm092

- ▶ Dcm_DemTriggerOnDTCStatus always returns `E_NOT_OK`



Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

For the Dcm_DemTriggerOnDTCStatus interface the `E_NOT_OK` error code is always returned.

Rationale:

The `onDTCStatusChange` subfunction of the `ResponseOnEvent` service is not implemented yet.

Requirements:

SWS_Dcm_00614

- ▶ The UDS service `ResponseOnEvent` (0x86) is implemented according to AUTOSAR 4.2.1 specification

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

The UDS service `ResponseOnEvent` (0x86) is implemented according to AUTOSAR 4.2.1 specification. The ROE specification from the AUTOSAR 4.0.3 specification was significantly reworked in the new release of the document.

Requirements:

Dcm604, Dcm605, Dcm710, Dcm606, Dcm607, Dcm608, Dcm609, Dcm610, Dcm611, Dcm612, Dcm613, Dcm619, Dcm859_Conf, Dcm780_Conf, Dcm782_Conf, Dcm781_Conf, Dcm933_Conf, Dcm603, Dcm582, Dcm524, Dcm523, Dcm137, Dcm597, Dcm618, Dcm712, Dcm713, Dcm714, Dcm731, Dcm748, Dcm765, Dcm772, Dcm730, Dcm521, Dcm679, Dcm852_Conf, Dcm853_Conf, Dcm854_Conf, Dcm855_Conf, Dcm857_Conf, Dcm934_Conf, Dcm162, Dcm695

- ▶ The UDS service `ResponseOnEvent` (0x86) is not fully implemented.

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

The UDS service `ResponseOnEvent` (0x86) according to AUTOSAR 4.2.1 specification is implemented in two phases. The requirements included in this deviation will be implemented in phase 2.

Requirements:



SWS_Dcm_00873.OnChangeOfDID, SWS_Dcm_00893.OnChangeOfDID, SWS_Dcm_00907, SWS_Dcm_00912, SWS_Dcm_00913, SWS_Dcm_00918, SWS_Dcm_00892.onChangeOfDataIdentifierNonPersistent, SWS_Dcm_00892.onChangeOfDataIdentifierPersistent

- ▶ Number of fixed length signals before a dynamic length signal

Affected ISO 14229-1 releases:

- ▶ ISO14229-1:2013

Description:

Dcm620 states that the data of a DID can have dynamic datalength only if this DID contains only one data. See configuration parameter `DcmDspDataFixedLength`. The latest AUTOSAR specification relaxes that requirement to constr_6039, which says: Signals with variable datalength: Only the last signal (`DcmDspDidSignal`) of a DID can have variable datalength (`DcmDspDataType` is set to `UINT8_DYN`). The implementation supports more than one signal of fixed length with the last signal allowed to be of dynamic length.

Requirements:

Dcm620

- ▶ The `DCM_FORCE_RCRRP_OK` value for the parameter `OpStatus` for the fallowing interfaces is not supported due to inconsistent requirements:

- ▶ `Dcm_ProcessRequestDownload`
- ▶ `Dcm_ProcessRequestUpload`
- ▶ `Dcm_ProcessRequestTransferExit`

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3
- ▶ CP Release 4.3.0

Rationale:

AUTOSAR 4.0.3 specifies `DCM_FORCE_RCRRP_OK` as a valid `OpStatus` value for the interfaces mentioned above. But AUTOSAR 4.3.0 specifies it as valid only for `Dcm_ProcessRequestUpload`. Moreover, `DCM_E_FORCE_RCRRP` is not listed as a required supported return value for `Dcm_ProcessRequestDownload`, `Dcm_ProcessRequestUpload`, and `Dcm_ProcessRequestTransferExit` in neither AUTOSAR 4.0.3 or AUTOSAR 4.3.0. Without this specific error code there would not be any subsequent call of these interfaces with `OpStatus DCM_FORCE_RCRRP_OK`.

Dcm754, Dcm756, Dcm755

- ▶ Multiplicity of `DcmDslProtocolTx`



Affected AUTOSAR releases:

- ▶ CP Release 4.3.0

Description:

Multiplicity of the container `DcmDslProtocolTx` is changed from 1 to 0..1, as the container is redefined to be optional.

Rationale:

To enable response suppression in case a response to a service request is not allowed or possible. See AUTOSAR Bugzilla https://www.autosar.org/bugzilla/show_bug.cgi?id=64765.

Requirements:

`ECUC_Dcm_00711`

- ▶ Reset of a security level's specific `AttemptCounter`

Affected ISO 14229-1 releases:

- ▶ ISO14229-1:2013

Description:

`SWS_Dcm_01357` allows a security level's specific `AttemptCounter` to be reset on either a successful `sendKey` subfunction request or an expired `SecurityDelayTimer`. However, ISO14229-1:2013 says nothing about the `AttemptCounter` being reset when the delay timer expires. In order to be ISO compatible, `SWS_Dcm_01357` is deviated so that the `AttemptCounter` is not reset upon an expired `SecurityDelayTimer`. `SWS_Dcm_01357` compliance may still be enabled via the vendor-specific parameter `DcmDspSecurityResetAttemptCounterOnTimeout` that should be configured to TRUE.

Requirements:

`SWS_Dcm_01357`

- ▶ ECU path used in `ImplementationDataType` `DataArrayType{DataType}_{{Data}}`

Affected AUTOSAR releases:

- ▶ R4.3.0

Description:

The mentioned ECUC paths should be:

- ▶ `{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataByteSize)}`
- ▶ `{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidDataByteSize)}`



- ▶ {ecuc(Dcm/DcmConfigSet/DcmDspData/DcmDspDataType)}
- ▶ {ecuc(Dcm/
 DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPid-
 DataType)}

Furthermore, in order to provide backwards-compatibility with AUTOSAR 4.0.3 bit-size of PID and DID signals, the AUTOSAR 4.0.3 parameters:

- ▶ {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspContentSize)}
 - ▶ {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidContentSize)}
- shall be used.

Furthermore, as a result of the fact that AUTOSAR 4.3 compatibility is not implemented for PID signal configuration, the parameter `DcmDspPidDataType` does not exist and therefore the necessary array type for PIDs, `DataArrayTypeUint8_{Data}Type`, is generated without the '`Dcm_-`' prefix, in conformance to AUTOSAR 4.2.1. Furthermore, the variation of this type is:

- ▶ When the related PID is usable for service \$01 and
- ▶ When the interface chosen for the related PID is `DcmDspPidDataUsePort = USE_DATA_SYNCH_-CLIENT_SERVER`

Rationale:

In order to provide backwards-compatibility with AUTOSAR 4.0.3 bit-size of PID and DID signals, the AUTOSAR 4.0.3 parameters `DcmDspContentSize` and `DcmDspPidContentSize` are used instead of `DcmDspDataByteSize` and `DcmDspDataType` from AUTOSAR 4.3.0. The ECUC paths are changed since the ones in the AUTOSAR 4.0.3 specification:

- ▶ {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspContentSize)}
- ▶ {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.DcmDspPidContentSize)}
- ▶ {ecuc(Dcm/DcmConfigSet/DcmDspData.DcmDspDataType)}
- ▶ {ecuc(Dcm/
 DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01.DcmDspPid-
 DataType)}

are non-existent.

Requirements:

`SWS_Dcm_01121`, `SWS_Dcm_01122`, `SWS_Dcm_01123`, `SWS_Dcm_01124`, `SWS_Dcm_01125`, `SWS_Dcm_01126`

- ▶ No support for `DcmDspData USE_DATA_ASYNCH_FNC_ERROR` and `USE_DATA_ASYNCH_CLIENT_SERVER_ERROR` interface types

Affected AUTOSAR releases:



► R4.3.0

Description:

Support for USE_DATA_ASYNCH_FNC_ERROR and USE_DATA_ASYNCH_CLIENT_SERVER_ERROR for DcmDspData elements is not available.

Rationale:

These interface types have been added in a version of the AUTOSAR Dcm SWS which is later than 4.0.-3. Only select functionality from later versions of AUTOSAR is available.

Requirements:

SWS_Dcm_00439, SWS_Dcm_00436, SWS_Dcm_00437, SWS_Dcm_00395, SWS_Dcm_91005, SWS_Dcm_91008, SWS_Dcm_91009, SWS_Dcm_91010, SWS_Dcm_91011, SWS_Dcm_00686 - ASR43, SWS_Dcm_01035, SWS_Dcm_91012, SWS_Dcm_CONSTR_6060, SWS_Dcm_01017, SWS_Dcm_00686

- The Dcm supports sender/receiver interfaces return error codes according to AUTOSAR 4.3.0 and not according to AUTOSAR 4.0.3

Affected AUTOSAR releases:

► R4.0.3

Description:

The Dcm supports the following return error codes for sender/receiver interfaces as specified in AUTOSAR 4.3.0 in addition to the ones specified in AUTOSAR 4.0.3:

- RTE_E_OUT_OF_RANGE
- RTE_E_COM_BUSY
- RTE_E_COM_STOPPED
- RTE_E_HARD_TRANSFORMER_ERROR
- RTE_E_SOFT_TRANSFORMER_ERROR

Rationale:

The sender/receiver interfaces implementation is based on AUTOSAR 4.3.0 specifications.

Requirements:

Dcm040

- DCM does not support RoutineControl checks(security level/session/mode condition) per operation



Affected AUTOSAR releases:

- ▶ CP Release 4.3.0
- ▶ R4.2 Rev 1

Description:

AUTOSAR 4.2.1 and AUTOSAR 4.3.0 specify that the RoutineControl checks(security level/session/mode condition) are per operation(DcmDspStartRoutineCommonAuthorizationRef, DcmDspStopRoutineCommonAuthorizationRef and DcmDspRequestRoutineResultsCommonAuthorizationRef) and this is not supported by DCM. DCM supports the RoutineControl checks(security level/session/mode condition) per configured Routine(DcmDspRoutineSecurityLevelRef, DcmDspRoutineSessionRef, DcmDspRoutineModeRuleRef).

Requirements:

SWS_Dcm_00570, SWS_Dcm_00571, SWS_Dcm_00570_ASR42, SWS_Dcm_00571_ASR42, SWS_Dcm_00824_ASR42, SWS_Dcm_01169, SWS_Dcm_01170, SWS_Dcm_01171, SWS_Dcm_01141, SWS_Dcm_01141_ASR42

- ▶ DCM does not support the AUTOSAR 4.3.0 configuration naming for RoutineControl

Affected AUTOSAR releases:

- ▶ CP Release 4.3.0

Description:

The AUTOSAR 4.3.0 configuration naming for RoutineControl:

- ▶ DcmDspRequestRoutineResultsFnc
- ▶ DcmDspRequestRoutineResultsOut
- ▶ DcmDspRequestRoutineResultsOutSignal
- ▶ DcmDspStopRoutineFnc
- ▶ DcmDspStopRoutineInSignal
- ▶ DcmDspStopRoutineOut
- ▶ DcmDspStopRoutineOutSignal
- ▶ DcmDspRoutineParameterSize

are changed for backward compatibility. The AUTOSAR 4.0.3 configuration naming is used.

Requirements:



ECUC_Dcm_00753, ECUC_Dcm_00831, ECUC_Dcm_00836, ECUC_Dcm_00838, ECUC_Dcm_00752, ECUC_Dcm_00832 ECUC_Dcm_00833, ECUC_Dcm_00842, ECUC_Dcm_00664, ECUC_Dcm_00847, ECUC_Dcm_00850, ECUC_Dcm_00839, ECUC_Dcm_00841, ECUC_Dcm_00844

- ▶ DCM does not support all configuration containers for RoutineControl from AUTOSAR 4.3.0

Affected AUTOSAR releases:

- ▶ CP Release 4.3.0

Description:

The AUTOSAR 4.3.0 configuration containers for RoutineControl:

- ▶ DcmDspRoutineInfoByte
- ▶ DcmDspRequestRoutineResults
- ▶ DcmDspRequestRoutineResultsCommonAuthorizationRef
- ▶ DcmDspStopRoutine
- ▶ DcmDspStopRoutineCommonAuthorizationRef
- ▶ DcmDspStartRoutine
- ▶ DcmDspStartRoutineCommonAuthorizationRef

are not supported.

Requirements:

ECUC_Dcm_01063, ECUC_Dcm_01023, ECUC_Dcm_01054, ECUC_Dcm_01022, ECUC_Dcm_01053, ECUC_Dcm_01021, ECUC_Dcm_01052

- ▶ DCM does not support confirmation operation for RoutineControl from AUTOSAR 4.3.0

Affected AUTOSAR releases:

- ▶ CP Release 4.3.0

Description:

The AUTOSAR 4.3.0 confirmation configuration containers and operations for RoutineControl:

- ▶ DcmDspRequestRoutineResultsConfirmationEnabled
- ▶ DcmDspRequestRoutineResultsConfirmationFnc
- ▶ DcmDspStopRoutineConfirmationEnabled
- ▶ DcmDspStopRoutineConfirmationFnc
- ▶ DcmDspStartRoutineConfirmationEnabled
- ▶ DcmDspStartRoutineConfirmationFnc



- ▶ Xxx_StartConfirmation
 - ▶ Xxx_StopConfirmation
 - ▶ Xxx_RequestResultsConfirmation
- are not supported.

Requirements:

ECUC_Dcm_01091, ECUC_Dcm_01090, ECUC_Dcm_01095, ECUC_Dcm_01096, ECUC_Dcm_01093, ECUC_Dcm_01094, SWS_Dcm_91016, SWS_Dcm_91017, SWS_Dcm_91018

- ▶ DCM does not follow the NRC handling for RoutineControl service

Affected AUTOSAR releases:

- ▶ CP Release 4.3.0
- ▶ R4.2 Rev 1

Description:

AUTOSAR 4.2.1 and AUTOSAR 4.3.0 specify that the NRCs order for RoutineControl service shall follow the ISO 14229-1/2013.(see Figure 25 — NRC handling for RoutineControl service from ISO 14229-1/2013) ISO 14229-1/2013 does not specify where the ModeRule check shall be made and also the "routineControlOptionRecord contains valid data for the requested RID" check shall be made by DCM. This check is not possible because just the application know if the routineControlOptionRecord contains valid data.

Requirements:

SWS_Dcm_01139, SWS_Dcm_01139_ASR42

- ▶ VARIABLE_LENGTH input signals for the UDS service RoutineControl operations are passed to the application using pointer to const.

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

In contrast to Dcm690, the UDS service RoutineControl operations use in their signature a pointer to const for the VARIABLE_LENGTH input signals.

Rationale:

This is done for compliance with the RTE where it is specified that all input parameters that are passed by reference or passed as an array expression shall be declared as pointer to const.

Requirements:



Dcm690

- ▶ The Dcm supports only `DcmDspDidControlMaskSize` values from 0 to 4.

Affected AUTOSAR releases:

- ▶ CP Release 4.4.0

Description:

AUTOSAR 4.4.0 specifies that `DcmDspDidControlMaskSize` values range is from 0 to 4294967294. Currently the Dcm supports only values range from 0 to 4 as specified by AUTOSAR 4.3.0.

Requirements:

`SWS_Dcm_01285`, `SWS_Dcm_01286`, `SWS_Dcm_01290`, `SWS_Dcm_00802`, `ECUC_Dcm_01060`

- ▶ The Dcm provides configurable interfaces declarations via `Dcm_API_Cfg.h`

Affected AUTOSAR releases:

- ▶ CP Release 4.4.0

Description:

AUTOSAR 4.4.0 specifies that configurable interfaces declarations shall be available via `Dcm_Externals.h`. Currently the Dcm provides configurable interfaces declarations via `Dcm_API_Cfg.h`.

Rationale:

In order to export configurable interfaces declarations a generated header file is necessary while `Dcm_Externals.h` is a static file, i.e. not generated.

Requirements:

`SWS_Dcm_01285`, `SWS_Dcm_01286`, `SWS_Dcm_01290`, `SWS_Dcm_00802`

- ▶ Interpretation of the value 0xFF of the parameter `DcmDslDiagRespMaxNumRespPend` by DCM

Affected AUTOSAR releases:

- ▶ CP Release 4.3.0

Description:

AUTOSAR 4.0.3 (`Dcm693_Conf`) specifies that the value 0xFF of the configuration parameter `DcmDslDiagRespMaxNumRespPend` shall mean that there is no limit on the number of NRC 0x78 (request-CorrectlyReceived-ResponsePending) to be transmitted by the DCM. However, whether an infinite amount of requestCorrectlyReceived-ResponsePending (0x78) NRCs is sent depends on the configured value of the parameter `DcmDslDiagRespMaxNumRespPendInfinityLimit`:



- ▶ If the configuration parameter `DcmDslDiagRespMaxNumRespPendInfinityLimit` is set to "255", then the parameter `DcmDslDiagRespMaxNumRespPend` can have a value from 0 to 255 (inclusive) where 255 (0xFF) is treated as the infinity limit for transmitting NRC 0x78 by the DCM.
 - ▶ If the configuration parameter `DcmDslDiagRespMaxNumRespPendInfinityLimit` is set to "65535", then the parameter `DcmDslDiagRespMaxNumRespPend` can have a value from 0 to 65535 (inclusive) where 65535 (0xFFFF) is treated as the infinity limit for transmitting NRC 0x78 by the DCM.
- The maximum number of requestCorrectlyReceived-ResponsePending (0x78) NRCs to be sent can thus be increased from 254 to 65534.

Rationale:

To allow DCM to be configured to increase the number of NRC 0x78 that can be transmitted from 0 - 254 (inclusive) to 0 - 65534 (inclusive). The new value indicating no limit on number of NRC 0x78 to be transmitted by DCM is 65535 instead of old value 255.

Requirements:

Dcm693_Conf

- ▶ For OBD service \$04, Dcm transmits NRC 0x22 instead of positive response after reaching the configured limit for number of sent RCRRPs

Affected ISO releases:

- ▶ ISO 15031-5- 2011-04-15

Description:

For OBD service \$04 (Clear/Reset emission-related diagnostic information), after the Dcm reaches the configured maximum number of NRCs with the negative response code 0x78 (requestCorrectlyReceivedResponsePending), it transmits a NRC response 0x22 (conditionsNotCorrect) instead of positive response. To avoid this scenario where the NRC 0x22 (conditionsNotCorrect) response is transmitted by the Dcm instead of positive response, the maximum number of RCRRPs (requestCorrectlyReceivedResponsePending) can be configured to infinity in Tresos.

Rationale:

The 0x10 NRC (generalReject) is not supported for the OBD service \$04 - Clear/Reset emission-related diagnostic information. Table 10 - Proper response from server/ECU with ISO 9141-2, ISO 14230-4 and SAE J1850 protocol Table 11 - Proper response from server/ECU for ISO 15765-4 protocol Table 16 - Negative response code (NRC) definition

Requirements:

Dcm120



- ▶ For OBD service \$04, Dcm transmits NRC 0x22 instead of NRC 0x10 after reaching the configured limit for number of sent RCRRPs

Affected AUTOSAR releases:

- ▶ CP Release 4.0.3

Description:

AUTOSAR 4.0.3 specifies that after reaching the value defined in the configuration parameter `DcmD-s1DiagRespMaxNumRespPend` the DCM module should stop processing of the active diagnostic request and shall send a negative response with NRC 0x10(GeneralReject). Instead of sending NRC 0x10(GeneralReject) for service 0x04 - Clear/Reset emission-related diagnostic information the Dcm will send a negative response with NRC 0x22(ConditionsNotCorrect).

Rationale:

The 0x10 NRC (generalReject) is not supported for the OBD service \$04 - Clear/Reset emission-related diagnostic information according to ISO 15031-5- 2011-04-15.

Requirements:

Dcm120

- ▶ `AttemptCounter` is not incremented according to ISO14229-1:2013 requirements

Affected AUTOSAR releases:

- ▶ ISO14229-1:2013

Description:

Security `AttemptCounter` is not incremented according to ISO14229-1:2013 requirements in the following circumstances:

- ▶ A SecurityAccess (0x27) service request for the CompareKey (2*SecurityLevel) subfunction for a Security Level is received which does not occur after a successful SecurityAccess (0x27) service request for the Request (2*SecurityLevel-1) subfunction for the same Security Level.
- ▶ A SecurityAccess (0x27) service request for the CompareKey (2*SecurityLevel) subfunction for a Security Level is received which occurs after a successful SecurityAccess (0x27) service request for the Request (2*SecurityLevel-1) subfunction for the same Security Level but the request for the CompareKey (2*SecurityLevel) subfunction has an incorrect length.

Rationale:

These two scenarios are not handled by the AUTOSAR.

Requirements:



Table I.2 - State transitions - disjunctive normal form representation

- ▶ DCM does not perform check for permission to clear DTC for the UDS service `ClearDiagnosticInformation` (0x14)

Affected AUTOSAR releases:

- ▶ CP Release 4.4.0

Description:

Dcm does not check whether application allows to clear DTC.

Requirements:

SWS_Dcm_01268

- ▶ DCM does not perform check for mode condition for the UDS service `ClearDiagnosticInformation` (0x14)

Affected AUTOSAR releases:

- ▶ CP Release 4.4.0

Description:

Dcm does not check if DTC can be cleared in current mode condition.

Requirements:

SWS_Dcm_01269

- ▶ Subfunction `reportDTCBySeverityMaskRecord` (0x08) of service `ReadDTCInformation` (0x19)

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

`reportDTCBySeverityMaskRecord` (0x08) subfunction of UDS service `ReadDTCInformation` (0x19) is implemented according to AUTOSAR CP Release 4.4.0.

Requirements:

Dcm379, Dcm380

- ▶ Subfunction `reportSeverityInformationOfDTC` (0x09) of service `ReadDTCInformation` (0x19)



Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

`reportSeverityInformationOfDTC (0x09)` subfunction of UDS service `ReadDTCInformation (0x19)` is implemented according to AUTOSAR CP Release 4.4.0.

Requirements:

Dcm381

- ▶ `ComM_DCM_ActiveDiagnostic()` is called in a non-default session as well

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

When in a non-default session, `ComM_DCM_ActiveDiagnostic()` is called for ComMChannels for which ActiveDiagnostic has not been requested, if a new request occurs on such a channel. This is the situation in which two different RxPduIDs reference two different ComM channels, but belong to the same DcmDslProtocol. In such a situation, if these RxPduIDs receive requests one after another, both these requests must be allowed to transmit their responses.

Requirements:

Dcm169

- ▶ `ComM_DCM_InactiveDiagnostic()` is invoked when `Dcm_TpTxConfirmation()` is called in the context of failed `RequestCorrectlyReceivedResponsePending (0x78)` NRC transmissions as well.

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

`RequestCorrectlyReceivedResponsePending (0x78)` NRC transmissions should not result in the Dcm going to Inactive Diagnostic on a ComMChannel, but in the case in which the transmission has failed, this leads to the request being terminated. This is equivalent to a final response.

Requirements:

Dcm165



- ▶ If Dem_EnableDTCSetting() is called according to [SWS_Dcm_00751] and [SWS_Dcm_00752] and returns DEM_PENDING, the Dcm does not call this function again.

Affected AUTOSAR releases:

- ▶ CP Release 4.4.0

Description:

In case DTCSetting is disabled and shall be re-enabled (see [SWS_Dcm_00751] and [SWS_Dcm_00752]), the Dcm does not call Dem_EnableDTCSetting() again if the first call has returned DEM_PENDING. If Dem provides an asynchronous implementation of Dem_EnableDTCSetting(), i.e. E_OK is not returned with the first call, Dcm expects that Dem processes the EnableDTCSetting request to the end without any subsequent calls until E_OK is returned. During this request processing by Dem, any other EnableDTCSetting or DisableDTCSetting request with same or different ClientId shall be possible and managed by Dem.

Requirements:

SWS_Dcm_01412

- ▶ Subfunction reportNumberOfDTCBySeverityMaskRecord (0x07) of service ReadDTCInformation (0x19)

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

reportNumberOfDTCBySeverityMaskRecord (0x07) subfunction of UDS service ReadDTCInformation (0x19) is implemented according to AUTOSAR CP Release 4.4.0.

Requirements:

Dcm293

- ▶ When receiving a request for reportUserDefMemoryDTCByStatusMask (0x17), reportUserDefMemoryDTCSnapshotRecordByDTCNumber (0x18) or reportUserDefMemoryDTCExtDataRecordByDTCNumber(0x19), the Dcm does not check if the access to the selected user defined memory in parameter MemorySelection is authenticated.

Affected AUTOSAR releases:

- ▶ CP Release 4.4.0

Description:

Dcm does not support service 0x29 - Authentication.



Requirements:

SWS_Dcm_01545, SWS_Dcm_01546, SWS_Dcm_01547

- ▶ When receiving a request for UDS service ReadDTCInformation (0x19) with subservice reportDTCSnapshotIdentification (0x03), the Dcm does not call the API Dem_GetNumberOfFreezeFrameRecords().

Affected AUTOSAR releases:

- ▶ CP Release 4.4.0

Description:

In case UDS Service ReadDTCInformation is requested with subservice reportDTCSnapshotIdentification (0x03) the Dcm does not call Dem_GetNumberOfFreezeFrameRecords(). The order of API calls is Dem_SetFreezeFrameRecordFilter() followed by Dem_GetNextFilteredRecord() according to AUTOSAR 4.3.1.

Requirements:

SWS_Dcm_00298

- ▶ The Dcm-Dem interface implementation is based on AUTOSAR 4.3.1 specification with corrections from AUTOSAR 4.4.0 specification.

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3
- ▶ R4.2.1
- ▶ R4.2.2

Description:

The Dcm implements Dem API calls based on AUTOSAR 4.3.1 for the following services: - UDS service 0x14 - ClearDiagnosticInformation - UDS service 0x19 - ReadDTCInformation - UDS service 0x85 - ControlDTCSetting - UDS service 0xAF - ReadGenericInformation - OBD service \$04 - Clear/Reset emission-related diagnostic information - OBD service \$03 - Request emission-related diagnostic trouble codes - OBD service \$07 - Request emission-related diagnostic trouble codes detected during current or last completed driving cycle - OBD service \$0A - Request emission-related diagnostic trouble codes with permanent status - OBD service \$02 - Request powertrain freeze frame data with corrections from AUTOSAR 4.4.0, such as Dem_GetDTCSelectionResultForClearDTC() API shall be used by service 0x14 instead of Dem_GetDTCSelectionResult() API. This means that, with respect to AUTOSAR 4.0.3 and to extension from AUTOSAR 4.2, the signatures of several Dem APIs are changed, some Dem APIs are no longer supported and some new Dem APIs are introduced. These interfaces are not anymore backward compatible. AUTOSAR 4.2 extensions for example are: - new subservices 0x17, 0x18, 0x18 for service 0x19 supporting user defined memory - new error codes DEM_CLEAR_BUSY and DEM_CLEAR_MEMORY_ERROR



for Dem_ClearDTC() API of services 0x14 and \$04 - optional parameter DTCSettingControlOptionRecord for service 0x85

Rationale:

The Dem API calls as specified in AUTOSAR 4.3.1 with corrections from AUTOSAR 4.4.0 are necessary to support parallel OBD and UDS protocol processing, i.e. to allow parallel access to the event related data and fault memory by multiple diagnostic requests at the same time.

Requirements:

Dcm371, Dcm372, Dcm702, Dcm005, Dcm705, Dcm706, Dcm708, Dcm739, Dcm740, Dcm293, Dcm377, Dcm008, Dcm378, Dcm297, Dcm295, Dcm296, Dcm478, Dcm474, Dcm475, Dcm476, Dcm386, Dcm382, Dcm298, Dcm383, Dcm384, Dcm385, Dcm441, Dcm393, Dcm466, Dcm766, Dcm465, Dcm304, Dcm783, Dcm406, Dcm784, Dcm751, Dcm752, Dcm077, Dcm289, Dcm412, Dcm004, Dcm413, Dcm703, SWS_-Dcm_00008, SWS_Dcm_00378, SWS_Dcm_00297, SWS_Dcm_00295, SWS_Dcm_00296, SWS_-Dcm_00474, SWS_Dcm_00475, SWS_Dcm_00476, SWS_Dcm_00386, SWS_Dcm_00302, SWS_-Dcm_01147, SWS_Dcm_01148, SWS_Dcm_01149, SWS_Dcm_00387, SWS_Dcm_00304, SWS_Dcm_-01063, SWS_Dcm_01064, SWS_Dcm_00783, SWS_Dcm_00406, SWS_Dcm_00784, SWS_Dcm_-00751, SWS_Dcm_00752, SWS_Dcm_00830, SWS_Dcm_00841, SWS_Dcm_00393, SWS_Dcm_-00466, SWS_Dcm_00766, SWS_Dcm_00289, SWS_Dcm_00412, SWS_Dcm_00330, SWS_Dcm_-00004, SWS_Dcm_00413, SWS_Dcm_00703, SWS_Dcm_01250, SWS_Dcm_01251

- ▶ Subfunction reportWWHOBDDTCByMaskRecord (0x42) of service ReadDTCInformation (0x19)

Affected AUTOSAR releases:

- ▶ CP Release 4.4.0

Description:

reportWWHOBDDTCByMaskRecord (0x42) subfunction of UDS service ReadDTCInformation (0x19) is not supported.

Requirements:

SWS_Dcm_01127, SWS_Dcm_01128, SWS_Dcm_01129, SWS_Dcm_01130, SWS_Dcm_01131

- ▶ Subfunction reportWWHOBDDTCWithPermanentStatus (0x55) of service ReadDTCInformation (0x19)

Affected AUTOSAR releases:

- ▶ CP Release 4.4.0

Description:



reportWWHOBDDTCWithPermanentStatus (0x55) subfunction of UDS service
ReadDTCInformation (0x19) is not supported.

Requirements:

SWS_Dcm_01343, SWS_Dcm_01344, SWS_Dcm_01345, SWS_Dcm_01346

- ▶ OBD support (reference to product description: ASCPD-33)

Affected AUTOSAR releases:

- ▶ R4.0 Rev 3

Description:

Dcm provides partial support for OBD services.

Currently not supported OBD DID ranges:

- ▶ 0xF500-0xF5FF
- ▶ 0xF700-0xF7FF

Currently NOT supported OBD services:

- ▶ Service 0x05 - Request oxygen sensor monitoring test results:

This Service is not supported by the Dcm.

Note: This is no deviation of the AUTOSAR specification, because it is not required here.

- ▶ For OBD service Request Vehicle Information 0x09, Dcm transmits NRC 0x22 instead of 0x12

Affected AUTOSAR releases:

- ▶ CP Release 4.4.0

Description:

For OBD service \$09 (Request Vehicle Information), if Xxx_GetInfoTypeValueData operation returns values different than E_OK or E_PENDING, then Dcm shall return NRC 0x22 instead of 0x12.

Rationale:

According to "Table 11 — Proper response from server/ECU for ISO 15765-4 protocol" from "ISO 15031-5:2011", the NRC 0x12 is not supported by OBD Service Request Vehicle Information 0x09. (Refer to AUTOSAR Jira <https://jira.autosar.org/browse/AR-96256>)

Requirements:



SWS_Dcm_01191

- ▶ Subfunction checks for Session/Security/ModeRule for UDS Service Routine Control (0x31).

Affected AUTOSAR releases:

- ▶ R4.0.3

Description:

For UDS service Routine Control (0x31), subfunction checks for Session/Security/ModeRule are not performed.

Rationale:

According to "Figure 5 — General server response behaviour" from "ISO 14229-1/2013", the checks for Session/Security/ModeRule are explicitly skipped.

- ▶ Subfunction `reportMirrorMemoryDTCExtDataRecordByDTCNumber (0x10)` of service `Read-DTCInformation (0x19)`

Affected AUTOSAR releases:

- ▶ CP Release 4.4.0

Description:

`reportMirrorMemoryDTCExtDataRecordByDTCNumber (0x10)` subfunction of UDS service `Read-DTCInformation (0x19)` is not supported.

Requirements:

SWS_Dcm_00371

- ▶ UDS Service `EcuReset (0x11)` for subfunctions `enableRapidPowerShutDown (0x04)` and `disableRapid-PowerShutDown (0x05)`.

Affected AUTOSAR releases:

- ▶ R4.2.1
- ▶ R4.2.2
- ▶ R4.3.0
- ▶ R4.3.1
- ▶ R4.4.0

Description:



For UDS Service EcuReset (0x11) with subfunctions enableRapidPowerShutDown (0x04) and disableRapidPowerShutDown (0x05) further requests to the Dcm shall not be ignored.

Rationale:

Ignoring further requests after UDS Services EcuReset (0x11) with subfunctions enableRapidPowerShutdown (0x04) and disableRapidPowerShutdown (0x05) would be too restrictive as it allows no further requests after these. For example, if you enable rapid power shutdown with 0x04, you won't be able to disable it again as the 0x05 request would be rejected.

Requirements:

SWS_Dcm_00834

- ▶ Multiplicity of `DcmDemClientRef`

Affected AUTOSAR releases:

- ▶ CP Release 4.4.0

Description:

Multiplicity of `DcmDemClientRef` is changed from 1 to 0..1.

Rationale:

The `Dcm` should be able to generate and compile with no `Dem` module/stub present regardless of the availability of UDS Services 0x19, 0x14, 0x85 or OBD Services.

Requirements:

ECUC_Dcm_01083

- ▶ For OBD service \$09, Dcm transmits a negative response with NRC ConditionsNotCorrect (0x22) after reaching the configured limit for number of sent RCRRPs

Affected ISO releases:

- ▶ ISO 15031-5- 2011-04-15

Description:

For OBD service \$09 (Request vehicle information), after the Dcm reaches the configured maximum number of NRCs with the negative response code 0x78 (requestCorrectlyReceivedResponsePending), it transmits a negative response with NRC ConditionsNotCorrect (0x22).

Rationale:



The NRC 0x10 (generalReject) is not supported for the OBD service \$09 - Request vehicle information. Please refer the following: Table 10 - Proper response from server/ECU with ISO 9141-2, ISO 14230-4 and SAE J1850 protocol Table 11 - Proper response from server/ECU for ISO 15765-4 protocol Table 16 - Negative response code (NRC) definition

Requirements:

Dcm120

- ▶ For OBD service \$09, Dcm transmits a negative response with NRC ConditionsNotCorrect (0x22) after reaching the configured limit for number of sent RCRRPs.

Affected AUTOSAR releases:

- ▶ CP Release 4.0.3

Description:

AUTOSAR 4.0.3 specifies that after reaching the value defined in the configuration parameter `DcmD-s1DiagRespMaxNumRespPend` the DCM module should stop processing of the active diagnostic request and shall send a negative response with NRC 0x10(GeneralReject). Instead of sending NRC 0x10(GeneralReject) for service 0x09 - Request vehicle information the Dcm shall sent a negative response with NRC ConditionsNotCorrect (0x22).

Rationale:

The 0x10 NRC (generalReject) is not supported for the OBD service \$09 - Request vehicle information according to ISO 15031-5- 2011-04-15.

Requirements:

Dcm120

- ▶ For OBD service Request Vehicle Information 0x09, Dcm transmits NRC 0x22 instead of 0x10

Affected AUTOSAR releases:

- ▶ CP Release 4.4.0

Description:

When OBD service \$09 Request vehicle information is requested and UDS service (0x22) ReadDataByIdentifier with DID inside the OBD InfoType range (F800-F8FF) is already running, the Dcm might transmit a RCRRP (0x78) before no final response or negative response with NRC CNC (0x22)

Rationale:



According to "Table 11 - Proper response from server/ECU for ISO 15765-4 protocol" from "ISO 15031-5:2011", the NRC 0x10 is not supported by OBD Service Request Vehicle Information 0x09.

Requirements:

Dcm120

- ▶ DcmDspCommonAuthorizationRef configuration parameter is not supported.

Affected AUTOSAR releases:

- ▶ CP Release 4.2.1

Description:

The Dcm module does not support the DcmDspCommonAuthorizationRef configuration parameter.

Rationale:

According to "Constr_6032" from CP Release 4.2.1, DcmDspCommonAuthorizationRef should be disabled if DcmDspRoutineTidRef is used.

Requirements:

constr_6032

- ▶ For UDS service 0x24 (ReadScalingDataByIdentifier), the Dcm does not support NRC 0x34 (authenticationRequired)

Affected ISO 14229-1 releases:

- ▶ ISO14229-1:2020

Description:

According to "Table 210 - Supported negative response codes" in ISO14229-1:2020, on receiving the UDS service 0x24 (ReadScalingDataByIdentifier) request, the Dcm shall send a response with NRC 0x34 (authenticationRequired) if the Data Identifier (DID) is secured and the client has insufficient rights based on its Authentication state.

However, the NRC 0x34 (authenticationRequired) is currently not supported by the Dcm.

Rationale:

The NRC 0x34 (authenticationRequired) is related to service 0x29 (Authentication Service) which is currently not supported by the Dcm.

Requirements:



Dcm.EB.ReadScalingDataByIdentifier.AuthenticationRequired.NRC0x34

- ▶ Function interface not specified

Affected AUTOSAR releases:

- ▶ ASR R20-11

Description:

The requirements SWS_Dcm_00798 and SWS_Dcm_91012 specify the synchronous and asynchronous interfaces for function as well as client-server operations respectively. However, since the requirement SWS_Dcm_00686 also specifies the interfaces for synchronous and asynchronous client-server operations, the scope of the requirements SWS_Dcm_00798 and SWS_Dcm_91012 is limited only to the interface definitions for synchronous and asynchronous function operations.

Rationale:

This deviation is required to avoid duplicate requirements regarding the definition of synchronous and asynchronous interfaces for client-server operations.

Requirements:

SWS_Dcm_00798, SWS_Dcm_91012

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.

- ▶ **Integration Note:** In addition to integration requirements documented in the current release notes, refer to the Dcm module references and to the EB tresos AutoCore Generic 8 documentation, Diagnostics/Error handler user's guide for further important information regarding the integration of the Dcm module.
- ▶ No support for link and post build time configuration (reference to product description: ASCPD-77)

Description:

The Dcm module can only be configured at pre-compile time. Link and post build time configuration is not supported.

Rationale:

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

Requirements:



Dcm054, Dcm172, Dcm173, VCC-DCM-007:Req22v1, VCC-DCM-007:Req23v1

- ▶ Handling of ServiceToRespond record in service ResponseOnEvent (0x86) (reference to product description: ASCPD-75)

Description:

The current implementation supports only the services 0x22 and 0x19 with subfunction 0x02 in ServiceToRespond record of the ResponseOnEvent service request.

Rationale:

Feature implementation has been limited to services ReadDataByIdentifier (0x22) and subfunction reportDTCByStatusMask (0x02) of service ReadDTCInformation (0x19) in current release.

Requirements:

Section 7.2.4.7

- ▶ Support for the container DcmDsdSubService

Description:

Subfunction level configuration support is provided for all UDS services except for RoutineControl (0x31).

Rationale:

Service RoutineControl (0x31) configuration is done via configuration parameters in the dedicated container DcmDspRoutine.

Requirements:

Dcm802_Conf

- ▶ Limited value of DcmDspSessionLevel parameter

Description:

According to Dcm SWS of AUTOSAR R4.0, the value of the parameter DcmDspSessionLevel ranges from 0 to 255. For the current implementation, the range is limited from 1 to 255.

Rationale:

The value of the default session level is 1. So it is limited from 1 to 255.

Requirements:

Dcm765_Conf



- ▶ Limited the value of parameter `DcmModeRuleNrcValue`

Description:

According to Dcm SWS of AUTOSAR R4.0, the value of the parameter `DcmModeRuleNrcValue` ranges from 0 to 255. But for the current implementation, the range is limited from 1 to 255.

Rationale:

Requires clarification from Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=59277

Requirements:

`Dcm949_Conf`

- ▶ Limited the value of parameter `DcmTaskTime`

Description:

According to Dcm SWS of AUTOSAR R4.0 Rev 3, the value of the parameter `DcmTaskTime` ranges from 0 to 100 seconds. But for the current implementation, the range is limited from 0.001 to 1 second.

Rationale:

A minimum value of 1 ms and a maximum value of 1 s should be sufficient for most use cases. Refer to AUTOSAR Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=56254.

`Dcm820_Conf`

- ▶ Limited multiplicity of `DcmDslProtocolRow`

Description:

According to Dcm SWS of AUTOSAR R4.0 Rev 3, the container `DcmDslProtocolRow` has an infinite upper multiplicity from 1 to *. But for the current implementation, the range is limited from 1 to 256.

Rationale:

To optimize the configuration, multiplicity is changed.

Requirements:

Sections 10.2.11, 10.2.12, `Dcm694_Conf`, `Dcm695_Conf`

- ▶ Limited timing unit reference of `DcmDspSecurityDelayTime` parameter value

Description:

Dcm SWS of AUTOSAR R4.0 Rev 3 contains contradicting statements regarding the timing unit reference of the parameter `DcmDspSecurityDelayTime`. It specifies both milliseconds and seconds which affect the specified range from 0 to 20.000. If milliseconds are applicable this would mean the higher range value



is 20 s. This value is less than what might be required by some applications. Thus the range is limited and corrected from 0 ms to 20.000 ms to a range of 0s to 65.335 s. This is also in accordance to Dcm SWS of AUTOSAR R4.1 Rev 1.

Rationale:

Dcm757_Conf High range value of 20.0000 ms is too less for some use cases. Refer to AUTOSAR Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=52073.

- ▶ Limited the value of parameters DcmDspSecurityAdrSize, DcmDspSecuritySeedSize, and DcmDspSecurityKeySize

Description:

According to Dcm SWS of AUTOSAR R4.0, the value of the parameters DcmDspSecurityAdrSize, DcmDspSecuritySeedSize, and DcmDspSecurityKeySize ranges from 1 to 4294967295. For the current implementation, the range is limited from 1 to 4294967293.

Rationale:

A security access service request or response consists of securityAccessDataRecord, securitySeed, or securityKey in addition to the service ID and subfunction ID. The service ID and subfunction ID occupy one byte each. This means that the payload has a maximum size equal to the maximum size of the containing buffer minus 2.

Requirements:

Dcm765_Conf Dcm725_Conf Dcm755_Conf

- ▶ Limited the range of DcmDspRoutineSignalLength

Description:

According to DCM SWS of AUTOSAR 4.0.3, the value of the parameter DcmDspRoutineSignalLength ranges from 0 to 65535. But for the current implementation, the range is limited from 1 to 65535.

Rationale:

A signal length of 0 is not useful.

Requirements:

Dcm838_Conf, Dcm841_Conf, Dcm844_Conf, Dcm847_Conf, Dcm850_Conf

- ▶ Limited multiplicity of DcmDspPidData

Description:

According to Dcm SWS of AUTOSAR R4.0 Rev 3, the container DcmDspPidData has an infinite upper multiplicity. But for the current implementation, the range is limited from 1 to 255.



Rationale:

The interface to read data from Dem (Dem_ReadDataOfOBDFreezeFrame) requires an uint8 parameter as index of PidData.

Requirements:

ECUC_Dcm_00865

- ▶ Limited type of source data identifier of a Dynamically Defined Data Identifier

Description:

According to ISO 14229-1 it is not prohibited by the UDS service DynamicallyDefineDataIdentifier (0x2C) but it is not recommended for the client to reference one dynamically defined data record from another. The deletion of the referenced record could create data consistency problems within the referencing record. EB implementation returns NRC 0x31 in this case.

Rationale:

Deletion of the referenced record could create data consistency problems.

Requirements:

Dcm.EB.DynamicallyDefineDataIdentifier.ISO

- ▶ Limited range of DcmDslProtocolPreemptTimeout

Description:

According to Dcm SWS of AUTOSAR R4.2.2, the parameter DcmDslProtocolPreemptTimeout has a lower limit of 0 and an upper limit of 1. But for the current implementation, the lower limit must be above 0 and below the value configured for P2ServerMax of the default session of the new protocol.

Rationale:

The handling of NRC 0x78 (RequestCorrectlyReceivedResponsePending) is bound to a protocol and a session. It is not defined how the reception layer shall handle this before starting a protocol.

Requirements:

ECUC_Dcm_00698

- ▶ Removal order of duplicated periodic DIDs in a UDS service request ReadDataByPeriodicIdentifier (0x2A)

Description:

When filtering out duplicated periodic DIDs in a ReadDataByPeriodicIdentifier request the first of a duplicated periodic DID is kept and the other duplicates are removed.



Rationale:

Reuse of an already present, tested, and working filtering function. Unlikelihood of the "keep first duplicate" approach causing problems.

- ▶ UDS services DiagnosticSessionControl (0x10), CommunicationControl (0x28), ReadDataByPeriodicIdentifier (0x2A), and ResponseOnEvent (0x86) cannot be configured for asynchronous operation

Description:

Services DiagnosticSessionControl (0x10), CommunicationControl (0x28), ReadDataByPeriodicIdentifier (0x2A), and ResponseOnEvent (0x86) must always be set up to operate synchronously, i.e. the configuration flag `DcmAsyncServiceExecution` must be set to 'false'. Setting that flag to 'true' for these services results in an error when generating the project.

Rationale:

Not allowing the services DiagnosticSessionControl (0x10), CommunicationControl (0x28), ReadDataByPeriodicIdentifier (0x2A), and ResponseOnEvent (0x86) service handler to run in the background helps avoid race conditions that would have to be taken care of. Managing those possible race conditions would degrade the performance of the service and add avoidable complexity.

- ▶ When using DynamicallyDefinedDIDs, the Dcm's `Dcm_Init()` API must be called after `NvM_ReadAll()` finishes

Description:

When using DynamicallyDefinedDIDs, the Dcm's `Dcm_Init()` API must be called after the `NvM_ReadAll()` operation returns a final status, i.e. a status which is not `NVM_REQ_PENDING`.

Rationale:

The DynamicallyDefinedDID persistency is implemented by having an NvM block maintaining the mirror of the DynamicallyDefinedDID configuration data, which is retrieved by calling the NvM API `NvM_ReadAll`. The `NvM_ReadAll()` request is handled asynchronously which means that the operation might still be ongoing when `Dcm_Init()` is called and this leads to possibly undefined behavior.

- ▶ If the configuration parameter `DcmDataTransferServicesASRVersion` is set to the value `DATA_TRANSFER_SERVICES_AUTOSAR_403`, the optional parameters of `Dcm_ProcessRequestTransferExit` callout according to AUTOSAR 4.0.3 are not supported

Description:

When requesting `RequestTransferExit` (0x37), the following optional parameters of `Dcm_ProcessRequestTransferExit` callout are not supported:

- ▶ `uint8* ParameterRecord`
- ▶ `uint32 ParameterRecordSize`



Rationale:

According to AUTOSAR 4.0.3 the parameters are optional and they refer to manufacturer specific data. They are not supported in the current implementation.

- ▶ Internal ROE-triggered request shall result in a ConditionsNotCorrect (0x22) NRC

Description

An internal ROE-triggered request for the UDS service ReadDataByIdentifier (0x22), occurring at the same time as an external request shall result in a ConditionsNotCorrect (0x22) NRC when it is being serviced for either of the following services:

UDS service	ReadDataByIdentifier (0x22)
	InputOutputControlByIdentifier (0x2F)
	WriteDataByIdentifier (0x2E)
OBD service	RequestOnboardMonitoringTestResults (0x06)

Table 3.3. Affected services

Rationale:

This limitation is necessary because of the inability of the underlying services to distinguish calling contexts to the functions called for executing read-operations on DID signals in the application. In case of OBD service 6 there is a possible reentrancy issue that needs to be guarded against for the interface Xxx_-GetDTRValue.

- ▶ internal ROE-triggered request results in a ConditionsNotCorrect (0x22) NRC

An internal ROE-triggered request for service ReadDTCInformation (0x19) occurring at the same time as an external request results in a ConditionsNotCorrect (0x22) NRC when it is being serviced for either service:

- ▶ ClearDiagnosticInformation (0x14)
- ▶ ReadDTCInformation (0x19)
- ▶ ReadGenericInformation (0xAF)
- ▶ RequestPowerTrainFreezeFrameData (\$02)
- ▶ Request emission-related diagnostic trouble codes (\$03)
- ▶ Clear/Reset emission-related diagnostic information (\$04)
- ▶ RequestOnboardMonitoringTestResults (\$06)
- ▶ Request emission-related diagnostic trouble codes detected during current or last completed driving cycle (\$07)



- ▶ Request emission-related diagnostic trouble codes with permanent status (\$0A)

Rationale:

The limitation is necessary because of the inability to distinguish calling contexts to the functions called in the Dem module for executing needed operations.

- ▶ Limited the value of parameter `DcmDspPidDataPos`

Description:

According to Dcm SWS of AUTOSAR R4.0, the value of the parameter `DcmDspPidDataPos` ranges from 0 to 2040. But for the current implementation, the range is limited from 1 to 2039.

Rationale:

A `DcmDspPidDataPos` of 2040 (starting on 0) is out of range of possible 255 bytes.

Requirements:

`Dcm866_Conf`

- ▶ Limited the multiplicity of container `DcmDspSession` if configuration parameter `DcmModeDeclarationSupport` is set to true

Description:

According to Dcm SWS of AUTOSAR R4.0.3, the multiplicity of the container `DcmDspSession` is 1 to *. But for the current implementation, the multiplicity is limited to 1 if configuration parameter `DcmModeDeclarationSupport` is set to true.

This implies that different `DcmDslProtocolRow` cannot use different `DcmDslProtocolSessionRef` references. If selective session availability per protocol is needed, it can be achieved in the following way:

1. Configure different entries of the UDS service `DiagnosticSessionControl` (0x10) in different `DcmDsdServiceTable` containers.
2. Configure different `DcmDsdSubService` support for each UDS service `DiagnosticSessionControl` (0x10) entry.
3. Configure each `DcmDslProtocolRow` to use the appropriate `DcmDsdServiceTable`.

Rationale:

If container `DcmDspSession` contains more than one entry and the configuration parameter `DcmModeDeclarationSupport` is set to true, then there are two default sessions and thus two references are generated to an initial mode declaration. It is not possible to have two references to an initial mode declaration. See also https://www.autosar.org/bugzilla/show_bug.cgi?id=58711.



- ▶ Use of RequestTransferExit (0x37) is limited

Description:

If the configuration parameter `DcmDataTransferServicesASRVersion` is set to the value `DATA_TRANSFER_SERVICES_AUTOSAR_430`, the processing of `RequestTransferExit` (0x37) service on a protocol with a shared Rx/Tx buffer is not possible. If a shared Rx/Tx buffer is desired, the user must ensure that the request data is not overwritten when Dcm transmits the response so that the request data is available for subsequent calls. This can be done by storing the request data in a local copy.

A shared Rx/Tx buffer means that for a `DcmDslProtocolRow`, the configuration parameter `DcmDslProtocolTxBufferID` is the same as `DcmDslProtocolRxBufferID` of the receive channel belonging to the `DcmDslMainConnection` on which the request is processed.

Rationale:

This limitation is necessary because in a configuration with a shared Rx/Tx buffer, `transferRequestParameterRecord` and `transferResponseParameterRecord` point to the same address. This means if the user writes data in `transferResponseParameterRecord` via `Dcm_ProcessRequestTransferExit`, the request data is not longer available for potential calls of the `Svch` with `OpStatus DCM_PENDING`.

- ▶ Limited check for NRC 0x71 (transferDataSuspended) conditions

Description

Dcm cannot perform the check for NRC 0x71 (transferDataSuspended) conditions for a `TransferData` (0x36) request before the check for NRC 0x73 (wrongBlockSequenceCounter) conditions as specified in Figure 28 from ISO 4229-1-2013

Rationale:

According to SWS_Dcm_01173 from AUTOSAR 4.3.0, NRC 0x71 (transferDataSuspended) is the responsibility of the callout function (`Dcm_ReadMemory` for an upload request or `Dcm_WriteMemory` for a download request). The callout function is called after the check for NRC 0x73 (wrongBlockSequenceCounter) is done.

- ▶ Limited checks for UDS services `RequestDownload` (0x34) and `RequestUpload` (0x35)

Description:

Dcm can not perform the following checks for the UDS services `RequestDownload` (0x34) and `RequestUpload` (0x35):

- ▶ Validity of `dataFormatIdentifier` for `RequestDownload` (0x34) as specified in Figure 26 from ISO 4229-1-2013
- ▶ Validity of `dataFormatIdentifier` for `RequestUpload` (0x35) as specified in Figure 27 from ISO 4229-1-2013



Rationale:

Dcm has no available information to decide whether a `dataFormatIdentifier` received in a request is valid or not. Dcm defers this responsibility to the application, as specified in SWS_Dcm_01132 and SWS_Dcm_01133 from AUTOSAR 4.3.0.

- ▶ The feature which is implemented along with the introduction of parameter `DcmDsdDisableGenericServiceImplementation` has the following limitations

Description:

The NRC responses from the internal generic Service Handler shall not be changed.

The external service handler shall call the internal generic Service Handler only via the provided API.

The requests which are processed by the external service handler are bound to the same limitations as the internal generic service handlers.

Rationale:

Those limitations are determined by the method used to call internal generic Service Handlers from the external service handlers.

- ▶ The Dcm does not support the configuration values `DCM_CONTROLMASK_NO` and `DCM_CONTROLMASK_INTERNAL` for `DcmDspDidControlMask`

Description:

The Dcm supports only the following configuration values for `DcmDspDidControlMask`:

- ▶ `DCM_CONTROLMASK_EXTERNAL`

`DCM_CONTROLMASK_NO` is not supported

`DCM_CONTROLMASK_INTERNAL` is not supported

Rationale:

The functionality related to `DCM_CONTROLMASK_NO` can be obtained by disabling the `DcmDspDidControlMask` parameter but in this case the number of signals of the controlled DID is limited to one. The functionality related to `DCM_CONTROLMASK_INTERNAL` can be obtained by disabling the `DcmDspDidControlMask` parameter and relying on AUTOSAR 4.0.3 behaviour. This means that that `DcmDspDidControlMask` shall correspond in length to the number of controlled signals.

- ▶ `DcmDspRoutineSignalLength` shall be an even multiple of base signal bit size for array types

Description:



- ▶ If `DcmDspRoutineSignalType` is set to `SINT8_N` or `UINT8_N`, then `DcmDspRoutineSignalLength` shall be an even multiple of 8
- ▶ If `DcmDspRoutineSignalType` is set to `SINT16_N` or `UINT16_N`, then `DcmDspRoutineSignalLength` shall be an even multiple of 16
- ▶ If `DcmDspRoutineSignalType` is set to `SINT32_N` or `UINT32_N`, then `DcmDspRoutineSignalLength` shall be an even multiple of 32

Rationale:

According to AUTOSAR 4.3.0 specifications (SWS_Dcm_91041, SWS_Dcm_91043, SWS_Dcm_91045, SWS_Dcm_91047, SWS_Dcm_91049) an array type signal has `DcmDspRoutineParameterSize` elements. Since the Dcm uses AUTOSAR 4.0.3 configuration schema, `DcmDspRoutineSignalLength` is limited in order to maintain the same understanding regarding array type signals size.

- ▶ The protocol currently preempting another protocol cannot itself be preempted

Description:

ProtocolA with higher priority cannot preempt protocolB with lower priority while protocolB is preempting protocol C ProtocolA's request will be rejected even if it has higher priority than the other two protocols.

Rationale:

The AUTOSAR 4.0.3 specification requirement Dcm015 is not applicable while no protocol is running, one is waiting for a preemption to end and the second one is being cancelled as a result of preemption. The AUTOSAR specification does not give more details on this use case.

- ▶ Limited multiplicity of `DcmDslCallbackDCMRequestService`

Description:

According to Dcm SWS of AUTOSAR R4.2.1, the multiplicity of the container `DcmDslCallbackDCMRequestService` is 0 to *. But for the current implementation, the multiplicity is limited from 0 to 255.

Requirements:

`Dcm690_Conf`, `ECUC_Dcm_00690`

Rationale:

Autosar gives no maximum number. The value 255 is expected to be sufficient for most use cases, since in any way, one function call can be used as a dispatcher for multiple condition checks if need to be, as all these calls are executed with the same parameter values.

- ▶ An OBD request cancels a request to switch to a non-default session

Description:



If an OBD request is received while the Server is processing a UDS request for service DiagnosticSessionControl (0x10), and the request is for a non-default session, the Server shall not switch to the non-default session.

Rationale:

To a Tester, this would be indistinguishable from the case in which the OBD request arrives right after the UDS request has completed, and therefore would result in a transition to the default session immediately. The OBD request thus would therefore cancel the transition to the non-default session.

- ▶ Parallel service processing in conjunction with asynchronous service handling is limited

Description:

Protocols of more than one `DcmDslProtocolStackNumber` shall not reference service tables `DcmDslProtocolSIDTable` containing services which have asynchronous service processing enabled, i.e. `DcmAsyncServiceExecution` set to 'true'.

Rationale:

This limitation is necessary because only one single asynchronous service processor instance is available and therefore two services cannot be executed asynchronously in parallel.

- ▶ The functionality related to `TxConfirmationNotification` is not available for `ResponseOnEvent` and `Periodic` transmissions

Description:

The Dcm does not call `Dcm_ApplicationTransmissionConfirmation()` on `ResponseOnEvent` and `Periodic` responses transmission confirmation.

Rationale:

`ResponseOnEvent` and `Periodic` transmissions are not triggered directly by external requests, they are internally generated by the Dcm.

- ▶ External sub-service handlers for service DiagnosticSessionControl (0x10) are not supported

Description:

Configuration of external sub-service handlers for service DiagnosticSessionControl (0x10) is not possible.

Rationale:

This limitation is necessary because the sub-service identifier only represents a session identifier.

- ▶ The handling of multiple "availability InfoTypes" for OBD service Request Vehicle Information 0x09



Description:

According to chapter "8.9.2.2 Request vehicle information response message definition (report supported INFOTYPE)" from ISO 15031-5:2011, states the following: "ECU(s) shall respond to all supported ranges if requested. A range is defined as a block of 32 INFOTYPES(e.g. range #1: INFOTYPE 0x01 to 0x20). The ECU shall not respond to unsupported INFOTYPE ranges unless subsequent ranges have a supported INFOTYPE(s)." In the current implementation, on reception of an OBD Service \$09 request for multiple availability InfoTypes the Dcm shall returns the availability Infotypes that fit into the response buffer.

Rationale:

The integrator has to make sure that the response buffer size determined by the `DcmDslBufferSize` configuration parameter is sufficiently sized to receive the data returned by the provider of the data.

- ▶ Metadata support is limited

Description:

Pdus which require metadata handling by the the Dcm shall have the EcuC configuration parameters `PduId` and `MetaDataTypeRef` configured.

Rationale:

This limitation is necessary because the `PduId` and `MetaDataTypeRef` are needed by the Dcm and EcuC modules to handle the metadata.

- ▶ Length is limited for `SOURCE_ADDRESS_16` and `TARGET_ADDRESS_16`

Description:

`SOURCE_ADDRESS_16` and `TARGET_ADDRESS_16` metadata items lengths for Pdus which are referenced by the Dcm shall be 2 bytes.

Rationale:

This limitation is necessary in order to match SourceAddress data type of `Xxx_Indication()` and `Xxx_Confirmation()` operations of the ServiceRequestNotification interface.

- ▶ The Dcm does not support `ReadScalingDataByIdentifier` (0x24) service request for a Dynamically-defined DID and Range DID

Description:

The Dcm does not support service `ReadScalingDataByIdentifier` (0x24) request for a Dynamically-defined Data Identifier (i.e. a DID whose individual DID configuration `DcmDspDid` references a `DcmDspDidInfo` that has the configuration parameter `DcmDspDidDynamicallyDefined` enabled) and for a Range Data Identifier (i.e. a DID that is configured via configuration parameter `DcmDspDidRange`). Consequently, the Dcm transmits a negative response with an NRC code equal to the value 0x31 (requestOut-



OutOfRange), if the service `ReadScalingDataByIdentifier` (0x24) is requested for any of the previously mentioned DIDs.

Rationale:

The Autosar specifications (4.0.3 and R20-11) as well as the ISO specifications (14229 - 2013 and 2020) does not mention anything on how the Scaling Information shall be retrieved for a Dynamically-defined DID and Range DID.

3.3.1.6. Open-source software

The `Dcm` module does not use open-source software.

3.3.2. Dem module release notes

- ▶ AUTOSAR R4.0 Rev 3
- ▶ AUTOSAR SWS document version: 4.2.0
- ▶ Module version: 6.4.3.B478580
- ▶ Supplier: Elektrobit Automotive GmbH

3.3.2.1. Change log

This chapter lists the changes between different versions.

Module version 6.4.3

2021-12-07

- ▶ Added support for intermediate DTC status storage

Module version 6.4.2

2021-10-27

- ▶ Changed development error ID in J1939 APIs from `DEM_E_PARAM_DATA` to `DEM_E_WRONG_CONFIGURATION`, if the requested Nm node is not configured.
- ▶ Improved Filter mask attributes set by `Dem_J1939DcmSetFreezeFrameFilter()` to be reset during Dem initialization.



- ▶ Implemented stubs for the APIs `Dem_J1939DcmFirstDTCwithLampStatus()`, `Dem_J1939DcmGetNextDTCwithLampStatus()` and `Dem_J1939DcmGetNextSPNInFreezeFrame()`, according to AUTOSAR R4.2.2
- ▶ Added debug macro in internal function `Dem_GetNumberJ1939DTCs()` and improved documentation of API `Dem_J1939DcmGetNumberOfFilteredDTC()`
- ▶ Improved development error handling if external data elements cannot be read: moved error reporting from Dem internal API to `Dem_MainFunction()`, `Dem_SetEventStatus()` and `Dem_PrestoreFreezeFrame()`
- ▶ ASCDEM-4568 Fixed known issue: `Dem_J1939DcmSetDTCFilter()` rejects certain J1939 DTC filter criteria if OBD is disabled and DET is enabled
- ▶ Changed invocation of J1939 callback `J1939Dcm_DemTriggerOnDTCStatus()` to outside of critical section
- ▶ Added configuration constraint if the total number of DIDs contained in a freeze frame record, considering both the common part as well as the event specific part, exceeds 255
- ▶ Updated Dem internal J1939 event configuration to reduce ROM consumption
- ▶ ASCDEM-4678 Fixed known issue: Memory sections conflict between declaration and definition
- ▶ Updated Dem internal J1939 freeze frame configuration to reduce ROM consumption
- ▶ ASCDEM-4697 Fixed known issue: Permanent faults are not stored in NVRAM at the end of ignition cycle but only at the subsequent DCY start under certain circumstances

Module version 6.4.1

2021-06-25

- ▶ Updated the generated configuration data structure to support larger Dem configurations, when event combination is used
- ▶ Updated internal function to improve runtime performance by checking MIL OnOff status
- ▶ ASCDEM-4374 Fixed known issue: Dem sets the UDS status bits CDTC and TFSLC for an aged event under certain circumstances
- ▶ ASCDEM-4407 Fixed known issue: The Dem module wrongly sets the TF, TFTOC, PDTC, and TFSLC bits during event memory entry storage
- ▶ ASCDEM-4415 Fixed known issue: Dem sets the CDTC bit after aging under certain circumstances
- ▶ ASCDEM-4426 Fixed known issue: Concurrent write access to a NvM block leads to a failed DTC clearing operation
- ▶ Added an optional configuration parameter `DemStatusBitStorageTestFailedPerEvent` to provide a more granular control of the test failed status bit storage.



- ▶ ASCDEM-4406 Fixed known issue: Dem might not compile due to the wrong memory mapping of an internal variable
- ▶ Improved event clearing algorithm to clear combined events within a minimal number of Dem_MainFunction() cycles
 - ▶ The number of required cycles to clear combined events is depending on the value of configuration parameter DemMaxNumberClearEventsPerCycle.
- ▶ ASCDEM-4502 Fixed known issue: EventDataChanged callback not called when only the J1939 freeze frame is stored/updated
- ▶ ASCDEM-4492 Fixed known issue: The API Dem_SetWIRStatus() does not accept the WIR status value FALSE consecutive to an ECU startup

Module version 6.4.0

2021-03-05

- ▶ Improved Dem_J1939DcmClearDTC() API
- ▶ ASCDEM-4315 Fixed known issue: Dem fills unavailable J1939 freeze frame data with padding value 0x00 instead of 0xFF
- ▶ ASCDEM-4322 Fixed known issue: PID\$21 is reset to zero even when the MIL status does not change from deactivated to activated
- ▶ Updated the internal functions to improve runtime performance when checking whether a newly reported event should displace an existing event memory entry
- ▶ Updated the arrangement of configuration parameters within Tresos Studio GUI to improve user experience
- ▶ ASCDEM-4284 Fixed known issue: Dem J1939 read functions also consider permanent DTCs that are only stored volatile
- ▶ Updated the memory mapping keywords used by the Dem service interfaces, when RTE is used
- ▶ Updated NvM header inclusion and access to NvM symbolic name values
- ▶ Updated the internal functions to improve runtime performance when checking the status of an indicator
- ▶ Updated header inclusion scheme
 - ▶ If a DemNvRamBlockId is configured in Dem, then NvM must be configured to include Dem.h.
- ▶ Added configuration constraint to restrict J1939DTCs to primary memory only

Module version 6.3.4

2021-02-12



- ▶ Implemented stubs for the APIs `Dem_DcmGetAvailableOBDMIDs()`, `Dem_DcmGetNumTIDsOfOBDMID()` and `Dem_DcmGetDTRData()` from AUTOSAR R4.3.1
- ▶ Improved documentation for `DemCallbackEventStatusChanged` and `DemCallbackEventDataChanged`
- ▶ Updated the stubs for the Dem APIs used to read infotype \$08 and \$0B, according to AUTOSAR R4.3.1
- ▶ Implemented stub for the API `Dem_DcmReadDataOfPID91()` from AUTOSAR R4.3.1
- ▶ ASCDEM-4163 Fixed known issue: Following a ClearDTC, a permanent DTC is erased before the permanent fault code cycle condition is satisfied
- ▶ Updated validation checks for configuration parameters
- ▶ ASCDEM-4290 Fixed known issue: Completion status of a disabled readiness group is wrongly shown as completed in PID\$41

Module version 6.3.3

2021-01-22

- ▶ Removed definition of obsolete data types
- ▶ ASCDEM-4105 Fixed known issue: J1939 lamp flash status is returned wrongly if the lamp is available and OFF
- ▶ Corrected the debug macro name used in `Dem_J1939DcmReadDiagnosticReadiness2()`

Module version 6.3.2

2020-12-18

- ▶ ASCDEM-4221 Fixed known issue: Dem module generates a BSWMD file with non-unique short-names for packages
- ▶ Updated configuration check to consider that `DemDataElementClass` can also be referenced by `DemSPNDataElementClassRef`
- ▶ ASCDEM-4217 Fixed known issue: A DCY event might not be confirmed at DCY qualification under certain conditions
- ▶ Improved description of Dem APIs and configuration parameters

Module version 6.3.1

2020-11-20

- ▶ Optimized the validation check regarding `DemUserDefMemoryId`



- ▶ Updated default value for the unsupported configuration parameter `DemOccurrenceCounterProcessing` to align with the implemented behavior
- ▶ ASCDEM-4153 Fixed known issue: No `EventStatusChanged` notification if `DEM_ONLY_THIS_CYCLE_AND_READINESS` status bits are reset

Module version 6.3.0

2020-11-11

- ▶ Updated support for NvM explicit synchronization block callback calculation in service needs
- ▶ ASCDEM-4125 Fixed known issue: The APIs `Dem_GetNextFreezeFrameData()` and `Dem_GetNextExtendedDataRecord()` silently skip records that cannot fit within the provided buffer
- ▶ ASCDEM-4122 Fixed known issue: Under certain conditions, the API `Dem_GetNextExtendedDataRecord()` does not return `DEM_NO SUCH ELEMENT` after all relevant records are returned
- ▶ Removed duplicated template file `Dem_Customization.h` from `generate\include\` folder
- ▶ Updated Dem to provide the C/S interface `PowerTakeOff` and the corresponding port only if OBD support is enabled
- ▶ Corrected missing APIs and hyperlinks for module reference
- ▶ ASCDEM-4031 Fixed known issue: A DCY event can be confirmed without qualification of the DCY under certain conditions
- ▶ ASCDEM-4196 Fixed known issue: Reset of OCC5 counter and setting of `SI30.DEM_SI30_STATUS_WIR_SLC` bit are not stored immediately
- ▶ Optimized run-time performance for NvM write trigger by reducing critical sections calls
- ▶ ASCDEM-4098 Fixed known issue: J1939 freeze frame is not stored correctly in all cases
- ▶ Added clear limit for `ClearThisCycleAndReadinessStatus`
- ▶ Removed all SERVER-CALL-POINTS from runnable entity `CddIfClearDTC` due to asynchronous ClearDTC processing
- ▶ ASCDEM-4013 Fixed known issue: Condition for erasing permanent DTCs after `ClearDTC` is stricter than required
 - ▶ Migration advice: With this change, the structure of the data stored in the NvM block for permanent memory is changed. Therefore, it is recommended to manually erase the NvM block while doing the SW update, if it is not configured to be handled automatically (i.e., if `DemNvDataConfigSignatureUsed == false` or `DemNvDataConfigSignatureCalcSet != DEM_NVM_SIGNATURE_ALL`).
- ▶ Refined descriptions of Dem calibration data in `Dem_Bswmd.arxml` file
- ▶ ASCDEM-4149 Fixed known issue: Configuration parameter `DemJ1939DTCValue` is not named according to AUTOSAR 4.2.2 in the schema



- ▶ The configuration parameter `DemJ1939DTC` is updated to `DemJ1939DTCValue`. During integration, the parameter name will have to be updated.
- ▶ Updated the AUTOSAR R4.3.1 compliant Dem APIs used by Dcm from prototype to mass production quality

Module version 6.2.0

2020-09-25

- ▶ Optimized out not required `NvM_WriteAll` attribute for `DEM_NVM_BLOCK_ID_PERMANENT`
- ▶ Introduced discarding of event memory entries during `Dem_Init()` when `NvM_SetDataIndex()` fails
- ▶ Improved usage of `DBG` macros
- ▶ Added `REQUIRES-INDEX` attribute at sorted lists in configuration
- ▶ Optimized ROM consumption for event failure class index
- ▶ Updated category `BITFIELD_TEXTTABLE` for CompuMethods `EventStatusExtendedType` and `DebouncingStateType`
- ▶ Improved handling of negative return values from `NvM` interfaces and inconsistent `NvM` data
- ▶ Limited the number of processed events per `Dem_MainFunction()` cycle for `ClearDTC`
- ▶ ASCDEM-4078 Fixed known issue: The Dem does not call the SW-C API <Module>`_SetClearDTC` for `ClearDTC` behavior `DEM_CLRRESP_NONVOLATILE_FINISH`

Module version 6.1.1

2020-08-28

- ▶ ASCDEM-3890 Fixed known issue: The Nv storage or clearing of an event memory entry can be omitted under specific race conditions if no `Dem_Shutdown` is executed
- ▶ Updated qualification concept for DCY and corresponding CDTC and WIR/MIL handling
- ▶ ASCDEM-4294 Fixed known issue: DCY event is not confirmed if no fault confirmation is configured and `DemResetConfirmedBitOnOverflow` is set to true
- ▶ ASCDEM-3922 Fixed known issue: A compiler error occurs when permanent storage with an `NvM` with conditional legacy symbolic names support is used, and no immediate storage/clear is activated
- ▶ Implemented the configuration parameter `DemMaxHandledInitMonitorReenabledPerScheduling` that limits the number of `DemInitMonitorForEvent` callbacks triggered in each call of `Dem_MainFunction()`
- ▶ ASCDEM-3991 Fixed known issue: OCC counters and SI30 are not stored immediately
- ▶ ASCDEM-3993 Fixed known issue: CSLF and CSFF are not restored as immediate data



- ▶ ASCDEM-3877 Fixed known issue: DTC status of combined events is not updated correctly in `Dem_SetAgingCycleState()` under certain conditions
- ▶ Changed optional configuration references of events to disabled by default

Module version 6.1.0

2020-07-20

- ▶ ASCDEM-3959 Fixed known issue: `Dem_ClearDTC` may wrongly return `DEM_CLEAR_BUSY` or the clearing is executed multiple times
- ▶ Extended J1939 support to work independently from OBD support

Module version 6.0.0

2020-06-26

- ▶ Changed all `NO_INIT` memory sections to `CLEARED`
- ▶ Added J1939 support for Dem
- ▶ Added support for parallel processing of OBD and UDS protocols by updating the Dem-Dcm APIs according to ASR R431; relevant changes are:
 - ▶ Added possibility to configure diagnostic clients via `DemClientId`
 - ▶ Feature *OBD powertrain data* updated the APIs `Dem_DcmGetDTCOfOBDFreezeFrame()` and `Dem_DcmReadDataOfOBDFreezeFrame()`, also to support UDS service `ReadDTCInformation (0x19)` sub-function `0x05`
 - ▶ Feature *DTC filtering* updated the APIs `Dem_SetDTCFilter()`, `Dem_GetNumberOfFilteredDTC()`, `Dem_GetNextFilteredDTC`, `Dem_GetNextFilteredDTCAndFDC()`, `Dem_GetNextFilteredDTCAndSeverity()`, `Dem_GetTranslationType()` and `Dem_GetDTCStatusAvailabilityMask()`
 - ▶ Feature *DTC selection mechanism* introduced the new APIs `Dem_SelectDTC()` and `Dem_DTCSelectionResult()`
 - ▶ Feature *control DTC setting* updated the APIs `Dem_DisableDTCSetting()` and `Dem_EnableDTCSetting()`
 - ▶ Feature *DTC record update* updated the APIs `Dem_DisableDTCRecordUpdate()` and `Dem_EnableDTCRecordUpdate()`
 - ▶ Feature *DTC status* updated the API `Dem_GetStatusOfDTC()`
 - ▶ In prototype quality, the feature *DTC attribute* updated the APIs `Dem_GetSeverityOfDTC()`, `Dem_GetFunctionalUnitOfDTC()` and `Dem_GetDTCByOccurrenceTime()`
 - ▶ In prototype quality, the feature *clear DTC* updated the API `Dem_ClearDTC()` and introduced the new API `Dem_DTCSelectionResultForClearDTC()`



- ▶ In prototype quality, the feature *read event related data* updated the APIs `Dem_GetSizeOfExtendedDataRecordSelection()`, `Dem_GetSizeOfFreezeFrameSelection()`, `Dem_GetNextExtendedDataRecord()` and `Dem_GetNextFreezeFrameData()`, and introduced the new APIs `Dem_SelectExtendedDataRecord()` and `Dem_SelectFreezeFrameData()`
- ▶ In prototype quality, the feature *stored DTC filter* updated the APIs `Dem_DcmSetStoredDTCFilter()` and `Dem_DcmGetNextFilteredStoredDTC()`
- ▶ In prototype quality, the feature *freeze frame record filter* updated the APIs `Dem_SetFreezeFrameRecordFilter()` and `Dem_GetNextFilteredRecord()`
- ▶ In prototype quality, the feature *user-defined memory identifier* updated the type `Dem_DTCOriginType` impacting the input parameter `DTCOrigin` of the APIs `Dem_DcmSetStoredDTCFilter()`, `Dem_SetDTCFilter()`, `Dem_SelectDTC()`, `Dem_GetEventMemoryOverflow()`, `Dem_GetNumberOfEventMemoryEntries()` and the operation `ClearDTC` of the C/S-interface `CddIf`.

The configured user defined memory Id is internally mapped to the secondary memory

Additional notes:

- ▶ All clients have access to all diagnostic data (no support for `DemEventMemorySet`)
- ▶ The same client shall not call multiple Dem-Dcm APIs in parallel
- ▶ `Dem_SelectDTC()` and its dependent APIs do not return `DEM_BUSY`
- ▶ The following APIs have their functionality restricted to one client at a time: `Dem_DisableDTCSetting()`, `Dem_EnableDTCSetting()`, `Dem_DisableDTCRecordUpdate()`, `Dem_EnableDTCRecordUpdate()`, `Dem_DcmSetStoredDTCFilter()`, `Dem_DcmGetNextFilteredStoredDTC()`, `Dem_SetFreezeFrameRecordFilter()`, `Dem_GetNextFilteredRecord()`
- ▶ While clearing is being processed for one client, a new clear request from a different client will return `DEM_CLEAR_BUSY`
- ▶ `Dem_GetDTCSelectionResult()` returns `DEM_WRONG_DTC` for a selection of a single OBD DTC
- ▶ The service Id of APIs `Dem_DcmSetStoredDTCFilter()` and `Dem_DcmGetNextFilteredStoredDTC()` is changed from `0x90`, respectively `0x91` to `0xD0`, respectively `0xD1`
- ▶ Implemented asynchronous triggering of `DemInitMonitorForEvent` callbacks when `ControlDTCSetting` is re-enabled by triggering at most 50 configured callbacks in each call of `Dem_MainFunction()`
- ▶ Implemented asynchronous processing of `Dem_ClearDTC()` requests in `Dem_MainFunction()` context

Module version 5.20.0

2020-05-22

- ▶ ASCDEM-3795 Fixed known issue: NvM blocks are not generated by the Calculate Service Needs wizard
- ▶ Added Calibration support for Dem time-based debouncing



- ▶ Updated calibration interface of counter based debouncing: removed unused `DemDebounceCounter-Based` and optimized size of `Dem_DebounceCounterClassIdx`

Module version 5.19.9

2020-04-24

- ▶ ASCDEM-3751 Fixed known issue: `Dem_GetNextDTCAndSeverity()` does not consider `DemDtcStatusAvailabilityMask` when reporting DTCStatus
- ▶ Improved calibration short names for OBD DTC with combination
- ▶ Extended calculation of `DemNvMConfigSignature`
- ▶ ASCDEM-3357 Fixed known issue: DTC record update cannot ensure event entry data consistency for `ReadDTCInformation`

Module version 5.19.8

2020-03-25

- ▶ Added support for replacement DTCs via Multi-Event Triggering limited to Event Reports

Module version 5.19.6

2020-02-21

- ▶ ASCDEM-3566 Fixed known issue: Callback functions `Dem_NvMWriteFinishedPermanentMemory()`, `Dem_NvMWriteCopyPermanentMemory()` and `Dem_NvMReadCopyPermanentMemory()` are not available, if immediate storage is disabled
- ▶ Added dedicated memory sections for calibratable data introduced with Event Combination with OBD
- ▶ Added configuration switch for `Dem_ReplUMPR*` functions
- ▶ ASCDEM-3676 Fixed known issue: Miscalculation of total freeze frame sizes used for validation in generator
- ▶ ASCDEM-3306 Fixed known issue: `Dem_ResetEventDebounceStatus()` API causes segmentation fault by calling for events with `DemDebounceMonitorInternal`
- ▶ ASCDEM-3680 Fixed known issue: `Dem_ClearDTC()` gets blocked by `NvM_WriteBlock()` that is triggered sporadically by event reports
- ▶ Note that module version 5.19.5 is skipped and used for a special extra delivery

Module version 5.19.4

2019-10-11



- ▶ Support for Event Combination together with OBD
- ▶ ASCDEM-3452 Fixed known issue: Dem_DcmReadDataOfPID01() does not report confirmed OBD DTCs that are not assigned to a readiness group
- ▶ ASCDEM-3441 Fixed known issue: OCC6 is wrongly increased more than once if a BSW event is reported as FAILED multiple times in the same operation cycle
- ▶ Dem-callouts moved into designated memory sections
- ▶ Removed warning generated by DemIndicatorFailureCycleRef even when the parameter is disabled, or not editable

Module version 5.19.3

2019-09-06

- ▶ ASCDEM-3342 Fixed known issue: Cleared DTCs could reappear when entering DEM_OPCYC_OBD_-DCY
- ▶ ASCDEM-3380 Fixed known issue: Wrong value for 'DemInternalDataElement' set to 'DEM_EVENT_ID' for combined events
- ▶ Added dedicated memory sections for calibratable data

Module version 5.19.2

2019-08-09

- ▶ ASCDEM-3295 Fixed known issue: Using the DTC value '0' when calling an API for the first time might lead to out-of-bounds access. Root cause already fixed with ASCDEM-3264

Module version 5.19.1

2019-07-12

- ▶ Corrected behavior where update of UDS status byte was not protected against concurrent access that may lead to inconsistent status
- ▶ Removed nested critical section for permanent memory processing
- ▶ Improved handling and documentation of consecutive calls of Dem_Init() and Dem_PreInit()
- ▶ ASCDEM-3279 Fixed known issue: ConfirmedDTC bit is wrongly reset on the next Dem_MainFunction() for the Combined Event DTC status

Module version 5.19.0

2019-06-14



- ▶ ASCDEM-2910 Fixed known issue: Dem might calculate a wrong UDS status if event processing is interrupted by Dem_ClearDTC or Dem_ResetEventStatus
- ▶ Optimized MemMap mapping of pointer arrays
- ▶ ASCDEM-3081 Fixed known issue: Dem does not generate A2L files
- ▶ Optimized calibratable data. A2L files should be regenerated regardless of configuration has changed or not
- ▶ Support for Multiple OBD freeze frame: Core Functionality (for record number 0x00)
- ▶ ASCDEM-3270 Fixed known issue: A compiler error occurs when RTE is used, calibration is enabled and no indicators are configured
- ▶ ASCDEM-3264 Fixed known issue: Wrong handling for DTC value 0 for several APIs
- ▶ ASCDEM-3284 Fixed known issue: Dem wrongly reports DET errors in case of ReportErrorStatus from satellite core

Module version 5.18.16

2019-05-17

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

Module version 5.18.15

2019-04-18

- ▶ Support for BSW event reporting from multiple cores - protect APIs for incorrect accessing
- ▶ Optimized EventClass configuration description table
- ▶ ASCDEM-3183 Fixed known issue: False warning on unused DataElementClass in a specific case

Module version 5.18.14

2019-04-11

- ▶ Implemented support for ASR 4.3 compatibility for selected interfaces

Module version 5.18.13

2019-03-22

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

**Module version 5.18.12**

2019-03-21

- ▶ ASCDEM-3144 Fixed known issue: Incorrect handling of OBD Freeze Frame in specific race conditions
- ▶ Implemented support for dynamic event availability via API

Module version 5.18.11

2019-02-20

- ▶ ASCDEM-3133 Fixed known issue: Callout function declarations of DemCallbackMemStackMainFuncTrigger and DemCalloutDynamicDTCFnc are not inside a CODE section
- ▶ ASCDEM-3134 Fixed known issue: Wrong compiler abstraction used for Read DataElement operations of the SenderReceiver interface
- ▶ Optimized implementation of Dem internal search function for the case when no event memory entry exists
- ▶ Implemented support for reset of Dem NvM data after incompatible configuration update
- ▶ Optimized implementation of Dem_SetOperationCycleState() - Init Monitor Callbacks reworked
- ▶ ASCDEM-3079 Fixed known issue: Confirmed emission-related BSW events are not stored immediately in permanent memory
- ▶ Clarified integration code hint for configuration parameter DemCallbackMemStackMainFuncTrigger
- ▶ Optimized implementation of Dem_SetOperationCycleState() - Number of critical section activations reduced
- ▶ Optimized implementation of Dem_SetOperationCycleState() - Disable via configuration parameter DemCallbackEventStatusChangedOpCycStartSupport

Module version 5.18.10

2019-01-23

- ▶ ASCDEM-2924 Fixed known issue: Memory cannot be cleared after system reinitialization
- ▶ ASCDEM-2931 Fixed known issue: Possible null pointer access or wrong data retrieved if no events have aging configured
- ▶ ASCDEM-3083 Fixed known issue: Warning when import Dem_swc_internal.arxml file
- ▶ Extended event displacement algorithm with criteria for OBD and TNCTOC
- ▶ ASCDEM-2678 Fixed known issue: OBD freeze frame might store incorrect data for large configurations
- ▶ ASCDEM-3072 Fixed known issue: OCC counters are not stored at Dem_Shutdown
- ▶ ASCDEM-2589 Fixed known issue: No NvM storage of event-related data on update of OCC5 value occurs if immediate storage is enabled under particular conditions



- ▶ ASCDEM-3109 Fixed known issue: Dem_DTCFilter: Memory section conflict between declaration and definition
- ▶ ASCDEM-3108 Fixed known issue: Dem module does not generate if queue length for DemEventInfoPort operations is calculated by RTE

Module version 5.18.9

2018-11-23

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

Module version 5.18.8

2018-10-26

- ▶ Refined the mapping in the memory of variables and constants for optimized memory consumption
- ▶ ASCDEM-2677 Fixed known issue: Dem does not compile if DemAgingAllowedSeperateFlag has a wrong value
- ▶ Support for DataService interface with EventID

Module version 5.18.7

2018-10-09

- ▶ Support for BSW event reporting from multiple cores
- ▶ ASCDEM-2492 Fixed known issue: Dem module does not fully support BSW Distribution for multi-core systems

Module version 5.18.6

2018-09-28

- ▶ Support identical OBD DTC feature for ACG-8

Module version 5.18.5

2018-08-24

- ▶ Changed signatures of the DemCallbackMemStackMainFuncTrigger and DemCalloutDynamicDTCFnc callout functions by removal of the const qualifier of the parameters

**Module version 5.18.4**

2018-07-27

- ▶ ASCDEM-2896 Fixed known issue: Dem Module does not compile if SenderReceiver interface is enabled and when no DemExtendedDataRecordClass or CS DataElements are configured
- ▶ ASCDEM-2886 Fixed known issue: Dem might return wrong data when Operation Cycles non-referenced from any DemOperationCycleRef or unused Aging Cycles are configured

Module version 5.18.3

2018-06-22

- ▶ Added support for Dem_SetDTCSuppression() API according to AUTOSAR 4.3.0
- ▶ ASCDEM-2749 Fixed known issue: Dem module wrongly reports status change notifications
- ▶ Extended list of supported PIDs with the OBD-specific Dcm <-> Dem services with *Internal calculation of PID \$21 handling - distance traveled while MIL is activated*

Module version 5.18.2

2018-05-25

- ▶ ASCDEM-2743 Fixed known issue: Dem_SetWIRStatus does not control the WIR-bit of ECU internal events
- ▶ Enhance checks and description of side allocation feature (config parameter DemCalloutDynamicDTCFnc)
- ▶ Implemented support for project specific Dem_ReportErrorStatus implementation

Module version 5.18.1

2018-04-20

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

Module version 5.18.0

2018-03-16

- ▶ Implemented missing memory sections in the Dem software component description causing incompliable code

Module version 5.17.29

2018-02-28



- ▶ Optimized code to eliminate dead assignments
- ▶ Optimized code for ordered DTC reporting
- ▶ ASCDEM-2675 Fixed known issue: Dem_GetSizeOfFreezeFrameByDTC() might return wrong value for large development freeze frames
- ▶ Implement support for SenderReceiver interface
- ▶ Changed Primitive Implementation Data Types to Redefinition Implementation Data Types for unspecified Implementation Data Types
- ▶ Implemented compliance to MISRA-C:2012

Module version 5.17.28

2017-12-23

- ▶ Implemented new configuration option for configurable queue sizes for Dem client server operations
- ▶ ASCDEM-2636 Fixed known issue: Dem_DcmCancelOperation() cancels ClearDTC requests of another user
- ▶ ASCDEM-2635 Fixed known issue: Dem_DcmCancelOperation() might not cancel a ClearDTC request
- ▶ Implement Reset/freeze of event debouncing via Dem_ResetEventDebounceStatus() API
- ▶ ASCDEM-2638 Fixed known issue: The DTC clearing process is not locked for other users in case of an NvM memory write error
- ▶ Implemented optimization for long Dem code generation time
- ▶ Support for Reset/freeze of event debouncing via non-fulfilled enable conditions and disabled ControlDTCSetting
- ▶ Support for interface DiagnosticInfo according to ASR 4.2

Module version 5.17.27

2017-10-13

- ▶ ASCDEM-2587 Fixed known issue: Data structures for DemExternalCSDataElement are not correctly generated for specific configuration
- ▶ ASCDEM-2581 Fixed known issue: Incomplete syntax in configuration check for DemFreezeFrameClass-Ref causes a generation error
- ▶ Optimized speed for configuration verifier checks
- ▶ Support for Operation Cycle Counters 'cycle since first failed' and 'cycle since first last failed' according to R4.2.1
- ▶ Implemented support for Dem_ClearDTC processing burst reduction



- ▶ Removed AUTOSAR 3.x compliant symbolic name value macros and updated the logic to only provide AUTOSAR 4.0.2 compliant macros if macro DEM_PROVIDE_LEGACY_SYMBOLIC_NAMES is defined

Module version 5.17.26

2017-07-24

- ▶ Implemented R4.2.1 clarification for EventDataChanged triggering-mechanism

Module version 5.17.25

2017-07-13

- ▶ Implemented support for healing completion based aging
- ▶ ASCDEM-2574 Fixed known issue: Incorrect handling of event combination data array causes out of bounds access
- ▶ ASCDEM-2454 Fixed known issue: Warning indicator calibration for BSW events does not work properly

Module version 5.17.24

2017-06-26

- ▶ Implemented support for immediate storage of internal events
- ▶ Implemented support for considering untested cycles for aging behavior
- ▶ Corrected misleading DET reports in store functionality
- ▶ ASCDEM-2506 Fixed known issue: Data structures for enable condition are not correctly generated for specific configuration
- ▶ ASCDEM-2491 Fixed known issue: Combined DTC status byte is not updated when the DTC status of combined event is modified

Module version 5.17.23

2017-04-28

- ▶ ASCDEM-2484 Fixed known issue: Generation error caused by missing invalidation for DemEvtCmbCommonParamMaster
- ▶ ASCDEM-2463 Fixed known issue: Absent filter handling for DTCStatusAvailabilityMask may impact reported DTCs by status mask for ReadDTCInformation (0x19) service
- ▶ ASCDEM-2482 Fixed known issue: Linker errors if no DemIndicatorAttribute exists for any event

**Module version 5.17.22**

2017-03-21

- ▶ ASCDEM-2387 Fixed known issue: Calibration of aging allowed flag can lead to out of bound access
- ▶ ASCDEM-2399 Fixed known issue: Compile error when legacy symbolic names are disabled via macro DEM_DONT_PROVIDE_LEGACY_SYMBOLIC_NAMES
- ▶ ASCDEM-2336 Fixed known issue: Warning indicator processing may be skipped for aged BSW events
- ▶ ASCDEM-2337 Fixed known issue: Warning indicator processing may be skipped for BSW events which are displaced
- ▶ ASCDEM-2372 Fixed known issue: StatusChanged notifications missing in API Dem_ResetReadiness()
- ▶ DTC significance as Internal Data Element with output as extended data
- ▶ ASCDEM-2410 Fixed known issue: Possible overflow for aging cycle symbolic name values
- ▶ Event id as internal data element with output as extended data
- ▶ ASCDEM-2430 Fixed known issue: Code generator does not correctly handle DemCallbackClearEventAllowed parameter
- ▶ ASCDEM-2427 Fixed known issue: Linker error: undefined reference to Dem_UpdateAgingCounter
- ▶ ASCDEM-2423 Fixed known issue: BSW Events: OCC7 counter is incremented more than once per operation cycle
- ▶ ASCDEM-2424 Fixed known issue: BSW Events: Fault confirmation counter is incremented more than once per operation cycle
- ▶ ASCDEM-2394 Fixed known issue: OBD freeze frame is not cleared in specific cases
- ▶ ASCDEM-2407 Fixed known issue: If DTC record update is disabled and event is aged, combined DTC status is not updated in case of ECU sleep and wakeup
- ▶ ASCDEM-2413 Fixed known issue: An event with immediate storage ON may not be stored immediately
- ▶ Corrected invalid XPath expression due to invalid DemMILIndicatorRef
- ▶ User controlled WarningIndicatorRequested-bit
- ▶ ASCDEM-2451 Fixed known issue: Synchronous server call points for FreezeFrameData interfaces are not generated in the Dem_MainFunction runnable
- ▶ ASCDEM-1426 Fixed known issue: Invalid OCC2 increment for multiple passed BSW events in one Dem_MainFunction cycle
- ▶ ASCDEM-2436 Fixed known issue: DemMaxNumberEventEntry[Permanent|Primary|Secondary|Mirror] configured to 255 may lead to an endless loop

Module version 5.17.21

2016-12-16



- ▶ Provided Handle-Id Wizards for all platforms

Module version 5.17.20

2016-11-04

- ▶ Implemented support for Configuration Signature
- ▶ ASCDEM-2240 Fixed known issue: Wrong calculation of freeze frame size if calibration is supported
- ▶ Implemented support of trigger on TestFailed for DemExtendedDataCapture and DemFreezeFrameCapture
- ▶ ASCDEM-2327 Fixed known issue: Code generator fails with combined events, common freeze frame enabled, and configured freeze frame record numbers
- ▶ Implemented support for API Dem_GetDTCByOccurrenceTime()
- ▶ ASCDEM-2331 Fixed known issue: XPath-expression error if combined slave event uses the MIL
- ▶ ASCDEM-2346 Fixed known issue: Compiler error occurs if event combination is enabled and fault confirmation is not used
- ▶ Implemented optional support for Monitor Re-initialization callback DemInitMonitorForEvent with InitMonitorReason DEM_INIT_MONITOR_REENABLED.
- ▶ ASCDEM-2340 Fixed known issue: Readiness group cannot be calibrated
- ▶ ASCDEM-2339 Fixed known issue: Indicator of combined events with DemEvtCmbCommonParamMaster = false cannot use events failure cycle and threshold
- ▶ ASCDEM-2342 Fixed known issue: ConfirmedDTC bit might not be set for events assigned to OBD driving cycle
- ▶ ASCDEM-2373 Fixed known issue: Permanent event memory entry is only removed if all assigned indicators are healed
- ▶ Implemented Configurable clear event allowed behavior.
- ▶ ASCDEM-2378 Fixed known issue: Incorrect reset of warning indicator failure counter

Module version 5.17.0

2016-07-22

- ▶ ASCDEM-2241 Fixed known issue: Dem compilation fail due to missing MIL reference
- ▶ Updated Dem_GetSizeOfExtendedDataRecordByDTC() API, in case DemGeneral/DemOBDSupport is disabled, to handle the DTCExtDataRecordNumber of 0xFE with DEM_GET_SIZEOFEDRBYDTC_W_RNUM NRC.
- ▶ ASCDEM-2247 Fixed known issue: OBD extended data records are wrongfully checked against 90U (decimal value)



- ▶ ASCDEM-2258 Fixed known issue: Dem_GetSizeOfExtendedDataRecordByDTC() API exits with incorrect return value
- ▶ Updated Fault Confirmation and WIR bit setting behavior:
 - ▶ *Updated Fault Confirmation handling for setting the ConfirmedDTC status bit according to the number of TestFailedThisOperationCycle transition configured via DemEventFailureCycleCounterThreshold (refer to RfC #70785).*
 - ▶ *Updated Warning Indicator handling for switching on the Warning Indicator status and setting the WarningIndicatorRequested status bit according to the number of TestFailedThisOperationCycle transition configured via DemIndicatorFailureCycleCounterThreshold (refer to RfC #71313).*
 - ▶ *Multi-indicator behavior has been updated for the Warning Indicator ON status transition. All indicators assigned to the same event are handled uniformly with the new threshold behavior.*
- ▶ ASCDEM-2249 Fixed known issue: Parameter 'Common part of Development Freeze Frame A Class' could be accessed with invalid reference

Module version 5.16.6

2016-06-15

- ▶ Implemented support for Event Combination - Combination On Storage
- ▶ ASCDEM-2229 Fixed known issue: File Dem_Bswmd.arxml has no unique short name for Dem_DbounceCounterClassIdx
- ▶ ASCDEM-2230 Fixed known issue: File Dem_Bswmd.arxml has wrong type reference for Dem_DbounceCounterClassIdx
- ▶ Implemented separate memory section SEC_VAR_SAVED_ZONE for non-volatile data variables. New memory sections will have to be set accordingly during integration(e.g. map it along with SEC_VAR_NOINIT)

Module version 5.16.5

2016-05-25

- ▶ ASCDEM-2197 Fixed known issue: Dem does not compile when legacy symbolic names are disabled via macro DEM_DONT_PROVIDE_LEGACY_SYMBOLIC_NAMES
- ▶ Implemented support for additional internal data element priority
- ▶ ASCDEM-2198 Fixed known issue: Dem_ReportErrorStatus() may report an unjustified Det error during pre-initialization phase
- ▶ Implemented support for additional internal data element "Aging Down Counter"

Module version 5.16.4

2016-04-29



- ▶ Implemented support for reading the number of event memory entries in API `Dem_GetNumberOfEventMemoryEntries`
- ▶ Implemented internal source code improvements
- ▶ Implemented support for 0x19-04 positive response with calibrated unused freeze frame class
- ▶ ASCDEM-2175 Fixed known issue: Service 0x19 03 loses development and OBD freeze frame records in `Dem_GetNextFilteredRecord`
- ▶ Added configuration check to limit the usage of `FDC_THRESHOLD` for `DemFreezeFrameCapture` and `DemExtendedDataCapture`
- ▶ Implemented support for clearing OBD freeze frame when event gets aged or if PDTC bit is cleared before CDTC bit was set

Module version 5.16.3

2016-04-01

- ▶ ASCDEM-2090 Fixed known issue: `Dem_DcmReadDataOfPID41()` may return wrong value if it is interrupted by `Dem_SetOperationCycleState()` or `Dem_SetEventDisabled()`

Module version 5.16.2

2016-03-04

- ▶ ASCDEM-2106 Fixed known issue: `Dem_DcmReadDataOfPID31` may return wrong value if it is interrupted by `Dem_MainFunction` or `ClearDTC`
- ▶ Implemented calibration of event availability and event storage handling for DTC value zero
- ▶ Improved event memory handling in `Dem_MainFunction()` regarding run-time and critical section length
- ▶ ASCDEM-2084 Fixed known issue: Data of Dem permanent event memory cannot be restored at startup
- ▶ Implemented freeze frame "online" calibration (FF class and DIDs in FF)

Module version 5.16.1

2016-02-15

- ▶ Implemented ClearDTC for disabled events and events without DTCs
- ▶ ASCDEM-2060 Fixed known issue: Debounce class is not generated correctly without license `DIAG_VCC` or `EB_DIAG_FP1`
- ▶ Implemented support common freeze frame for events without individual freeze frame



- ▶ ASCDEM-2074 Fixed known issue: DemCallbackMemStackMainFuncTrigger is not editable if DemClearDTCBehavior == NONVOLATILE and immediate storage is disabled
- ▶ Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeader-File`
- ▶ Added development freeze frame replacement when confirmed event is aged
- ▶ ASCDEM-2092 Fixed known issue: Event/DTCStatusChanged callback not called for PDTC clear in case of aged event replacement with `DEM_TRIGGER_ON_FDC_THRESHOLD`

Module version 5.16.0

2016-01-15

- ▶ Provision of ExtendedDataRecords with always available data via 0x19 also if event is NOT stored
- ▶ Implemented support for PID1C in API `Dem_DcmReadDataOfPID1C`
- ▶ Added support for convenient calibration of aging allowed
- ▶ Optimized enable condition data structures and processing
- ▶ Implemented calibration support for `DemEnableConditionGroupRef`
- ▶ ASCDEM-1982 Fixed known issue: Initialization of indicators write out of bound in case of calibration support is enabled
- ▶ ASCDEM-1966 Fixed known issue: OBD freeze frame is saved on unconfirmedDTC timing, but not on `testFailed`
- ▶ Adapted data structure according advanced event displacement (`DemAdvDisplacementPasiveTableLine`) based on vendor specific requirements
- ▶ Added support for calibration of `DemEventOBDReadinessGroup` with multiple readiness groups per event
- ▶ Implemented measurement support for `Dem_FFSegIdx<Name>`
- ▶ Implemented support for PID30 in API `Dem_DcmReadDataOfPID30`
- ▶ ASCDEM-1983 Fixed known issue: ConfirmedDTC bit is set for aged events if `DemExtendedDataCapture == DEM_TRIGGER_ON_FDC_THRESHOLD`
- ▶ ASCDEM-1967 Fixed known issue: Counter-based debouncing does not perform jump behavior in initial state. Hint: Please check your NvM configuration after applying this fix, since the size of the Dem default NV-block may change
- ▶ Extended `FreezeFrameRecordFilter` (0x19-03) with OBD- and Development freeze frames
- ▶ Implemented support for PID31 in API `Dem_DcmReadDataOfPID31`
- ▶ Implemented selectable PendingDTC bit behavior
- ▶ Implemented unused value for `DemPidIdentifier`
- ▶ Implemented support for reporting order of event memory entries



Module version 5.15.4

2015-11-06

- ▶ Improved usability in tresos Studio GUI by introducing a new tab for related OBD parameters
- ▶ ASCDEM-1907 Fixed known issue: MIL indicator is activated, even if DemObdDTC was calibrated to "0"
- ▶ ASCDEM-1919 Fixed known issue: The API "Dem_GetSizeOfFreezeFrameByDTC" calculates a wrong size for all freezeframes (0xFF) if an OBD-freezeframe is configured
- ▶ ASCDEM-1902 Fixed known issue: Calibration of configuration parameter DemEventFailureCycleCounterThreshold does not change qualification of UDS status bit 7 (warningIndicatorRequested)
- ▶ ASCDEM-1903 Fixed known issue: Service 0x0a Permanent faults are not removed if DemExtendedDataRecordTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD
- ▶ Added support for Common OperationCycleRef in EventClass with calibration
- ▶ ASCDEM-1703 Fixed known issue: Dem might use uninitialized data when processing event reports during start-up

Module version 5.15.3

2015-10-02

- ▶ ASCDEM-1839 Fixed known issue: OBD Aging can not be disabled

Module version 5.15.2

2015-09-07

- ▶ ASCDEM-1755 Fixed known issue: If permanent memory is enabled and immediate storage is disabled, the Dem might use wrong NvM blocks
- ▶ Implemented support for freeze frame classes and extended data records greater than 255 byte.
- ▶ Removed compiler warning concerning zero-size DevFFAData[] array
- ▶ ASCDEM-1616 Fixed known issue: P-Ports DemOperationCycle may use wrong port-defined argument values or port / macro may not been generated

Module version 5.15.1

2015-08-28

- ▶ Implemented support for record number 0x00 in APIs Dem_GetFreezeFrameDataByDTC() and Dem_GetSizeOfFreezeFrameByDTC()
- ▶ ASCDEM-1843 Fixed known issue: PID41 DataB bit2 is not forced to enabled.



- ▶ ASCDEM-1816 Fixed known issue: Dem_GetFreezeFrameDataByDTC() does not return `DEM_GET_FF-DATABYDTC_WRONG_RECORDNUMBER` when RecordNumber 0x00 is requested
- ▶ ASCDEM-1824 Fixed known issue: If OCC2-based aged event reoccurs, previous data are not deleted
- ▶ ASCDEM-1845 Fixed known issue: OCC5 is not initialized when an event is stored in memory for the first time
- ▶ ASCDEM-1812 Fixed known issue: ClearDTC and aging do not correctly reset warning indicator-related status
- ▶ Implemented support for calibration of PID41 DataB bit2.
- ▶ Implemented support for Indicator class combination.
- ▶ ASCDEM-1855 Fixed known issue: Compile error if development freeze frame is configured but DET is disabled
- ▶ Implemented support for DTC disabling via calibration
- ▶ Implemented support for deleting internal events via `Dem_ClearDTC()`

Module version 5.15.0

2015-07-10

- ▶ Implemented common freeze frame support
- ▶ Added support for calibration of event classes.
- ▶ Added support for a minimum number of event classes.
- ▶ Added support for Indicator classes.
- ▶ Added support for a minimum number of Indicator classes.
- ▶ ASCDEM-1800 Fixed known issue: Wrong OBDFreezeFrame data stored if internal data elements are configured for a DemPidClass
- ▶ ASCDEM-1771 Fixed known issue: `Dem.GetSizeOfFreezeFrameByDTC()` and `Dem.GetSizeOfExtendedDataRecordByDTC()` do not return error if called with wrong record number for non-stored DTC

Module version 5.14.1

2015-06-20

- ▶ ASCDEM-1725 Fixed known issue: `NvM_ReadAll()` does not load the Dem status block if nonvolatile behavior is used for `Dem_ClearDTC()`

Module version 5.14.0

2015-06-19



- ▶ Implemented support for enhanced OBD readiness groups
- ▶ Implemented support for advanced event displacement
- ▶ ASCDEM-1727 Fixed known issue: A wrong configuration check for the NvMWriteRamBlockToNvCallback name breaks the OBD permanent memory feature
- ▶ Implemented support for filtering of stored DTCs via Dcm (vendor specific)
- ▶ Extended upper multiplicity of DemFreezeFrameClass to 65535 (RFC #61653)
- ▶ ASCPD-179 Added additional calibration parameter
- ▶ ASCDEM-1701 Fixed known issue: Using nonvolatile behavior for Dem_ClearDTC() may block it or lead to incomplete data in NvM
- ▶ Changed default value of the DemExtendedDataCapture and DemFreezeFrameCpature configuration parameters to DEM_TRIGGER_EVENT_MEMORY_STORAGE
- ▶ Dem_SetSI30Symptom is not available if Trigger FiM Reports = false
- ▶ Support for OCC5 based aging of OBD events
- ▶ ASCPD-201 Implemented feature "event burst reduction"/"processor load reduction"
- ▶ Implemented DTC-group DEM_DTC_GROUP_EMISSION_REL_DTCS as a optional group
- ▶ ASCDEM-1698 Fixed known issue: NVM_BLOCK_DEM_PERMANENT is not generated correctly by the Calcualte Service Needs assistant under particular conditions
- ▶ ASCPD-199 Implemented additional freeze frame support for development purpose
- ▶ ASCPD-179 Added additional calibration parameter
- ▶ Modified the optional DTC-group DEM_DTC_GROUP_EMISSION_REL_DTCS to statically contain all DTCs with kind "emission-related" (i.e. the contained DTCs are not configurable any more) and updated Dem_ClearDTC() accordingly

Module version 5.13.0

2015-04-24

- ▶ ASCDEM-1526 Fixed known issue: Compile error for undeclared Dem_FFRecNumerationClass[] occurs if no freeze frames are configured, but configured freeze frame record numbers are enabled
- ▶ Implemented support for OCC5 (operation cycle counter 5)
- ▶ ASCDEM-1634 Fixed known issue: Linker error for extended FiM if DemSupportFiMFDCThreshold is enabled and DemExtendedDataCapture is not set to DEM_TRIGGER_ON_FDC_THRESHOLD
- ▶ ASCDEM-1538 Fixed known issue: Configuration check missing: NvM is triggered with BlockId 0

Module version 5.12.2

2015-03-27



- ▶ Implemented support for OBD Mode \$04
- ▶ ASCDEM-1511 Fixed known issue: The precompiler defines DEM_OPCYC_OBD_DCY_ID and/or DEM_-OPCYC_IGNITION_ID are not generated under certain conditions
- ▶ ASCDEM-1540 Fixed known issue: Dem module triggers a DET error during NvM_WriteAll
- ▶ ASCPD-166 Implemented support of OBD readiness groups in Pid 0x01 and Pid 0x41
- ▶ Added support for further RTE interfaces according to AUTOSAR 4.2.1: DiagnosticInfo:GetEventFreezeFrameData, DiagnosticInfo:GetDTCOfEvent, Callback-EventDataChanged, GeneralCallbackEventDataChanged, CallbackEventStatusChange, GeneralCallbackEventStatusChange, DataServices
- ▶ Added vendor specific configuration switch DemOBDIumprFunctionsEnabled to disable the IUMPR functions: Dem_RepIUMPRFaultDetect(), Dem_RepIUMPRDenLock() and Dem_RepIUMPRDenRe-lease()
- ▶ Implemented support for OBD driving cycle

Module version 5.12.1

2015-03-06

- ▶ Implemented support for OBD DTC
- ▶ Implemented support for OBD Freeze Frame
- ▶ ASCDEM-1609 Fixed known issue: Event qualification with PREFAILED/PREPASSED report doesn't work if the first PREFAILED/PREPASSED report shall qualify the event as FAILED/PASSED
- ▶ Improved calling strategy for EventDataChanged callback on overflow indication
- ▶ ASCDEM-1458 Fixed known issue: Unnecessary trigger of EventStatusChanged / DTCStatusChanged callbacks
- ▶ ASCDEM-1510 Fixed known issue: "EventStatusChanged"/"DTCStatusChanged" callbacks reports a wrong status for BSW events

Module version 5.12.0

2015-02-17

- ▶ ASCPD-179 Added first calibration parameter
- ▶ ASCPD-176 Added OBD variant
- ▶ ASCPD-181 Added extended FiM support
- ▶ ASCDEM-1483 Fixed known issue: "EventStatusChanged" / "DTCStatusChanged" callbacks report a wrong status if event memory is already full



- ▶ ASCDEM-1460 Fixed known issue: If `DemIndicatorFailureCycleSource` is `DEM_FAILURE_CYCLE_INDICATOR`, the configuration parameters `DemIndicatorFailureCycleRef` and `DemIndicatorFailureCycleCounterThreshold` are erroneously not editable
- ▶ ASCDEM-1508 Fixed known issue: The data structures used for permanent memory are mapped to the incorrect memory map section
- ▶ Optimized implementation of `Dem_GetExtendedDataRecordByDTC()` and `Dem_GetEventExtendedDataRecord()`
- ▶ Added licence-based feature-set for:
 - ▶ ASCPD-225 unconfirmed threshold mechanism (supporting additional extended data and freeze frame capture, DTC status indicators, fault detection counters and operation cycle counters)
 - ▶ ASCPD-226 extended displacement mechanism (configurable)
 - ▶ ASCPD-227 extended aging mechanism (configurable), not removing aged event entries unless space is needed
- ▶ ASCDEM-1501 Fixed known issue: If 'immediate storage' is disabled, the NvM callback functions '`Dem_NvMWriteFinishedPermanentMemory`', '`Dem_NvMWriteCopyPermanentMemory`', and '`Dem_NvMReadCopyPermanentMemory`' are not available
- ▶ ASCDEM-1534 Fixed known issue: If OBD support and RTE usage are enabled, the Dem module generates invalid content
- ▶ ASCDEM-1478 Fixed known issue: `EventFailureCycleCounter` is incremented under wrong conditions
- ▶ ASCDEM-1503 Fixed known issue: If permanent memory is enabled, the `Dem_GetNextFilteredDTCAndFDC` and `Dem_GetNextFilteredDTCAndSeverity` functions are returning `DEM_E_WRONG_CONDITION` if DTC origin filter is set to `DEM_DTC_ORIGIN_PERMANENT_MEMORY`
- ▶ ASCDEM-1617 Fixed known issue: The generation of Dem module fails if `DemMILIndicatorRef` parameter is enabled with invalid reference and not editable
- ▶ ASCDEM-1518 Fixed known issue: `Dem_ClearDTC` might not delete an event memory entry consistently if events are reported/qualified in parallel
- ▶ Added support for multiple `Dem_<...>ClearDTC()` interfaces according to AUTOSAR 4.2.1
- ▶ Extended upper multiplicity of `DemDidClass` and `DemDataElementClass` to 65535 (RFC #55110)
- ▶ ASCDEM-1756 Fixed known issue: Wrong internal data element values may be returned and data might get corrupted if more than 242 `DemDataElementClass` elements are configured

Module version 5.9.100

2014-10-22

- ▶ Removed writing of data initialization to NvRAM for empty memory entries if *immediate storage* is enabled to reduce NvRAM aging



- ▶ Updated setting of the CDTC status bit which is now triggered by an event report and does not need a restart of the operation cycle
- ▶ ASCDEM-1220 Fixed known issue: Status bit CDTC might get set incorrectly if *fault confirmation* is enabled and *immediate storage* is in use
- ▶ Replaced simple semaphore mechanism in Dcm interface functions by critical sections
- ▶ ASCDEM-1286 Fixed known issue: CDTC bit is not set in specific configurations with multiple events with EventFailureCycleCounterThreshold enabled
- ▶ ASCDEM-1317 Fixed known issue: BSW event status bit WIR will be kept on if set and reset in the same power cycle
- ▶ ASCDEM-1234 Fixed known issue: ClearDTC and aging do not correctly reset warning indicator-related cycle counters
- ▶ Implemented storing the event status block immediately if DemClearDTCBehavior is set to a DEM_CLR-RESP_NONVOLATILE option
- ▶ ASCDEM-1422 Fixed known issue: The status of the warning indicator may not be updated if the update of event data APIs are called concurrently (i.e. tasks are interrupting each other)
- ▶ Implementation of OBD Permanent Memory support

Module version 5.9.1

2014-04-25

- ▶ ASCDEM-1142 Fixed known issue: Dem_InternalSetEventStatus() != E_OK in Dem_Debounce-TimeTimerTick is wrong and has to be == DEM_E_DET_REPORT
- ▶ ASCDEM-1135 Fixed known issue: Extended data of a new event is not written in case an event has been displaced before
- ▶ Extended the *warning indicator handling* to support that WIR bits can be switched on immediately on FAILED reports
- ▶ ASCDEM-1167 Fixed known issue: Dem_GetFreezeFrameDataByDTC/Dem_GetEventExtended-DataRecord does not provide correct data if more than two DIDs/Extended Data records are configured per event
- ▶ ASCDEM-1163 Fixed known issue: If indicators are configured, but none is linked to any event, a compiler error occurs
- ▶ ASCDEM-1172 Fixed known issue: Constraint definitions for IndicatorIdType and RatioIdType might be incorrectly generated
- ▶ ASCDEM-1153 Fixed known issue: Time-based debouncing gets blocked internally if event qualification is locked due to an enable condition
- ▶ Corrected several warnings during importing the file Dem_Bswmd.arxml into the system model



- ▶ ASCDEM-1202 Fixed known issue: Event-related data of BSW events might get updated even if the reported event status does not change
- ▶ ASCDEM-1224 Fixed known issue: For *warning indicator handling* the Dem might ignore cycles with **PASSED** results but count untested cycles for counting failure cycles

Module version 5.9.0

2013-11-21

- ▶ Extended *event displacement* not ignoring the occurrence order even if all events have neither extended data nor freeze frames configured
- ▶ Reworked memory sections for internal variables used in configurations with *immediate storage* feature enabled
- ▶ Added non-volatile storage support for operation cycle states and FDCs for counter-based debounced events
- ▶ Implemented support for `ClearDTC` behavior configuration
- ▶ Corrected several warnings during importing the file `Dem_Bswmd.arxml` into the system model

Module version 5.8.0

2013-10-11

- ▶ ASCDEM-1019 Fixed known issue: `DemCallbackEventDataChanged` is incorrectly triggered on aging for events with any internal data element except aging counter configured to extended data
- ▶ ASCDEM-1028 Fixed known issue: Undefined identifier errors are reported for certain Dem configurations with Rte usage enabled
- ▶ Implemented Handheld Wizard support for calculation of `DemEventId`, `DemEnableConditionId` and `DemIndicatorID`
- ▶ Updated default value of configuration parameter `DemDebounceAlgorithmClass` to `DemDebounce-MonitorInternal`
- ▶ Added warning message to `DemExtendedDataRecordClass` if the user maps any internal data elements to an ED segment where `DemExtendedDataRecordUpdate` parameter is set to false
- ▶ Implemented support for *Event Displacement*
- ▶ Updated NvM service needs calculation - disabled permanent RAM block CRC calculation and removed static block length
- ▶ Updated `DTCRecordUpdate` functionality according to AUTOSAR 4.0 Rev 3
- ▶ ASCDEM-1091 Fixed known issue: Configuration parameter `DemImmediateNvStorageLimit` set to 255 does not limit the write to NV memory
- ▶ Updated `ResetEventStatus` functionality from AUTOSAR 4.0 Rev 2 to AUTOSAR 4.0 Rev 3



- ▶ Added support for function tracing via AUTOSAR Debugging
- ▶ Updated clearing of event memory on aging for an event with DTC record update disabled

Module version 5.6.0

2013-06-14

- ▶ ASCDEM-976 Fixed known issue: Implementation Data Types of category `TYPE_REFERENCE` refer a `DataConstr` when it is not allowed
- ▶ Added internal Dcm/Dem support functions to save the event ID of the last requested DTC for subsequent use (run-time optimization)
- ▶ Added support for NvM single block callback in Service Needs Assistant
- ▶ ASCDEM-959 Fixed known issue: Events without DTC-class are incorrectly assigned to DTC-group `DEM_DTC_GROUP_EMISSION_REL_DTCS`
- ▶ ASCDEM-991 Fixed known issue: If *immediate storage* is enabled and no freeze frame classes are configured, unresolved symbol error occurs
- ▶ Updated *immediate storage* to support immediate deletion of event memory entry from NVRAM on *event displacement*
- ▶ Updated handling of valid, but non-stored records by `Dem_GetEventFreezeFrameData` and `Dem_GetEventExtendedDataRecord`
- ▶ Updated return values of `Dem_GetFreezeFrameDataByDTC()` in case of valid but not stored entries to match requirement Dem630
- ▶ Implemented support for *fault confirmation*
- ▶ Implemented support for *Event healing - Warning indicator handling*
- ▶ Added support for dynamic DTC values as part of *side allocation* feature

Module version 5.5.0

2013-02-08

- ▶ Updated support for *BSW error-queue handling*
- ▶ Updated implementation of `Dem_GetSizeOfExtendedDataRecordByDTC()` based on configuration parameter `DemGetSizeOfExtendedDataRecordByDTCOptimization`, considering both AUTOSAR 4.0 Rev 3 and Bugzilla issue http://www.autosar.org/bugzilla/show_bug.cgi?id=52426
- ▶ Implemented support for *configured freeze frame record numbers*

Module version 5.4.0

2012-10-12



- ▶ ASCDEM-882 Fixed known issue: Dem does not compile when legacy symbolic names are disabled via macro `DEM_DONT_PROVIDE_LEGACY_SYMBOLIC_NAMES`
- ▶ ASCDEM-885 Fixed known issue: Aging related internal function is called but not defined in some configurations and aging disabled
- ▶ ASCDEM-898 Fixed known issue: `Dem_SetAgingCycleState()` will not be available if more than one aging cycle is configured
- ▶ ASCDEM-899 Fixed known issue: If an event configures only one freeze frame record slot, this record will be updated if the event re-occurs
- ▶ Implemented support for `DemImmediateNvStorageLimit`
- ▶ Updated configuration and API to AUTOSAR 4.0.3:
 - ▶ Added the following configuration parameters: `DemAgingCycle`, `DemStatusBitHandlingTestFailedSinceLastClear`, `DemObdDTC`, `DemUdsDTC`, `DemSecondaryFunctionIdRef`
 - ▶ Updated the following configuration parameters: `DemExtendedDataCapture`, `DemFreezeFrameCapture`, `DemAgingCycleRef`
 - ▶ Removed the following configuration parameters: `DemDTC`, `DemDTCKind`
 - ▶ Renamed the DET error code for invalid address from `DEM_PARAM_ADDRESS` to `DEM_E_PARAM_POINTER`
 - ▶ Adapted the following types: `Dem_ReturnGetStatusOfDTCType`, `Dem_ReturnGetNextFilteredDTCType`, `Dem_ReturnClearDTCType`, `Dem_ReturnGetFreezeFrameDataByRecordType`, `Dem_ReturnGetExtendedDataRecordByDTCType`, `Dem_ReturnGetDTCByOccurrenceTimeType`, `Dem_ReturnGetFreezeFrameDataByDTCType`, `Dem_ReturnGetSizeOfExtendedDataRecordByDTCType`, `Dem_ReturnGetSizeOfFreezeFrameByDTCType`, `Dem_ReturnGetSeverityOfDTCType`
 - ▶ Added the following types: `Dem_ReturnDisableDTCRecordUpdateType` and `Dem_DTCTFormatType`
 - ▶ Renamed the return value from `E_NO_DTC_AVAILABLE` to `DEM_E_NO_DTC_AVAILABLE`
 - ▶ Updated the argument from `DTCKind` to `DTCTFormat` in the following APIs: `Dem_GetDTCOfEvent()`, `Dem_ClearDTC()`
 - ▶ Removed the argument `BuffSize` from the following APIs: `Dem_GetEventExtendedDataRecord()`, `Dem_GetEventFreezeFrameData()`
 - ▶ Added the argument `DTCTFormat` in the following APIs: `Dem_SetDTCSuppression()`, `Dem_SetDTCFilter()` (retained kind), `Dem_SetFreezeFrameRecordFilter()`
 - ▶ Removed the argument `DTCKind` from the following APIs: `Dem_GetFreezeFrameDataByRecord()`, `Dem_GetFreezeFrameDataByDTC()`, `Dem.GetSizeOfFreezeFrameByDTC()`, `Dem_GetExtendedDataRecordByDTC()`, `Dem.GetSizeOfExtendedDataRecordByDTC()`
 - ▶ In the port `GetEventExtendedDataRecord` the output parameter name changed from `ExtData` to `DestBuffer`



- ▶ In the port `GetEventFreezeFrameData` the following parameter names changed: `ReportFullRecord` to `ReportTotalRecord`, `DID` to `DataId`, `FFData` to `DestBuffer`
- ▶ Renamed the `PortArgument` from `OverflowIndPersistentMemory` to `OverflowIndPermanentMemory`
- ▶ Updated the following Operations/Dataelements: `GetDTCOfEvent`, `ClearDTC`: updated the parameter from `DTCKind` to `DTCFormat`. `DTCSuppression`: added the parameter `DTCFormat`
- ▶ The top-level structure of the software-component description in the ARXML files changed from `/AUTOSAR/Dem` to `/AUTOSAR_Dem`

Module version 5.3.0

2012-06-25

- ▶ Added configuration parameter `DemDcmUsage` which enables or disables the usage of Dcm by Dem
- ▶ Introduced Unreachable code assert and Precondition assert using `DemDefensiveProgramming` container
- ▶ ASCDEM-862 Fixed known issue: `SwcBswMappings` is located at the wrong location in BSWMD
- ▶ Implemented support for Reset PDTC bit as per upcoming ISO 14229-1 and Bugzilla issue http://www-autosar.org/bugzilla/show_bug.cgi?id=45036
- ▶ Updated counter based debouncing jump behavior (as per AUTOSAR 4.1 Rev 1 and Bugzilla issue http://www-autosar.org/bugzilla/show_bug.cgi?id=52969)
- ▶ ASCDEM-883 Fixed known issue: Definition of Implementation Data Type `Dem_DTCOriginType` is defined in `Dem_swc_internal.arxml` only

Module version 5.2.0

2012-03-23

- ▶ ASCDEM-692 Fixed known issue: `DATA-CONSTR-REF` at wrong position in arxml-files of Dem
- ▶ ASCDEM-694 Fixed known issue: Recommended configuration schema cannot be found while creating Dem project in EB tresos Studio
- ▶ ASCDEM-695 Fixed known issue: `Dem_GetSizeOfFreezeFrameByDTC()` uses old return-type
- ▶ ASCDEM-697 Fixed known issue: `Dem_MaxDataValueType` type definition missing in SW-CD
- ▶ ASCDEM-696 Fixed known issue: Missing calls to `GeneralCBStatusEvt` port
- ▶ Introduced BSW-event debouncing during pre-initialized phase (between `Dem_PreInit()` and `Dem_Init()`) in `Dem_ReportErrorStatus()`
- ▶ Implemented support for *Clear event allowed* callbacks (via RTE and C-function)



- ▶ Updated naming scheme for macros of symbolic name values to AUTOSAR 4.0 Rev 3 naming scheme
- ▶ Implemented support for *Event memory overflow indication*
- ▶ Implemented support for *Occurrence counter* and internal data element `DEM_OCCCTR`
- ▶ Updated Dem-Dcm interaction to AUTOSAR R4.0:
 - ▶ Adapted `Dem_GetNextFilteredRecord()` by returning relative addressed FF record.
 - ▶ Implemented support for *Functional unit* via API `Dem_GetFunctionalUnitOfDTC()`.
 - ▶ Removed record number from the output data of `DestBuffer` in `Dem_GetExtendedDataRecordByDTC()`
 - ▶ Updated output data of `SizeOfExtendedDataRecord` to include size of extended record number(s) in `Dem.GetSizeOfExtendedDataRecordByDTC()`
 - ▶ Implemented support for `ExtendedDataNumber 0xFE` via API `Dem.GetSizeOfExtendedDataRecordByDTC()`
 - ▶ Introduced check for `DTCSettingAllowed` in `Dem_ReportErrorStatus()` and `Dem_SetEventStatus()`
- ▶ Updated naming of `Fim_DemTriggerOnEventStatus()` to `FIM_DemTriggerOnEventStatus()` (prefix changed) as per AUTOSAR 4.0 Rev 3
- ▶ Implemented support for *Immediate storage*
- ▶ ASCDEM-832 Fixed known issue: `DTCStatusChanged` and `Dem_DcmTriggerOnEventStatus()` callbacks are not triggered for some configurations

Module version 5.1.0

2011-09-07

- ▶ Implemented support for internal data element `DEM_AGINGCTR` and updated aging processing according to AUTOSAR R4.0
- ▶ Implemented support for separate aging counters via API `Dem_SetAgingCycleState()`
- ▶ Initial AUTOSAR R4.0 version

3.3.2.2. New features

- ▶ Support for intermediate DTC status storage.

Description:

To preserve the result of the last diagnosis in case of a power failure or hard reset, the Dem module was extended with support for the intermediate DTC status storage.



If configuration parameter `DemIntermediateNvStorageOfDTCStatus` is enabled, the Dem stores the DTC status in the NvRAM whenever the immediate storage of the event related data is triggered, in addition to the usual storage at shutdown.

3.3.2.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- ▶ Extended buffer mechanism for the API `Dem_ReportErrorStatus()`

Description:

The buffer mechanism for the API `Dem_ReportErrorStatus()` (configured via `DemBswErrorBufferSize`) is extended to the ECU sleep and wake-up phases (i.e. after `Dem_Shutdown()`).

- ▶ Dependency on Dcm can be switched off

Description:

The inclusion of `Dcm_Types.h`, which defines `Dcm_OpStatusType` for the Dcm function calls `Dem_DcmGetInfoTypeValue08` and `Dem_DcmGetInfoTypeValue0B`, can be controlled via the configuration parameter `DemDcmUsage`. This allows a simplified integration of the Dem module and ECUs without a Dcm module.

- ▶ Optimized resource usage according to HIS recommendations

Description:

The Dem generates ports and interface functions only for those events, which actually need them by implementing the HIS requirements HisDem0001, HisDem0002, HisDem0003, HisDem0004, HisDem0005, HisDem0006, HisDem0007, HisDem0008, HisDem0009.

- ▶ Dem behavior when occurrence counter exceeds `DemImmediateNvStorageLimit`

Description:

Following modifications are adapted:

- ▶ Dem module continues to update the Dem internal event memory as against Dem552. However, it will not trigger the storage in non-volatile memory when the occurrence counter has exceeded the value defined by the configuration parameter `DemImmediateNvStorageLimit`.
- ▶ If the occurrence counter of an event (with `DemImmediateNvStorage` enabled) exceeds `DemImmediateNvStorageLimit`, the event memory entry and its event related data will be stored persistently during the shutdown phase.
- ▶ Updated return values of APIs `Dem_GetEventFreezeFrameData` and `Dem_GetEventExtendedDataRecord`



Description:

The return values of the APIs `Dem_GetEventFreezeFrameData` and `Dem_GetEventExtended-DataRecord` are updated to `E_NOT_OK` for valid, but non-stored FF records and ED records.

- ▶ Support for *side allocation*

Description:

Statically configured DTCs of events can be changed by application components during runtime. The Dem supports configuring a callout function to be called every time the Dem needs to retrieve a DTC for an event.

- ▶ Function tracing support via AUTOSAR Debugging

Description:

The module Dem 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 non-volatile storage of FDCs

Description:

FDC is stored non-volatile for events with counter based debouncing if `DemDebounceCounterStorage` is enabled.

- ▶ Support for frequency-based debouncing of diagnostic events

Description:

The frequency-based debouncing algorithm feature from R3.1 (Dem SWS chapter 7.2.11.3.3) is still supported and it can be configured by setting container `DemDebounceAlgorithmClass` to `DemDebounceFrequencyBased`.

- ▶ Enhanced multiplicity of `DemFreezeFrameClass` and `DemDidClass` and `DemDataElementClass`

Description:

The upper multiplicity of `DemFreezeFrameClass` and `DemDidClass` and `DemDataElementClass` configuration container has been extended from 255 to 65535 according to RFC 61653 and RFC 55110. This



provides the possibility to configure ECUs, which can contain several thousand of different FF class, DIDs (and data elements), e.g., in the powertrain domain.

- ▶ Support for configurable displacement strategy according to R4.2

Description:

The displacement strategy can be selected via the R4.2 configuration parameter `DemEventDisplacementStrategy`, i.e., the consideration of the active/passive status can be disabled by choosing the configuration option `DEM_DISPLACEMENT_PRIO_OCC`.

- ▶ Support for additional specific features (ASCPD-225, ASCPD-226, ASCPD-227)

Description:

New parameters, counters and status bits are added or extended to allow processing events also on "unconfirmed" threshold during debouncing, which can be lower than the failed threshold:

- ▶ Added configuration parameter `DemDebounceCounterUnconfirmedThreshold`. This is the unconfirmed threshold on which the event is processed.
- ▶ Extended configuration parameters `DemExtendedDataCapture` and `DemFreezeFrameCapture` by the option `DEM_TRIGGER_ON_FDC_THRESHOLD` to allow processing on unconfirmed threshold
- ▶ Added new internal data elements, which can be mapped to extended data records:

Data element	Description
<code>DEM_CURRENT_FDC</code>	Fault detection counter
<code>DEM_FDC12</code>	Maximum fault detection counter since last clear
<code>DEM_OCC1</code>	Operation cycles since last unconfirmedDTC counter
<code>DEM_OCC2</code>	unconfirmedDTC aging counter
<code>DEM_OCC3</code>	Operation cycles since first unconfirmedDTC counter
<code>DEM_OCC4</code>	unconfirmedDTC operation cycle counter
<code>DEM_OCC5</code>	Number of warm-up cycles since the DTC commanded the MIL to switch off
<code>DEM_OCC6</code>	Consecutive failed operation cycle counter
<code>DEM_OCC7</code>	Qualified/unconfirmedDTC operation cycle counter
<code>DEM_SI30</code>	DTC status indicators

- ▶ Extended displacement mechanism to be configurable based on unconfirmedDTC aging counter (OCC2).



- ▶ Extended healing and aging algorithms to be configurable based on unconfirmedDTC aging counter (OCC2).
- ▶ Support of additional enumerations in DemEventOBDReadinessGroup

Description:

In the configuration parameter DemEventOBDReadinessGroup the additional enumeration value DEM_-OBD_RDY_FLSYS_NONCONT is supported according to AUTOSAR R4.2.1.

- ▶ Support for operation cycle automatic end feature for all operation cycle types

Description:

AUTOSAR defines the automatic end of operation cycles feature only for the OBD driving cycle in SWS_-Dem_00697, ECUC_Dem_00837. Since the feature of operation cycles automatic end may also be useful for other operation cycles than OBD driving cycle, the Dem supports this for every configured operation cycle which has the configuration parameter DemOperationCycleAutomaticEnd configured to TRUE. Every operation cycle, that has DemOperationCycleAutomaticEnd configured to TRUE is ended automatically by Dem during Dem_Shutdown().

- ▶ Partial support of AUTOSAR R4.2.1 RTE interfaces

Description:

The following AUTOSAR R4.2.1 RTE interfaces are supported:

- ▶ DiagnosticInfo:GetEventFreezeFrameData
 - ▶ DiagnosticInfo:GetDTCOfEvent
 - ▶ CallbackEventDataChanged
 - ▶ GeneralCallbackEventDataChanged
 - ▶ CallbackEventStatusChange
 - ▶ GeneralCallbackEventStatusChange
 - ▶ DataServices
- ▶ Support for advanced event displacement

Description:

The advanced event displacement algorithm uses the static and dynamic event priority. Static event priority is used for displacement configured by DemEventPriority. Dynamic event priority is calculated in consideration of the extended data items SI30 (Bit4, 5, 6), OCC1 (DemAdvDisplacementOcc1Limit) and the configured passive table. The passive table consists of passive table lines holding references to reported events DemReportedEventRef and references to passive events DemPassiveEventRef. Every passive table line holds one reported event and several passive events, which can be displaced by that reported event under certain conditions.



The following configuration parameters were introduced:

Parameter	Description
DemAdvDisplacementOcc1Limit	The number of OCC1 counts used for selecting entry by the advanced displacement algorithm.
DemAdvDisplacementPassiveTableLine	Describes a line in the passive table used in the advanced displacement algorithm.
DemReportedEventRef	Reference to a <code>DemEventParameter</code> corresponding to the event currently reported and trying to get an event memory entry.
DemPassiveEventRefs	List of references to <code>DemEventParameter</code> corresponding to an event currently stored in an event memory entry and subject for displacement.
DemPassiveEventRef	Reference to a <code>DemEventParameter</code> corresponding to the event currently stored in an event memory entry and subject for displacement.

The advanced event displacement is protected by the license `EB_DIAG_FP1`.

The advanced event displacement is enabled under the following conditions:

- ▶ `DemEventDisplacementSupport` is set to `TRUE`
- ▶ `DemEventDisplacementStrategy` is set to `DEM_DISPLACEMENT_FULL`
- ▶ `DemExtendedDataCapture` is set to `DEM_TRIGGER_ON_FDC_THRESHOLD`
- ▶ Internal data elements `DEM_SI30`, `DEM_OCC1`, `DEM_OCC2`, `DEM_CURRENT_FDC` must be configured.
- ▶ Support for filtering of stored DTCs via Dcm

Description:

The Dem module is extended with the following vendor specific API functions:

- ▶ `Dem_DcmSetStoredDTCFilter()`
- ▶ `Dem_DcmGetNextFilteredStoredDTC()`

These new API functions are also based on the SetDTCFilter concept of AUTOSAR.

- ▶ Support additional OCC2 and OCC5 based event aging

Description:

The Dem module was extended by an algorithm for event aging. This algorithm takes the OCC2 counter for aging of UDS events and the OCC5 counter for aging of OBD events into account.



The Dem module is extended with the following vendor specific API configuration parameter:

- ▶ `DemOBDAgingCycleCounterThreshold`
- ▶ Event Burst Reduction/Processor Load Reduction

Description:

To prevent an excessive workload of the `Dem_MainFunction` the user can specify a maximum count of failed and passed error queue entries to be processed each schedule of the `Dem_MainFunction`.

For that the Dem module is extended with the following vendor specific configuration parameters:

- ▶ `DemMaxHandledPassedEventsPerScheduling`
- ▶ `DemMaxHandledFailedEventsPerScheduling`
- ▶ Support for development freeze frames

Description:

Development freeze frames are additional freeze frames which are fetched together with the first freeze frame record. The development freeze frames can be requested with a specific record number by the Dcm. The Dem provides two kinds of development freeze frames: A and B. For each kind the number of events which can store a development freeze frame can be configured to save resources. If no space is available, development freeze frames are displaced based on the same criteria as regular event memory entries. The development freeze frame A provides a common freeze frame part. This part is equal in all development freeze frame A.

The development freeze frames are not supported if event combination is enabled. In addition it is required that event displacement support is enabled.

The development freeze frames can only be used for the events with primary memory destination. Immediate storage functionality does not affect the development freeze frames, i.e. they are never stored immediately.

The development freeze frames are not prestored, i.e., they are not supported by the API `Dem_Pre-storeFreezeFrame()`, and they are only accessible to Dcm, i.e., they cannot be read by the API `Dem_-GetEventFreezeFrameData()`.

The development freeze frames are protected by the license `EB_DIAG_OBD` and `EB_DIAG_FP1`.

The Dem module is extended with the following vendor specific configuration parameters:

- ▶ `DemDevFreezeFrameSupport`
- ▶ `DemCommonFFDataDevAClassRef`
- ▶ `DemMaxNumberDevFreezeFrameAEntry`
- ▶ `DemMaxSizeDevFreezeFrameAEntry`



- ▶ DemMaxNumberDevFreezeFrameBEntry
- ▶ DemMaxSizeDevFreezeFrameBEntry
- ▶ DemDevFreezeFrameAClassRef
- ▶ DemDevFreezeFrameARecNum
- ▶ DemDevFreezeFrameBClassRef
- ▶ DemDevFreezeFrameBRecNum
- ▶ Support for common freeze frame

Description:

The common freeze frame is an additional freeze frame which is captured with the first freeze frame record configured for a specific event. The content of a standard freeze frame (common + individual part of freeze frame) can be requested by standard API `Dem_GetNextFreezeFrameData()` where the common part is appended to the individual part (event specific) of freeze frame. If the configured event does not reference to a freeze frame class (no individual freeze frame) only the common freeze frame will be stored.

The Dem module is extended with the following vendor specific configuration parameters:

- ▶ DemCommonFFDataClassRef
- ▶ Calibration of Common and Development Freeze Frames

Description:

Support for calibration of enabling, disabling, and modification of the: regular Common Freeze Frame class, individual Development Freeze Frame A class, Common Development Freeze Frame A class, and individual Development Freeze Frame B class.

- ▶ Implemented support for DTC disabling via calibration

Description:

The Dem module was extended with support to completely disable a DTC by calibration (both storage and visibility towards tester). Calibration is done via separate bit-array which has one bit for each configured event. This feature is only available when the configuration parameter `DemCalibrationSupport` is configured to `TRUE`.

- ▶ Implemented support to also clear disabled DTCs

Description:

The Dem module was extended with support to also clear disabled DTCs in the API request `Dem_ClearDTC`, if all DTCs are requested per DTC group. This feature can be selected with the configuration parameter `DemClearDTCOfDisabledEvents`, with value `TRUE`.

- ▶ Support of AUTOSAR R4.3.0 `Dem_SetDTCSuppression()` API



Description:

The Dem module was extended with support to dynamically change the DTC suppression status via `Dem_SetDTCSuppression()` API. A suppressed DTC is not visible for an external tester via any service request but does not stop the event processing of the corresponding event. This feature can be enabled with configuration parameter `DemSuppressionSupport`.

- ▶ Support of AUTOSAR R4.3.0 `Dem_SetEventAvailable()` API

Description:

The Dem module was extended with support to dynamically change the event availability via `Dem_SetEventAvailable()` API. If an event is set to unavailable, it is treated as if it is not configured in the system. This feature can be enabled with configuration parameter `DemAvailabilitySupport`.

- ▶ Implemented support for always available Extended Data Records

Description:

The Dem module configuration has a new parameter allowing the user to configure an Extended Data Record to be always available, regardless of the existence of an event memory entry.

This feature can be used for all extended data records which consist of data elements for which the Dem module can always provide or calculate the value.

- ▶ Support of common operation cycle

Description:

The common operation cycle provides the ability to configure one operation cycle (per event) that serves as the following event specific operation cycle references.

- ▶ `DemAgingCycleRef`
- ▶ `DemEventFailureCycleRef`
- ▶ `DemIndicatorFailureCycleRef`
- ▶ `DemIndicatorHealingCycleRef`

If the common operation cycle support is enabled, these different operation cycle references are referencing the common operation cycle reference `DemOperationCycleRef`. This approach makes it easier to configure and calibrate the different operation cycles because only one operation cycle reference has to be configured respectively calibrated.

The Dem module is extended with the following vendor specific configuration parameters:

- ▶ `DemCommonOperationCycleSupport`
- ▶ Support of different occurrence order (`OccOrder`) processing

Description:



The Dem supports an additional behavior for the processing of the occurrence order (OccOrder) of an event.

Under consideration of the configuration parameter `DemUpdateOccOrderOnEventEntryUpdate` the OccOrder is processed in the following ways:

- ▶ Parameter `DemUpdateOccOrderOnEventEntryUpdate` is set to TRUE: Update by highest value + 1 when a new entry is added into event memory or an entry is updated due to an event failure report.
- ▶ Parameter: `DemUpdateOccOrderOnEventEntryUpdate` is set to TRUE: Update by highest value + 1 only when a new entry is added into event memory for the first time.

The Dem module is extended with the following vendor specific configuration parameter:

- ▶ `DemUpdateOccOrderOnEventEntryUpdate`. The default value is TRUE.

▶ Event storage handling for DTC value zero

Description:

An event with the configured/calibrated DTC value 0 will not be stored or updated in event memory (i.e. no entry, no freeze frame data, no extended data) dependent on the value of the EB-specific configuration parameter `DemStoreInternalEvents`.

▶ Record numbers without Freeze Frame Class

Description:

Support of non-storable freeze frame records, i.e., valid freeze frame record numbers without freeze frame data/class. This results into positive response (without freeze frame data) in diagnostic service (e.g. 0x19 04) in case of requesting a valid record number without freeze frame class.

▶ Read number of event memory entries

Description:

Support of the AUTOSAR R4.2.1 API `Dem_GetNumberOfEventMemoryEntries()` and RTE interface `EvMemOverflowIndication` extension with operation `GetNumberOfEventMemoryEntries` to return the number of event memory entries currently stored in the event memory.

▶ Extended behavior of clearing OBD freeze frames

Description:

OBD freeze frames are cleared under the following conditions:

- ▶ The event is displaced by an event with higher priority.
- ▶ Aging conditions are fulfilled, and the event becomes aged.
- ▶ Pending DTC status is cleared (PDTC status bit changes 1 -> 0) before the DTC was confirmed.



▶ Internal data element event priority

Description:

The event priority can be used as internal data element and mapped to extended data.

▶ Internal data element event id

Description:

The event id can be used as internal data element and mapped to extended data.

▶ Internal data element Cycles since last failed

Description:

An internal data element DEM_CYCLES_SINCE_LAST_FAILED can be mapped to extended data. The implementation is according to AUTOSAR R4.1/R4.2/R4.3 specification.

▶ Internal data element Cycles since first failed

Description:

An internal data element DEM_CYCLES_SINCE_FIRST_FAILED can be mapped to extended data. The implementation is according to AUTOSAR R4.1/R4.2/R4.3 specification.

▶ Monitor re-initialization with `InitMonitorReason DEM_INIT_MONITOR_REENABLED`

Description:

If the configuration parameter `DemGeneral/DemCallbackInitMForEReenabledSupport` is set to 'true' Dem offers support for callback `DemInitMonitorForEvent` with `InitMonitorReason DEM_INIT_MONITOR_REENABLED` as described in AUTOSAR R4.2.2 specification. Range of `Dem_InitMonitorReasonType` is extended with `DEM_INIT_MONITOR_REENABLED` and `DEM_INIT_MONITOR_STOREAGE_REENABLED`.

▶ `Dem_ClearDTC` burst reduction

Description:

Dem module was extended with support to asynchronous processing of `Dem_ClearDTC()` API requests from the `Dem_MainFunction()` context. The maximum number of processed events per each `Dem_MainFunction()` call can be adjusted with the configuration parameter `DemMaxNumberClearEventsPerCycle`.

▶ Support for configurable queue sizes generated by RTE for Dem

Description:



To prevent an overflow of the RTE queue for Dem CS operation calls, the Dem module was extended with a configurable queue size for all RTE operations. New configuration parameters have been introduced for setting each queue size value per operation.

- ▶ Reset/freeze event debounce status for frequency-based debouncing via `Dem_ResetEventDebounceStatus()` API

Description:

If an event is configured with frequency-based debouncing, `Dem_ResetEventDebounceStatus()` API can be used in order to control the internal debounce frequency timer via the following behaviors: FREEZE and RESET.

Since the current AUTOSAR specification (R4.0.3) does not describe the frequency-based event debouncing, internal EB requirements are defined in order to specify the functionality for the reset/freeze behavior for event debouncing via `Dem_ResetEventDebounceStatus()`. Frequency-based event debouncing is a type of debouncing that is supplier-specific, ported from R3.1.

- ▶ Reset/freeze event debounce status for frequency-based debouncing via non-fulfilled enable conditions and disabled ControlDTCSettings

Description:

If an event is configured with frequency-based debouncing, the internal debounce frequency timer can be freezed/reset/freeze via non-fulfilled enable conditions and disabled ControlDTCSetting, depending on configuration parameter `DemDebounceBehavior`.

Since current AUTOSAR specifications (ASR4.0.3) do not describe the frequency-based event debouncing (this type of debouncing is supplier-specific, ported from R3.1), internal EB requirements are defined in order to specify the functionality for the reset/freeze behavior for event debouncing via non-fulfilled enable conditions and disabled ControlDTCSetting.

- ▶ Support project specific API `Dem_ReportErrorStatus()`

Description:

If all Dem events are configured as SWC, Dem will not provide the declaration and definition of the API `Dem_ReportErrorStatus()`. Support project specific API `Dem_ReportErrorStatus()` implementation through inclusion of separate module external header-files into provided Dem template header file. Inclusion can be controlled via the configuration parameter `DemIncludeProjectSpecificCustomizationSupport`.

- ▶ Implemented support for DTC DataService interface with EventID

Description:



The Dem module was extended with support for extended Callback interface for CS External Data Elements. If the configuration parameter `DemDataElementInterfaceWithEventParameter` of the Data Element is enabled, the prototype of the Callback will be extended with another parameter, `EventID`.

- ▶ Support for event displacement according to ASR4.3.1

Description:

The event displacement algorithm is extended by the features:

- ▶ protection of memory entries of emission related events to be displaced by new reported events with the same priority and
- ▶ displacement of memory entries of events which are not tested the current operation cycle.

New configuration parameters:

Parameter	Description
<code>DemOBDEventDisplacement</code>	Enables/disables a different displacement behavior for OBD events. If enabled, an OBD event that triggers the MIL indicator or holds the OBD Freeze Frame or has the Pending-DTC status bit set will not be displaced by a new event with the same priority.
<code>DemEventDisplacementCriterionTNCTOC</code>	Enables/disables the consideration of the TNCTOC event status bit. If enabled, event entries with TNCTOC bit == 1 will be considered for displacing in accordance with ASR4.3.1. If disabled, the TNCTOC bit will not be considered (ASR4.0.3 compliant behavior).

- ▶ Support for disabling the trigger for status changed notification callbacks on start of operation cycle

Description:

To reduce execution time of the `Dem_SetOperationCycleState()` with state `DEM_CYCLE_STATE_START` the user can disable the trigger point for `EventStatusChanged/DTCStatusChanged` callbacks. Such an option is of interest, for projects where the application does not need to know the evolution of the DTC status for events over operation cycle start/re-start. This feature can be controlled via the configuration parameter `DemCallbackEventStatusChangedOpCycStartSupport`.

- ▶ Optimized calibratable data

Description:

A2L files should be regenerated regardless of whether configuration has changed or not.



- ▶ Support for parallel processing of multiple diagnostic requests

Description:

The EB Dem solution provides the possibility of defining multiple diagnostic clients. A diagnostic client in the EB Dem solution is defined as a BSW (e.g. Dcm) module that accesses the Dem module in order to delegate diagnostic services, which are related to the event memory. There are multiple clients that can request access to the fault memory. A ClientId defines a unique identifier for a Dem client. This number is used by a client in the ClientId parameter in all APIs used for diagnostic services. With this new feature, the parallel processing on the same event memory by multiple clients (BSW/SW-C/CDD) is enabled and DTC selection process for operations used to retrieve event data, clear DTC, DTC status is optimized.

In particular, the signatures of the following Dem APIs, which are used by the Dcm module to process the OBD and UDS diagnostic requests, are implemented according to AUTOSAR R4.3.1 and support ClientId as a parameter:

- ▶ `Dem_ClearDTC()`
- ▶ `Dem_GetDTCSelectionResult()`
- ▶ `Dem_GetDTCSelectionResultForClearDTC()`
- ▶ `Dem_SelectDTC()`
- ▶ `Dem_GetTranslationType()`
- ▶ `Dem_GetDTCStatusAvailabilityMask()`
- ▶ `Dem_GetStatusOfDTC()`
- ▶ `Dem_GetSeverityOfDTC()`
- ▶ `Dem_GetFunctionalUnitOfDTC()`
- ▶ `Dem_SetDTCFilter()`
- ▶ `Dem_GetNumberOfFilteredDTC()`
- ▶ `Dem_GetNextFilteredDTC()`
- ▶ `Dem_GetNextFilteredDTCAndFDC()`
- ▶ `Dem_GetNextFilteredDTCAndSeverity()`
- ▶ `Dem_SetFreezeFrameRecordFilter()`
- ▶ `Dem_GetNextFilteredRecord()`
- ▶ `Dem_GetDTCByOccurrenceTime()`
- ▶ `Dem_DisableDTCRecordUpdate()`
- ▶ `Dem_EnableDTCRecordUpdate()`
- ▶ `Dem.GetSizeOfExtendedDataRecordSelection()`
- ▶ `Dem.GetSizeOfFreezeFrameSelection()`



- ▶ `Dem_GetNextExtendedDataRecord()`
 - ▶ `Dem_GetNextFreezeFrameData()`
 - ▶ `Dem_SelectExtendedDataRecord()`
 - ▶ `Dem_SelectFreezeFrameData()`
 - ▶ `Dem_DisableDTCSetting()`
 - ▶ `Dem_EnableDTCSetting()`
 - ▶ `Dem_DcmReadDataOfOBDFreezeFrame()`
 - ▶ `Dem_DcmGetDTCOfOBDFreezeFrame()`
- ▶ Synchronous Dcm interface

Description:

Following API functions are implemented in a synchronous way and do not make use of the `DEM_PENDING` return values: `Dem_GetNumberOfFilteredDTC()`, `Dem_GetNextFilteredDTC()`, `Dem_GetNextFilteredDTCAndFDC()`, `Dem_GetNextFilteredDTCAndSeverity()`, `Dem_GetDTCElectionResult()`, `Dem_GetDTCSelectionResultForClearDTC()`, `Dem_GetStatusOfDTC()`, `Dem_DisableDTCRecordUpdate()`, `Dem_EnableDTCRecordUpdate()`, `Dem_GetSeverityOfDTC()`, `Dem_GetFunctionalUnitOfDTC()`, `Dem_GetNextFilteredRecord()`, `Dem.GetSizeOfExtendedDataRecordSelection()`, `Dem.GetSizeOfFreezeFrameSelection()`, `Dem_GetNextExtendedDataRecord()`, `Dem_GetNextFreezeFrameData()`. This allows for a more compact and ROM-saving implementation.

- ▶ Synchronous J1939Dcm interface

Description:

Following API functions are implemented in a synchronous way and do not make use of the `DEM_NUMBER_PENDING` respectively `DEM_FILTERED_PENDING` return values: `Dem_J1939DcmGetNumberOfFilteredDTC()`, `Dem_J1939DcmGetNextFilteredDTC()`, `Dem_J1939DcmGetNextFreezeFrame()`. This allows for a more compact and ROM-saving implementation.

- ▶ Support of `uint16` type for `Dem_DTCOriginType`

Description:

The ClearDTC operation of C/S-interface CddIf and APIs `Dem_GetNumberOfEventMemoryEntries()` are implemented according to AUTOSAR R4.0/R4.2, with the exception of the `DTCOrigin` parameter type. The `Dem_DTCOriginType` has been adapted from `uint8` to `unit16` type.

- ▶ Support of user-defined memory identifier for accessing the secondary event memory

Description:



The Dem module was extended with support to select the secondary event memory with a configurable user-defined memory identifier.

This feature can be enabled with configuration parameter `DemUserDefMemoryId`.

If configured, and the `DTCOrigin` value of a service request matches with `DemUserDefMemoryId + 0x100`, the `DTCOrigin` value of the request is mapped to `DEM_DTC_ORIGIN_SECONDARY_MEMORY`.

This feature cannot be used for operations on C/S interfaces, where `DTCOrigin` is used as a port defined argument (operations `GetEventMemoryOverflow` and `GetNumberOfEventMemoryEntries` of C/S interface `EvMemOverflowIndication`).

► Configuration of the test failed status bit storage for individual events

Description:

The Dem module was extended with an optional configuration parameter `DemStatusBitStorageTestFailedPerEvent` to provide a more granular control of the test failed status bit storage.

This feature provides the capability to configure `DemStatusBitStorageTestFailed` per event. If configured, the value of `DemStatusBitStorageTestFailedPerEvent` is preferred over the general `DemStatusBitStorageTestFailed` to determine whether the `TestFailed` status for an event is stored volatile or non-volatile.

► Priority of indicator status applied by the API `Dem_GetIndicatorStatus()`

Description:

If an indicator is activated in multiple modes (e.g. OBD and J1939) at the same time, the API `Dem_GetIndicatorStatus()` returns the status considering the following priority:

- `DEM_INDICATOR_FAST_FLASH` (highest priority)
 - `DEM_INDICATOR_SLOW_FLASH`
 - `DEM_INDICATOR_BLINK_CONT`
 - `DEM_INDICATOR_BLINKING`
 - `DEM_INDICATOR_CONTINUOUS`
 - `DEM_INDICATOR_OFF` (lowest priority)
- Support for intermediate DTC status storage

Description:

To preserve the result of the last diagnosis in case of a power failure or hard reset, the Dem module was extended with support for the intermediate DTC status storage.



If configuration parameter `DemIntermediateNvStorageOfDTCStatus` is enabled, the Dem stores the DTC status in the NvRAM whenever the immediate storage of the event related data is triggered, in addition to the usual storage at shutdown.

3.3.2.4. Deviations

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

- ▶ `DemDataElementClass` type `DemExternalSRDataElementClass` is supported according to R4.3.0

Description:

The Dem does not support the configuration structure `DemDataElementClass` for type `DemExternalSRDataElementClass` according to AUTOSAR R4.0.3. `DemExternalSRDataElementClass` is implemented according to AUTOSAR R4.3.0. Therefore, R4.0.3 configuration parameter `DemDataElementInstanceRef` was removed. Additionally, an important difference is that the size of data elements is configured in bytes instead of bits. This schema adaptation does not conform to [ecuc_sws_6007] and [ecuc_sws_6008].

Rationale:

Not fully specified in Dem SWS R4.0.3.

Requirements:

`Dem615_Conf`, `Dem770_Conf`, `SWS_Dem_00903` (with respect to "unused bits")

- ▶ DTC suppression support according to R4.3.0

Description:

The DTC suppression feature (API, C/S interface, configuration) is implemented according to AUTOSAR R4.3.0. Therefore, the configuration parameter that controls the availability of this feature is `DemSuppressionSupport` of type `EcucEnumerationParamDef`, instead of R4.0.3 `DemDTCSuppressionSupport`. This schema adaptation does not conform to [ecuc_sws_6007] and [ecuc_sws_6008].

Requirements:

`Dem780_Conf`

- ▶ The *Event combination* functionality is implemented fully compatible according to AUTOSAR Specification of Diagnostic Event Manager R4.0.3 schema.

Description:



Range of parameter `DemEventCombinationSupport` is kept compatible with AUTOSAR Specification of Diagnostic Event Manager R4.0.3 schema. The following conditions are set by AUTOSAR Specification of Diagnostic Event Manager R4.2.2:

- ▶ There is no need for a dedicated `EventId`, because no combined (parent/master) event will be configured.
- ▶ No combined event (parent/master event) is configured by the user, therefore the freeze frame and extended data will be configured commonly for the events involved in combination.
- ▶ If a combined DTC is displaced, also UDS status bit 2 (PDTC) and 5 (TFSLC) of the involved events can be reset to 0.

`DemEvtCmbCommonParamMaster` configuration parameter defines the master configuration event for combined DTCs.

Requirements:

Dem024, Dem537, Dem538

- ▶ *Event combination* on retrieval is not supported

Description:

Event combination on retrieval (event combination type 2) is not supported.

Requirements:

Dem539, Dem540, Dem541, Dem542

- ▶ Limited support for occurrence counter processing

Description:

Occurrence counter triggering after successful fault confirmation (by setting the value `DEM_PROCESS_OC-CCTR_CDTc` for configuration parameter `DemOccurrenceCounterProcessing`) is not supported.

Requirements:

Dem580, Dem767_Conf

- ▶ No Dlt interaction (reference to product description: ASCPD-68)

Description:

The APIs `Dem_DltGetMostRecentFreezeFrameRecordData()`, `Dem_Dlt GetAllExtended-DataRecords()`, and the notification `Dlt_DemTriggerOnEventStatus()` are not supported.

Requirements:

Dem517, Dem632, Dem633, Dem634, Dem635, Dem636, Dem637, Dem255, Dem718_Conf



- ▶ No *Storage conditions* support (reference to product description: ASCPD-36)

Description:

The feature *Storage conditions* is not supported.

Requirements:

Dem451, Dem543, Dem453, Dem455, Dem458, Dem591, Dem459, Dem556, Dem727_Conf, Dem728_-Conf, Dem730_Conf, Dem731_Conf, Dem773_Conf, Dem768_Conf, Dem769_Conf

- ▶ Extension of `DemFreezeFrameCapture` and `DemExtendedDataCapture` with additional trigger point `DEM_TRIGGER_ON_FDC_THRESHOLD`

Description:

The trigger point `DEM_TRIGGER_ON_FDC_THRESHOLD` (value for configuration parameter `DemFreezeFrameCapture` and `DemExtendedDataCapture`) triggers the collection of freeze frame/extended data for the initial event memory entry. See AUTOSAR Bugzilla RFC http://www.autosar.org/bugzilla/show_bug.cgi?id=50073.

Requirements:

Dem663_Conf, Dem672_Conf

- ▶ External aging is not supported (reference to product description: ASCPD-36)

Description:

External aging via `Dem_SetAgingCycleCounterValue()` (by setting the value `DEM_PROCESS_AGINGCTR_EXTERN` for configuration parameter `DemAgingCycleCounterProcessing`) is not supported.

Rationale:

This feature was removed in AUTOSAR R4.3.0 (refer to Bugzilla RfC #http://www.autosar.org/bugzilla/show_bug.cgi?id=59615).

Requirements:

Dem472, Dem644, Dem647, Dem019, Dem488, Dem491, Dem639, Dem640, Dem641, Dem642, Dem555, Dem603_Conf

- ▶ Extension of `DemInternalDataElement` range

Description:

`DemInternalDataElement` is extended with new status indicator bits, fault detection counters, operation cycle counters, event priority, aging counter up/down, event id, cycles since last failed, and cycles since first failed.



Requirements:

Dem616_Conf, ECUC_Dem_00616

- ▶ Multiple event destinations not supported

Description:

Each event can only be associated to one event memory.

The multiplicity of the related container `DemEventDestination` is limited from 0..4 to 1..1.

Rationale:

The behavior is undefined as per Dem SWS (see also AUTOSAR Bugzilla RfCs http://www.autosar.org/bugzilla/show_bug.cgi?id=41418 and http://www.autosar.org/bugzilla/show_bug.cgi?id=53633).

Requirements:

Dem658_Conf

- ▶ No support for *Init monitor for function* callback

Description:

The related parameter `DemCallbackInitMForFFnc` is not used.

Rationale:

The behavior is undefined as per Dem SWS (see also AUTOSAR Bugzilla RfC http://www.autosar.org/bugzilla/show_bug.cgi?id=35637).

Requirements:

Dem335, Dem049, Dem258, Dem614, Dem600_Conf, Dem633_Conf

- ▶ Prioritization of `InitMonitorReason` for multiple triggers of `DemInitMonitorForEvent` occurring simultaneously is not implemented.

Description:

The simultaneous occurrence condition is not fully specified and an asynchronous `DemInitMonitorForEvent` callback triggering would be needed to implement the prioritization. This would require an additional queue for storing the events to be processed in the next `Dem_MainFunction()`.

Requirements:

SWS_Dem_01046

- ▶ No support for `DemMaxNumberPrestoredFF`

**Description:**

Freeze frame pre-storage is supported but the optimization regarding the configuration for the maximum number of pre-stored freeze frames is not available.

Requirements:

Dem692_Conf

- ▶ Multiplicity of `DemCallbackGetFDC`

Description:

The multiplicity of the configuration container `DemCallbackGetFDC` was extended from 1..1 to 0..1. This schema adaptation does not conform to [ecuc_sws_6008].

Rationale:

The multiplicity of 0 is required to allow monitor-internal debounced events without fault detection counter support, i.e. neither RTE nor C `GetFaultDetectionCounter` callback (see also AUTOSAR Bugzilla RfC http://www.autosar.org/bugzilla/show_bug.cgi?id=44038).

Requirements:

Dem630_Conf

- ▶ 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 startup. 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:

Dem124

- ▶ Tracing of variables is not supported via AUTOSAR Debugging

Description:

Dem does not provide support for tracing global variables.



Requirements:

Dem519, Dem520, Dem521, Dem522

- ▶ External operation cycle counter is not supported

Description:

- ▶ Setting of the external operation cycle counter value via API-function `Dem_SetOperationCycleCountValue()` is not supported. It is provided as stub.
- ▶ The configuration parameter `DemOperationCycleProcessing` does not support `DEM_PROCESS_OPCYC_COUNTER`.

Rationale:

This API will be removed in AUTOSAR R4.1 (refer to Bugzilla RfC #http://www.autosar.org/bugzilla/show_bug.cgi?id=48567).

Requirements:

Dem485, Dem486, Dem487, Dem019, Dem488, Dem553, Dem783_Conf

- ▶ Reset PDTC bit as per ISO 14229-1

Description:

The `PendingDTC` bit is reset as specified in the ISO 14229-1 (UDS) standard.

Rationale:

The Dem must conform to the ISO 14229-1 standard. This bug is fixed in the AUTOSAR R4.1 (see also AUTOSAR Bugzilla RfC http://www.autosar.org/bugzilla/show_bug.cgi?id=45036).

Requirements:

Dem390

- ▶ Updated jump behavior to allow jump up and down in place of plain jump

Description:

Each reporting of a pre-failed/pre-passed value while the current debounce counter value is smaller/greater than the `DemDebounceCounterJumpUpValue`/`DemDebounceCounterJumpDownValue` shall first reset the debounce counter to `DemDebounceCounterJumpUpValue`/`DemDebounceCounterJumpDownValue` before performing the pre-failed/pre-passed debounce event.

Rationale:



The Dem must conform to the ISO 14229-1 standard. This is clarified in the AUTOSAR R4.1 (see AUTOSAR Bugzilla RfC http://www.autosar.org/bugzilla/show_bug.cgi?id=52969). The jump behavior is implemented to conform to requirements SWS_Dem_00423 and SWS_Dem_00425 from AUTOSAR R4.2.1.

Requirements:

Dem423, Dem425

- ▶ No support for configuration parameter `DemTriggerMonitorInitBeforeClearOk`

Description:

Configuration of monitor re-initialization to be triggered before or after the Dem module returns `DEM_CLEAR_OK` is limited to perform triggering only before returning `DEM_CLEAR_OK`.

Requirements:

Dem573, Dem765_Conf

- ▶ Development error `DEM_E_PARAM_LENGTH` is not supported

Description:

Wrong values of length parameters are handled via the development error `DEM_E_PARAM_DATA`.

Requirements:

Dem173

- ▶ Multiplicity of `DemFreezeFrameClass` and `DemDataElementClass`

Description:

The multiplicity of the configuration container `DemFreezeFrameClass` and `DemDataElementClass` was extended from 0..255 to 0..65535. This schema adaptation does not conform to [ecuc_sws_6008].

Rationale:

Due to enhancement "Enhanced multiplicity of `DemFreezeFrameClass` and `DemDataElementClass`", the multiplicity of `DemFreezeFrameClass` and `DemDataElementClass` is extended to 0..65535. See AUTOSAR Bugzilla RFCS https://www.autosar.org/bugzilla/show_bug.cgi?id=61653 and https://www.autosar.org/bugzilla/show_bug.cgi?id=55110.

Requirements:

Dem673_Conf, Dem610_Conf

- ▶ Multiplicity of `DemDidClass`

Description:



The multiplicity of the configuration container `DemDidClass` was extended from 0..255 to 0..65534. This schema adaptation does not conform to [ecuc_sws_6008].

Rationale:

Due to enhancement "Enhanced multiplicity of `DemDidClass`", the multiplicity of `DemDidClass` is extended to 0..65534. Value 65535 is used for internal processing (calibration unused value). See AUTOSAR Bugzilla RFC https://www.autosar.org/bugzilla/show_bug.cgi?id=55110.

Requirements:

`Dem706_Conf`

- ▶ Unlimited MIL activation configuration

Description:

The MIL configuration is not limited to the dedicated activation with the CDTC transition to 1. However, warning indicators (including MIL) are activated synchronously (according to Dem504), while the CDTC bit is updated via the `Dem_MainFunction()` asynchronously according to Dem379.

Rationale:

The deviated requirement is removed in AUTOSAR R4.2.

Requirements:

`Dem544`

- ▶ Reduced set of PIDs calculated Dem internally (reference to product description: ASCPD-135)

Description:

The following Dem internal calculated PIDs are not supported:

- ▶ PID \$4D - time with MIL On (2 byte)
- ▶ PID \$4E - time since last fault clear (2 byte)

The parameters listed below are not supported in the current implementation:

- ▶ `DemOBDInputAcceleratorPaddleInformation`
- ▶ `DemOBDInputAmbientTemperature`
- ▶ `DemOBDInputDistanceInformation`
- ▶ `DemOBDInputEngineSpeed`
- ▶ `DemOBDInputEngineTemperature`
- ▶ `DemOBDInputProgrammingEvent`



Requirements:

Dem347, Dem304, Dem377, Dem378, Dem323, Dem324, Dem627, Dem704_Conf, Dem763_Conf, Dem762_Conf, Dem761_Conf, Dem759_Conf, Dem757_Conf, Dem772_Conf, Dem760_Conf, Dem602_Conf, Dem756_Conf

- ▶ No support for In Use Monitoring Performance Ratio (IUMPR)

Description:

Dem does not support IUMPR, therefore the following configuration parameters are not supported:

- ▶ DemRatioId
- ▶ DemIUMPRGroup
- ▶ DemRatioIdType
- ▶ DemDiagnosticEventRef
- ▶ DemFunctionIdRef
- ▶ DemSecondaryFunctionIdRef

Also the related API-functions Dem_DcmGetInfoTypeValue08() and Dem_DcmGetInfoTypeValue0B() are not supported. They are provided as stubs and their return values are always E_NOT_OK.

Requirements:

Dem359, Dem360, Dem296, Dem361, Dem362, Dem297, Dem308, Dem299, Dem302, Dem313, Dem314, Dem315, Dem734_Conf, Dem737_Conf, Dem741_Conf, Dem735_Conf, Dem736_Conf, Dem782_Conf, SWS_Dem_01101, SWS_Dem_00316, SWS_Dem_00317, SWS_Dem_00298, SWS_Dem_00357, SWS_Dem_00358, SWS_Dem_00912, SWS_Dem_00913

- ▶ Fault Confirmation and Warning Indicator behavior

Description:

The Dem module provides a correction for Fault Confirmation and Warning Indicator handling according to OEM requirements. See ISO 14229-1:2006/2013 Figure D.9, RFC 70785, and RFC 71313. Now the following configuration parameters define the number of TestFailedThisOperationCycle status bit transitions (i.e. tested and failed cycles) for the respective DTC status bits to be activated:

- ▶ DemEventFailureCycleCounterThreshold
- ▶ DemIndicatorFailureCycleCounterThreshold

This means, the handling for setting the Fault Confirmation and Warning Indicator Requested status bits represents the number of triggering points (defined by the value of the configuration parameters mentioned above) with FAILED qualification of the event.



Rationale:

AUTOSAR R4.0.3 describes different triggering points which present the changes of the failure cycle (and not the FAILED qualification). This would require an additional cycle to the value configured and therefore the meaning of the threshold values would differ. In order to implement the OEM requirements described above the Dem module deviates slightly from the requirements Dem529 with Dem391, and Dem504 with Dem395. These requirements describe the setting behavior of Fault Confirmation and WarningIndicator according to Figure D.4 in ISO 14229-1:2013 (which contradicts Figure D.9).

Requirements:

Dem529, Dem504

- ▶ Development error `DEM_E_UNINIT` is not supported in the APIs `Dem_PreInit()` and `Dem_Init()`

Description:

The Dem module provides a correction for APIs `Dem_PreInit()` and `Dem_Init()` where reporting of development error `DEM_E_UNINIT` is replaced by `DEM_E_WRONG_CONDITION`.

Rationale:

AUTOSAR defines in requirement Dem124, if development error detection is enabled and any instance calls any Dem API, before the Dem was fully initialized (not pre-initialized in case of Dem364), the Dem module shall set the error code `DEM_E_UNINIT`. This requirement is violated in the case of APIs `Dem_PreInit()` and `Dem_Init()` which directly control initialization process. Therefore, `DEM_E_WRONG_CONDITION` (see Dem518) is a more suitable development error code for these APIs.

Requirements:

Dem124, SWS_Dem_00124, Dem364

- ▶ The Dem interfaces used by Dcm are based on AUTOSAR R4.3.1 specification

Description:

The Dem interfaces called by the Dcm module for the processing of the OBD and UDS service requests are based on the AUTOSAR R4.3.1 specification, but with the following limitations:

- ▶ The APIs `Dem_DcmGetAvailableOBDMIDs()`, `Dem_DcmGetNumTIDsOfOBDMID()` and `Dem_DcmGetDTRData()`, used for the processing of OBD service \$06 request, are not supported and they are provided as stubs and will always return `E_NOT_OK`.
- ▶ The API `Dem_DcmReadDataOfPID91()`, used for fetching the value of PID\$91 in the OBD service \$01, is not supported. It is provided as stub and its return value is always `E_NOT_OK`.
- ▶ The API `Dem_GetDTCSeverityAvailabilityMask()`, used for the processing of UDS service 0x19 sub-function 0x42 (reportWWHOBDTCByMaskRecord), is not supported.



- ▶ The API `Dem_DisableDTCRecordUpdate()` does not protect the event related data of the selected DTC within the selected DTCOrigin from being deleted.
- ▶ The result of `Dem_ClearDTC()` does not depend on the result of `DemClearEventAllowed` callback.
- ▶ The Dem module does not support a special handling of the mirror memory. `DemClearEventAllowed` callback is called for mirror memory as well.
- ▶ If the DTC is configured to use event combination, the callback `DemClearEventAllowed` is only called for the master event as representative of the event combination and regardless of its availability status.
- ▶ Development error `DEM_E_WRONG_CONFIGURATION` is not reported if the function `Dem_SetDTCFilter()` is called with DTCFormat or DTCOrigin values which are not supported by configuration.
- ▶ The configuration schema is based on AUTOSAR R4.0.3 and therefore, the full AUTOSAR R4.3.1 compliant functionality is not offered. For example:
 - ▶ Dem does not support the below AUTOSAR R4.3.1 configuration parameters related to the APIs called by Dcm:
 - ▶ `DemClearDtcNotification`
 - ▶ `DemClearDTCLimitation`
 - ▶ `DemTriggerMonitorInitBeforeClearOk`
 - ▶ Dem does not support the below configuration parameters related to `DemClient` container:
 - ▶ `DemClientFunctionality`
 - ▶ `DemClientUsesRte`
 - ▶ `DemEventMemorySetRef`
 - ▶ The configuration parameter `DemDTCFunctionalUnit` is not optional.
 - ▶ Lower 5 bits of `SeverityMask` (see [SWS_Dem_00937]) cannot be configured.
- ▶ The C/S interface `ClearDTC` is not supported.

Rationale:

The APIs specified in the AUTOSAR R4.3.1 are necessary to allow parallel access to the event related data and fault memory to service multiple diagnostic requests at the same time.

Requirements:

From AUTOSAR R4.0.3: Dem327, Dem596, Dem597, Dem623, Dem624, Dem035, Dem626, Dem079, Dem242, Dem080, Dem243, Dem057, Dem061, Dem216, Dem217, Dem228, Dem229, Dem287, Dem288, Dem513, Dem208, Dem214, Dem215, Dem227, Dem281, Dem013, Dem230, Dem060, Dem213, Dem059, Dem171, Dem172, Dem270, Dem648, Dem212, Dem233, Dem234, Dem232, Dem594, Dem221, Dem218, Dem593, Dem209, Dem224, Dem595, Dem226, Dem225, Dem609, Dem547, Dem560, Dem629, Dem071, Dem576, Dem630, Dem074, Dem075, Dem631, Dem076,



Dem240, Dem238, Dem239, Dem236, Dem574, Dem070, Dem235, Dem575, Dem316, Dem317, Dem298, Dem357, Dem358

From AUTOSAR R4.1.3: SWS_Dem_00298

From AUTOSAR R4.3.1: SWS_Dem_00270, SWS_Dem_01181, SWS_Dem_01182, SWS_Dem_-01295, SWS_Dem_01202, SWS_Dem_01205, SWS_Dem_01206, SWS_Dem_01066, SWS_Dem_-01067, SWS_Dem_01240, SWS_Dem_01241, SWS_Dem_00670, SWS_Dem_00573, SWS_Dem_-91002, SWS_Dem_00666, SWS_Dem_01009, SWS_Dem_01294, SWS_Dem_01140, SWS_Dem_-00766, SWS_Dem_00767, SWS_Dem_00768, SWS_Dem_00316, SWS_Dem_00317, SWS_Dem_-01187, SWS_Dem_01168, SWS_Dem_01177, SWS_Dem_00760, SWS_Dem_00761, SWS_Dem_-00762, SWS_Dem_00759, SWS_Dem_01301, SWS_Dem_01101, SWS_Dem_01108, SWS_Dem_-00293, ECUC_Dem_00790, ECUC_Dem_00925, ECUC_Dem_00943, ECUC_Dem_00933, ECUC_Dem_00942

- ▶ Base type of `Dem_DTCOriginType` changed to `uint16`.

Description:

The base type of `Dem_DTCOriginType` has been changed from `uint8` to `uint16` according to AUTOSAR 4.3.1. specification.

Rationale:

The Dem-Dcm interface implementation is based on AUTOSAR 4.3.1 specification. Consequently, the global `Dem_DTCOriginType` type, which is used in SW-C, CDD and Dem-Dcm APIs, is adapted from base type `unit8` to base type `uint16`. For SW-C/CDD interfaces, the Dem interpretation of `DTCOrigin` values [0x0001..0x0004] is not changed i.e., interpretation is still according to AUTOSAR 4.0/4.2. In addition, if `DemUserDefMemoryId` is configured, the secondary event memory can be accessed by a `DTCOrigin` value, equal to the configured `DemUserDefMemoryId` + 0x0100.

- ▶ Upgrade of event memory overflow indication.

Description:

The API `Dem_GetEventMemoryOverflow()` has the type for the input parameter `DTCOrigin` upgraded to an `uint16`.

Rationale:

The functionality event memory overflow indication defined by the C API `Dem_GetEventMemoryOverflow()` has the `DTCOrigin` information represented by the `Dem_DTCOriginType` definition according to AUTOSAR R4.3. The `Dem_DTCOriginType` has been adapted from `uint8` to `unit16` type for supporting the extended range of user defined memory, as defined by AUTOSAR R4.3. The Rte interface is not affected since the argument for `DTCOrigin` is a `PortDefinedArgumentValue` for `GetEventMemoryOverflow` operation and the C-level API `Dem_GetEventMemoryOverflow()` is also not impacted since a larger type is provided.



Requirements:

Dem559

- ▶ No support for the APIs `GetCycleQualified()` and `RestartOperationCycle()`.

Description:

The operations `GetCycleQualified()` and `RestartOperationCycle()` of the interface `OperationCycle` and the corresponding APIs are not implemented. The restart of a driving cycle is done with the function `Dem_SetOperationCycleState()`, which is kept for backwards compatibility.

Rationale:

These APIs were not requested by the customer.

Requirements:

SWS_Dem_00601

- ▶ J1939 sub-features

Description:

The following J1939 sub-features are not implemented: Monitor Performance Ratio (DM20), Diagnostic Readiness 2 (DM21), Diagnostic Readiness 3 (DM26), Expanded Freeze Frame (DM25), DTC to Lamp Association (DM31) and callback on every J1939 DTC status change configured in `DemCallbackJ1939DTCStatusChanged`.

The following APIs are not supported (they are provided as stubs only):

- ▶ `Dem_J1939DcmSetRatioFilter()`
- ▶ `Dem_J1939DcmGetNextFilteredRatio()`
- ▶ `Dem_J1939DcmReadDiagnosticReadiness2()`
- ▶ `Dem_J1939DcmReadDiagnosticReadiness3()`
- ▶ `Dem_J1939DcmFirstDTCwithLampStatus()`
- ▶ `Dem_J1939DcmGetNextDTCwithLampStatus()`
- ▶ `Dem_J1939DcmGetNextSPNInFreezeFrame()`

Requirements:

SWS_Dem_00880, SWS_Dem_00881, SWS_Dem_00882, SWS_Dem_00883, SWS_Dem_00884, SWS_Dem_00885, SWS_Dem_00886, SWS_Dem_00887, SWS_Dem_00888, SWS_Dem_00889, SWS_Dem_00890, SWS_Dem_00891, SWS_Dem_00892, SWS_Dem_00893, SWS_Dem_00894, SWS_Dem_00895, SWS_Dem_00896, SWS_Dem_00897, SWS_Dem_00898, SWS_Dem_00904, SWS_Dem_00905, SWS_Dem_00906, SWS_Dem_00907, SWS_Dem_00910, SWS_Dem_00911,



SWS_Dem_00912, SWS_Dem_00913, SWS_Dem_00974, SWS_Dem_00975, ECUC_Dem_00834, SWS_Dem_00983, SWS_Dem_00770, SWS_Dem_00979, SWS_Dem_00980, SWS_Dem_00981, SWS_Dem_00987

- ▶ Dem configuration structure for J1939 extension is deviating from AUTOSAR R4.2.2

Description:

The J1939 Dem extension is implemented as per AUTOSAR R4.2.2 whereas the Dem configuration schema is originally based on AUTOSAR R4.0.3. The following deviations are applied due to compatibility with AUTOSAR R4.0.3 and to simplify the configuration:

- ▶ The Dem J1939 feature is enabled via the configuration switch `DemJ1939Support`. The container `DemGeneralJ1939` is always present. This adaptation does not conform to [SWS_Dem_00845] and [ECUC_Dem_00864].
- ▶ Parameters `DemAmberWarningLampIndicatorRef` [ECUC_Dem_00821], `DemProtectLampIndicatorRef` [ECUC_Dem_00822] and `DemRedStopLampIndicatorRef` [ECUC_Dem_00820] are placed in container `DemGeneral` instead of `DemGeneralJ1939`.
- ▶ Container `DemJ1939Node` [ECUC_Dem_00817] and reference `DemJ1939DTC_J1939NodeRef` [ECUC_Dem_00819] are not implemented. Instead parameter `DemJ1939NodeRef` is available in container `DemDTCClass` providing the functionality of the AUTOSAR parameter `DemJ1939NmNodeRef` [ECUC_Dem_00818]. This implementation allows to assign one DTC to one J1939NmNode.
- ▶ `DemJ1939DTCValue` [ECUC_Dem_00892] is part of container `DemDTCClass` due to compatibility with AUTOSAR R4.0.3.
- ▶ `DemJ1939FreezeFrameClassRef` [ECUC_Dem_00835] is part of container `DemDTCClass` due to compatibility with AUTOSAR R4.0.3.

Requirements:

SWS_Dem_00845, ECUC_Dem_00864, ECUC_Dem_00817, ECUC_Dem_00818, ECUC_Dem_00819, ECUC_Dem_00820, ECUC_Dem_00821, ECUC_Dem_00822, ECUC_Dem_00892, ECUC_Dem_00835

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.

- ▶ No support for link time and post-build time configurations (reference to product description: ASCPD-77)

Description:

The Dem module can only be configured at pre-compile time. Link time and post-build time configurations are not supported.



Rationale:

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

Requirements:

Dem108, Dem268

- ▶ Implementation-specific parameter range and container multiplicity limitations

Description:

The configuration-scheme as specified in `AUTOSAR_EcucParamDef.arxml` version 3.1.2 was changed as follows:

- ▶ Parameter `DemBswErrorBufferSize`: range limited and corrected from 0..255 to 1..255, because the error queue cannot be switched off completely.
- ▶ Parameter `DemImmediateNvStorageLimit`: range limited and corrected from 1..255 to 1..254.
- ▶ Parameter `DemTaskTime`: range limited and corrected from 0..0.1 to 0.001..0.1, because 0 is no valid task time (refer to RfC #50351).
- ▶ Container `DemOperationCycle`: multiplicity limited and corrected from 1..256 to 1..255, because the API `Dem_SetOperationCycleState()` accepts only 8-bit wide IDs.
- ▶ Parameter `DemEnableConditionID`: range limited and corrected from 0..255 to 0..254, because the parent container `DemEnableCondition` multiplicity is limited to 255. Also it is not possible to configure a value of 255 for the `DemEnableConditionID` due to the additional zero-based and consecutive constraint.
- ▶ Parameter `DemIndicatorID`: range limited and corrected from 0..255 to 0..254, because the parent container `DemIndicator` multiplicity is limited to 255. Also it is not possible to configure a value of 255 for the `DemIndicatorID` due to the additional zero-based and consecutive constraint.
- ▶ Parameter `DemPidIdentifier`: range limited and corrected from 0..255 to 0..254, because the value 255 is used to disable PIDs in context of calibration.
- ▶ Container `DemGroupOfDTC`: multiplicity limited from 0..255 to 1..30.
- ▶ Container `DemConfigSet`: multiplicity limited to 1..1, because only pre-compile time configuration is supported.
- ▶ Container `DemEventParameter`: multiplicity limited from 1..65535 to 1..8190.
- ▶ Parameter `DemEventId`: range limited from 1..65535 to 1..8190.
- ▶ Parameter `DemMaxNumberFreezeFrameRecords`: range limited and corrected from 0..255 to 1..255, because the existence of FF data is controlled via `DemFreezeFrameClassRef` multiplicity.
- ▶ Parameter `DemAgingCycleCounterThreshold`: range limited from 1..256 to 1..255.



- ▶ **Parameter** `DemAgingCycleCounterThreshold`: multiplicity limited and corrected from 0..1 to 1..-1, due to the dependency to the parameter `DemAgingAllowed`.
- ▶ **Parameter** `DemEventFailureCycleCounterThreshold`: range limited from 1..256 to 1..254
- ▶ **Parameter** `DemIndicatorFailureCycleCounterThreshold`: range limited from 0..255 to 1..255 (refer to the deviation "Fault Confirmation and Warning Indicator behavior" and RfC #71313)
- ▶ **Parameter** `DemDebounceCounterDecrementStepSize`: range limited and corrected from 0..-32768 to 1..32768, because step-size zero does not lead to any change (refer to RfC #50351).
- ▶ **Parameter** `DemExtendedDataCapture`: multiplicity limited and corrected from 0..1 to 1..1 (refer to RfC #59376)
- ▶ **Parameter** `DemFreezeFrameCapture`: multiplicity limited and corrected from 0..1 to 1..1 (refer to RfC #59376)
- ▶ **Parameter** `DemDebounceCounterFailedThreshold`: range limited and corrected from 0..32767 to 1..32767, because debounce counter start-value is not allowed as qualification threshold (refer to RfC #50351).
- ▶ **Parameter** `DemDebounceCounterIncrementStepSize`: range limited and corrected from 0..-32767 to 1..32767, because step-size zero does not lead to any change (refer to RfC #50351).
- ▶ **Parameter** `DemDebounceCounterPassedThreshold`: range limited and corrected from -32768..-0 to -32768..-1, because debounce counter start-value is not allowed as qualification threshold (refer to RfC #50351).
- ▶ **Parameters** `DemDebounceTimeFailedThreshold` and `DemDebounceTimePassedThreshold`: range limited and corrected from 0 to 0.001, because 0 is no valid threshold for time (refer to RfC #50351).
- ▶ **Container** `DemExtendedDataClass`: multiplicity limited from 0..unbound to 0..256.
- ▶ **Parameter** `DemExtendedDataRecordNumber`: range limited and corrected from 0..253 to 1..239, because 0x00 and 0xF0-0xFF is reserved according to ISO 14229-1.
- ▶ **Parameter** `DemEnableConditionGroup`: Multiplicity limited and corrected from 0..255 to 0..254, because the value of 255 is used for internal processing.
- ▶ **Parameter** `DemEventPriority`: range limited and corrected from 1..256 to 1..255, because the priority can be mapped to extended data.

Rationale:

Parameter limitations allow for more efficient implementation and solve SWS issues.

Requirements:

`Dem625_Conf`, `Dem715_Conf`, `Dem701_Conf`, `Dem654_Conf`, `Dem683_Conf`, `Dem679_Conf`, `Dem634_-Conf`, `Dem661_Conf`, `Dem659_Conf`, `Dem605_Conf`, `Dem623_Conf`, `Dem750_Conf,1`, `Dem753_Conf`,



Dem635_Conf, Dem618_Conf, Dem637_Conf, Dem636_Conf, Dem716_Conf, Dem717_Conf, Dem664_Conf, Dem666_Conf, Dem662_Conf

- ▶ Limitation on usage of Dem configuration parameter `DemPidDataElement` and API `Dem_DcmReadDataOfOBDFreezeFrame()`

Description:

If the order of data elements that are configured in `DemPidDataElement` differs from the configured order of data elements inside a PID (see configuration parameter `DcmDspPidDataPos`), the API `Dem_DcmReadDataOfOBDFreezeFrame()` returns wrong PID data elements.

Rationale:

The current implementation of API `Dem_DcmReadDataOfOBDFreezeFrame()` does not take into account the PID data element order that is configured in Dcm (`DcmDspPidDataPos`). Therefore, the order of data elements inside a PID of Dem that are configured in `DemPidDataElement` must accord with the configured order of data elements inside a PID of Dcm in `DcmDspPidDataPos`.

Requirements:

SWS_Dem_00597

- ▶ For AUTOSAR R4.2 service API: Function `Dem_ClearDTC()` is not re-entrant

Description:

According to AUTOSAR R4.2.1, requirement SWS_Dem_00665, `Dem_ClearDTC()` shall be re-entrant. The current implementation of this API is defined as "non re-entrant" according to AUTOSAR R4.0.3.

Rationale:

The behavior of this interface is not clearly specified for this case. RFC 64149 was created for clarification of this topic.

Requirements:

SWS_Dem_00665

- ▶ Usage of internal data elements is limited only for extended data records

Description:

The configuration parameters `DemPidDataElementClassRef`, `DemDidDataElementClassRef` and `DemSPNDataElementClassRef` do not support referencing to `DemInternalDataElementClass`. The `DemInternalDataElementClass` elements can only be referenced by an `DemExtendedDataRecordClass`.

Rationale:



There are no known use cases where PIDs, DIDs or SPNs need to contain data defined via `DemInternalDataElementClass` elements. Newer AUTOSAR versions also specify this constraint explicitly.

Requirements:

`Dem617_Conf`, `Dem733_Conf`, `ECUC_Dem_00832` and `Dem469`

- ▶ For AUTOSAR R4.2 service API: Function `Dem_<...>ClearDTC()` supports only `DEM_DTC_GROUP_ALL_DTCS` and `DEM_DTC_GROUP_EMISSION_REL_DTCS` of the DTC groups if the DTC format is set to `DEM_DTC_FORMAT_OBD`

Description:

According to AUTOSAR R4.2.1, requirement `SWS_Dem_00665`, `Dem_ClearDTC()` shall clear groups of DTCs for different DTC formats. The current implementation of this API prohibits the clearing of DTC groups with `DEM_DTC_FORMAT_OBD`, except for DTC group `DEM_DTC_GROUP_ALL_DTCS` (0xffffffff) and `DEM_DTC_GROUP_EMISSION_REL_DTCS` (0x0000000).

Rationale:

The behavior of this interface is not clearly specified for this case.

Requirements:

`SWS_Dem_00665`

- ▶ Functions `Dem_GetNextFreezeFrameData()`, `Dem_SetFreezeFrameRecordFilter()` and `Dem_GetNextFilteredRecord()` consider the global OBD freeze frame only for the DTC that triggered the storage

Description:

Only for the emission related DTC that stored the OBD freeze frame, the API `Dem_GetNextFreezeFrameData()` responds with the currently stored OBD freeze frame data, when record number 0x00 is requested.

For all other emission related DTCs, `Dem_GetNextFreezeFrameData()` returns `E_OK` with a size of 0, when record number 0x00 is requested. `Dem_GetNextFreezeFrameData()` is used, e.g., to respond to diagnostic service 0x19 0x04.

Similar behavior applies also to the APIs `Dem_SetFreezeFrameRecordFilter` and `Dem_GetNextFilteredRecord()` which are used to respond to diagnostic service 0x19 0x03.

Rationale:

If Dem is configured to support only one global OBD freeze frame via `DemFreezeFrameCapture`, then only one DTC can return the OBD freeze frame in function `Dem_GetNextFreezeFrameData()`. Even



if Dem is configured to support multiple OBD freeze frames, `Dem_GetNextFreezeFrameData()` will return the OBD freeze frame only for one DTC to have the same behavior.

In order to have a consistent response for service 0x19 subfunctions 0x03 and 0x04, the APIs `Dem_SetFreezeFrameRecordFilter()` and `Dem_GetNextFilteredRecord()` have the same limitation.

Note: Pids may be different between OBD service 0x02 and UDS service 0x19 0x04.

- ▶ OBD service 0x02: The Dcm arranges the DemPidDataElements inside the Pid.
- ▶ UDS service 0x19 0x04 record number 0x00: The Dcm passes the data directly from the Dem. The Dem cannot arrange the DemPidDataElements because the information is not available inside the Dem.

Requirements:

SWS_Dem_00576, SWS_Dem_00209

- ▶ Processing of unconfirmedDTC status during Dem pre-initialization

Description:

The unconfirmedDTC status for events reported via `Dem_ReportErrorStatus()` is not processed if the Dem module is pre-initialized, i.e., `Dem_Init()` was not called yet. The unconfirmedDTC status can be reached via debouncing.

Rationale:

At this time the event memory entries are not restored from NVRAM yet and therefore the related counters and status data (stored in the memory entries) cannot be processed together with the unconfirmedDTC status.

Requirements:

VCC_DEM_Table_12, VCC_DEM_Table_13

- ▶ Capture of `DemExtendedDataCapture` and/or `DemFreezeFrameCapture` on `DEM_TRIGGER_TEST-FAILED`

Description:

Synchronous data capturing (`DemExtendedDataCapture` and/or `DemFreezeFrameCapture` equals `DEM_TRIGGER_TESTFAILED`) is possible only for events reported via `Dem_SetEventStatus()`. The data capturing for events reported via `Dem_ReportErrorStatus()` is always done asynchronously in the main function regardless of configured `DemExtendedDataCapture` and `DemFreezeFrameCapture`.

Rationale:

The data for events reported via `Dem_ReportErrorStatus()` can only be captured asynchronously during the processing of the event queue in the main function.



Requirements:

Dem461, Dem467

- ▶ Function `Dem_<...>ClearDTC()` clears only the first event matching the requested single OBD DTC to be cleared, in case when more than one event are referring to same OBD DTC value

Description:

Any call of function `Dem_<...>ClearDTC()` with DTC which is used by several DTC classes and `DEM_DTC_FORMAT_OBD` as input parameters, only the first matching event to the DTC will be cleared.

Rationale:

As long as the behavior of this interface is not clearly specified, in case when identical OBD DTC is supported, the current implementation of this API prohibits the clearing of all events assigned to the same OBD DTC value.

Requirements:

Dem077

- ▶ Implementation-specific parameter range and container multiplicity limitations

Description:

The configuration-scheme as specified in `AUTOSAR_EcucParamDef.arxml` version 4.3.1 was changed as follows:

- ▶ Parameter `DemClientId`: range limited from 0..255 to 0..253.

Rationale:

Parameter limitations allow for more efficient implementation.

Requirements:

ECUC_Dem_00932

- ▶ Only one client at a time can control DTC setting

Description:

`Dem_EnableDTCSetting()` and `Dem_DisableDTCSetting()` cannot control the DTC setting if it is already disabled by a different client.

Rationale:

This is because the `DemEventMemorySet` functionality is missing which causes a call of `Dem_EnableDTCSetting()` or `Dem_DisableDTCSetting()` to affect all events associated with a DTC.



Requirements:

SWS_Dem_00242, SWS_Dem_00243

- ▶ All clients have access to the same event memory set.

Description:

Dem supports only one global event memory set. It is accessible by all configured clients and the Dem APIs do not work differently based on the ClientId.

Rationale:

The feature of multiple `DemEventMemorySets` and restricting access to a single `DemEventMemorySet` for a given client, as specified by AUTOSAR R4.3.1, is not supported currently.

Requirements:

SWS_Dem_00242, SWS_Dem_00243, SWS_Dem_00219, SWS_Dem_01263, SWS_Dem_00231, ECUC_Dem_00940, ECUC_Dem_00939

- ▶ Only events with UDS DTC are impacted by control DTC setting

Description:

`Dem_EnableDTCSetting()` and `Dem_DisableDTCSetting()` do not affect events configured with DTCs which are OBD only.

Rationale:

The current Dem implementation does not consider configurations with events which have only OBD DTCs assigned without having configured also UDS DTCs.

Requirements:

SWS_Dem_01290

- ▶ No support for the parallel call to the DTC selection related APIs for the same client

Description:

The Dem implementation does not support parallel call to the DTC selection related APIs for the same client. The relevant APIs are `Dem_SelectDTC()` and all the APIs that require a `Dem_SelectDTC()` according to SWS_Dem_01253. Consequently, these APIs do not support the return value `DEM_BUSY` as well.

Rationale:



Explicit exclusion of parallel call to the DTC selection related APIs for the same diagnostic client allows for a more efficient implementation. When considered together with the fact that the DTC selection related APIs are implemented as synchronous, such an exclusion does not lead to any loss of functionality in practice and becomes only a theoretical consideration.

Requirements:

SWS_Dem_01305, SWS_Dem_01306

- ▶ Impact of DTC suppression and event availability on DTC selection

Description:

If any operation based on the current DTC selection has already started, then any changes in the event available status and DTC suppression status have an impact on the functionality of the following APIs only after a new DTC selection. The impacted APIs are:

- ▶ Dem_GetStatusOfDTC
- Dem_GetSeverityOfDTC
- Dem_GetFunctionalUnitOfDTC
- Dem_SelectFreezeFrameData
- Dem_SelectExtendedDataRecord

Rationale:

This behavior is necessary to ensure that coherent results are returned by the above mentioned APIs, even if the event availability/DTC suppression status is changed in-between.

Requirements:

SWS_Dem_01100, SWS_Dem_01101

- ▶ Only one client at a time can control DTC record update

Description:

`Dem_DisableDTCRecordUpdate()` and `Dem_EnableDTCRecordUpdate()` cannot control the DTC record update if it is already disabled by a different client.

Rationale:

Parallel processing of UDS service 0x19, sub-functions 0x04, 0x06 is not supported and a diagnostic client can select only a single DTC. Therefore, there is no use case for locking multiple DTCs/event memory entries at the same time.



Requirements:

SWS_Dem_00233, SWS_Dem_00234

- ▶ No support for user defined memories

Description:

The user defined memories as specified by AUTOSAR versions R4.3.1 and newer are not supported. However, a possibility is provided to access the secondary memory as a user defined memory with the help of the configuration parameter `DemUserDefMemoryId`.

Rationale:

The event memory implementation is based on AUTOSAR R4.0.3 which does not support user defined memories.

Requirements:

SWS_Dem_00548

- ▶ Each event is assigned to exactly one event memory only

Description:

Each event is assigned to exactly one event memory only. Therefore, events without DTC event memory reference are not applicable in the EB solution.

Rationale:

There is no use case yet for supporting events which are not assigned to any event memory.

Requirements:

SWS_Dem_01249, ECUC_Dem_00941

- ▶ Clearing operation is locked until the result is returned to the requesting client or `Dem_SelectDTC` is called for the same client

Description:

The Dem does not release the locked clearing process until the result is returned to the requesting client or the same client calls `Dem_SelectDTC`, indicating the client is no longer interested in the result of the clearing operation.

Rationale:

Keeping the clearing process locked until the result is read or dropped allows a more efficient implementation.



Requirements:

SWS_Dem_01042

- ▶ Only one client at a time can filter stored DTCs

Description:

`Dem_DcmSetStoredDTCFilter()` and `Dem_DcmGetNextFilteredStoredDTC()` cannot be called by a client while a different client has already set the filter and has not yet finished reading all the filtered DTCs.

Rationale:

The APIs are used only by the UDS services and currently, the Dem implementation does not support UDS in parallel to UDS. Besides, these APIs are used together with the APIs related to DTC record update, which themselves do not support multiple clients in parallel. Therefore, there is no use case currently to support concurrent use of `Dem_DcmSetStoredDTCFilter()` and `Dem_DcmGetNextFilteredStoredDTC()` by multiple clients.

- ▶ DM05 "Diagnostic Readiness 1" is not supported for non-OBD ECUs.

Description:

According to AUTOSAR R4.2.2, requirement SWS_Dem_00909, `Dem_J1939DcmReadDiagnosticReadiness1` shall report the value five (5) for "OBD Compliance", when called for non-OBD ECUs. In the current implementation, `Dem_J1939DcmReadDiagnosticReadiness1` is not available for non-OBD ECUs.

Rationale:

There is no use case yet for supporting DM05 "Diagnostic Readiness 1" on non-OBD ECUs.

Requirements:

SWS_Dem_00909

3.3.2.6. Open-source software

The Dem module does not use open-source software.

3.3.3. FiM module release notes

- ▶ AUTOSAR R4.0 Rev 3



- ▶ AUTOSAR SWS document version: 2.2.0
- ▶ Module version: 2.5.4.B478580
- ▶ Supplier: Elektrobit Automotive GmbH

3.3.3.1. Change log

This chapter lists the changes between different versions.

Module version 2.5.4

2021-10-08

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

Module version 2.5.3

2021-06-25

- ▶ Refined descriptions of configuration parameter

Module version 2.5.2

2021-03-24

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

Module version 2.5.1

2020-12-18

- ▶ ASCFIM-323 Fixed known issue: Permission status of some FIDs is wrong for large configurations

Module version 2.5.0

2020-11-11

- ▶ Refined descriptions of FiM calibration data in FiM_Bswmd.arxml file
- ▶ Adapted the range check for the configuration parameters `FiMInputSumEventRef` and `FiMInhEventRef`



Module version 2.4.2

2020-09-25

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

Module version 2.4.1

2020-08-28

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

Module version 2.4.0

2020-06-19

- ▶ Change all NO_INIT memory sections to CLEARED

Module version 2.3.11

2020-05-22

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

Module version 2.3.10

2020-04-24

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

Module version 2.3.9

2020-01-24

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

Module version 2.3.8

2019-10-11

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

Module version 2.3.7

2019-09-06



- ▶ ASCFIM-275 Fixed known issue: BSWMD file references wrong namespace AUTOSAR_Std instead of AUTOSAR_Platform
- ▶ Added dedicated memory sections for calibratable data

Module version 2.3.6

2019-06-14

- ▶ ASCFIM-269 Fixed known issue: Importer for FiM calibration descriptor issues an error

Module version 2.3.5

2018-10-26

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

Module version 2.3.4

2018-09-28

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

Module version 2.3.3

2018-07-27

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

Module version 2.3.2

2018-06-22

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

Module version 2.3.1

2018-05-25

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

Module version 2.3.0

2018-03-16



- ▶ Improved module generation time
- ▶ Added memory sections for the FiM software component description

Module version 2.2.11

2018-02-23

- ▶ Implemented compliance to MISRA-C:2012

Module version 2.2.10

2017-10-09

- ▶ Optimized speed for configuration verifier checks

Module version 2.2.9

2017-07-07

- ▶ ASCFIM-217 Fixed known issue: Wrong generated type for FiM_IndexType

Module version 2.2.8

2017-03-31

Module version 2.2.7

2016-09-09

- ▶ Implemented support for Configuration Signature

Module version 2.2.6

2016-04-29

- ▶ Removed const qualifier from function parameter

Module version 2.2.5

2016-01-15



- ▶ Measurement support for selected arrays

Module version 2.2.4

2015-08-28

- ▶ ASCFIM-184 Fixed known issue: FiM does not generate without OBD license
- ▶ ASCFIM-183 Fixed known issue: Symbolic name of FIDs are generated without prefix

Module version 2.2.3

2015-07-24

- ▶ Added initialization of inhibitions to FiM_Init for extended FiM

Module version 2.2.2

2015-06-19

- ▶ Added support for inhibition mask FIM_FAILED_THIS_OP_CYCLE

Module version 2.2.1

2015-05-22

- ▶ ASCFIM-177 Fixed known issue: FiM_MainFunction can not be assigned to a task

Module version 2.2.0

2015-04-24

- ▶ ASCPD-180 Support for calibration of configuration parameter (message connection, condition event, forced release event, symptom bit and high prio inhibition event).

Module version 2.1.7

2015-03-27

- ▶ ASCPD-177 Extended inhibition triggers by vendor-specific options (for triggered mode).
- ▶ ASCPD-178 Support for high priority inhibitions.
- ▶ ASCPD-204 Support for FID messages.



- ▶ ASCPD-205 Support for FID symptom bit.
- ▶ ASCPD-206 Support for FID force release.
- ▶ ASCPD-207 Support for recovery.

Module version 2.1.6

2015-03-06

- ▶ Added requirement SWS_Fim_00097 which was introduced in ASR 4.2.1 and clarifies the behavior for negative return values of Dem_GetEventStatus.

Module version 2.1.5

2014-10-01

- ▶ ASCPD-176 Extended configuration scheme by vendor-specific parameters.
- ▶ ASCPD-177 Extended inhibition triggers by vendor-specific options (for polling mode).
- ▶ ASCPD-179 Implemented support for calibration of configuration parameters (Event Id and Inhibition Mask).

Module version 2.1.4

2014-04-25

- ▶ Improved implementation of exclusive areas
- ▶ ASCFIM-129 Fixed known issue: FiM does not compile if legacy symbolic names are disabled for the Dem module
- ▶ Introduced export of module configuration data structure for `Fim_Init()`

Module version 2.1.3

2013-10-11

- ▶ Added support for function tracing via AUTOSAR Debugging
- ▶ Updated Implementation Data Type of category `TYPE_REFERENCE` which referred to `DataConstr` which was not allowed

Module version 2.1.2

2013-06-14



- ▶ Updated include-structure according to AUTOSAR R4.0.3

Module version 2.1.1

2013-02-08

- ▶ Refined configuration parameter descriptions
- ▶ ASCFIM-87 Fixed known issue: Exclusive area function calls (`SchM_<Enter/Exit>_Fim_SCHM_FIM_-EXCLUSIVE_AREA_0`) in FiM are not according to BSW module name

Module version 2.1.0

2012-10-12

- ▶ Updated configuration schema to AUTOSAR R4.0.3
- ▶ Implemented status bits of `FiMInhInhibitionMask` according to AUTOSAR R4.0.3
- ▶ Changed the top-level structure of the software-component description in the ARXML files from `/AUTOSAR/FiM` to `/AUTOSAR_Fim`

Module version 2.0.1

2012-06-25

- ▶ ASCFIM-44 Fixed known issue: Timing-event for `FiM_MainFunction` can not be mapped if `FiM-CyclicEventEvaluation` is disabled
- ▶ ASCFIM-48 Fixed known issue: The FiM enters an endless loop if triggered with event IDs smaller than the smallest event ID configured in FiM

Module version 2.0.0

2012-03-15

- ▶ Updated to AUTOSAR 4.0

Module version 1.0.0

2010-10-08

- ▶ First release of EB tresos AutoCore Fim module



3.3.3.2. New features

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

3.3.3.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- ▶ FiM supports additional development errors

Description:

In addition to development error values defined by the FiM SWS R4.0.3 specification, the FiM implementation supports user-defined development error values:

- ▶ `FIM_E_EVENTID_OUT_OF_RANGE` (as per FiM SWS R3.1)

- ▶ Support for function tracing with EB macro tracing

Description:

FiM supports tracing of function entry and exit via EB Dbg macros. Tracing is supported by all external and selected internal APIs.

The entry macro takes all API parameters. The exit macro takes the return value and all API parameters

- ▶ FiM Extended Support

Description:

FiM supports vendor specific configuration parameters. The configuration switch `FiMGeneral/FiMExtendedSupport` enables support for the following parameters:

- ▶ `FiMGeneral/FiMExtendedLevel`
- ▶ `FiMGeneral/FiMMaxHighPrioEventToFidLinks`
- ▶ `FiMGeneral/FiMMaxHandledFidEventLinksPerScheduling`
- ▶ `FiMConfigSet/FiMFID/FiMSetSymptom`
- ▶ `FiMConfigSet/FiMFID/FiMMessageref`
- ▶ `FiMConfigSet/FiMFID/FiMConditionEventRef`
- ▶ `FiMConfigSet/FiMFID/FiMForcedReleaseEventRef`
- ▶ `FiMConfigSet/FiMFID/FiMRecoveryEventRef`
- ▶ `FiMConfigSet/FiMMessageref`
- ▶ `FiMConfigSet/FiMMessageref/FiMMessagerefId`



- ▶ `FiMConfigSet/FiMIInhibitionConfiguration/FiMHighPrioInhibition`

The configuration parameter `FiMIInhInhibitionMask` is extended by three additional vendor specific items:

- ▶ `FIM_LAST_UNCONFIRMED`
- ▶ `FIM_FAILED_THIS_OP_CYCLE`
- ▶ `FIM_UNCONFIRMED_THIS_OP_CYCLE`

NOTE
License required


A license is required for FiM Extended Support.

3.3.3.4. Deviations

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

- ▶ 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 startup. 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:

FIM060

- ▶ Tracing of variables is not supported via AUTOSAR Debugging

Description:

FiM does not provide support for tracing global variables.

Requirements:

FIM085, FIM086, FIM087, FIM088, FIM009



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.

- ▶ No support for link time and post-build time configurations (reference to product description: ASCPD-77)

Description:

The FiM module can be configured at pre-compile time. Link time and post-build time configurations are not supported.

Rationale:

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

Requirements:

FIM091, FIM030, FIM008

- ▶ Implementation-specific parameter range and container multiplicity limitations

Description:

The configuration scheme as specified in `AUTOSAR_MOD_ECUConfigurationParameters.arxml` version 4.0.3 was changed as follows:

- ▶ Container `FiMInhibitionConfiguration`: multiplicity limited and corrected from 0..unbound to 1..unbound
- ▶ Parameter `FiMFunctionId`: range limited and corrected from 0..65535 to 1..65535
- ▶ Reference `FiMInputSumEventRef`: multiplicity limited from 1..unbound to 1..1
- ▶ Reference `FiMIinhSumRef`: multiplicity limited from 1..unbound to 1..1

Rationale:

If there is no `FiMInhibitionConfiguration` then it indicates that there is no function to be inhibited. There will be no meaning of having the FiM module without `FiMInhibitionConfiguration`.

The parameter `FiMFunctionId` was changed, because of the requirement FIM011.

Limited multiplicities of references `FiMInputSumEventRef` and `FiMIinhSumRef` can be covered by creating one separate entry in `FiMEventSummary/FiMInhibitionConfiguration` per `FiMInputSumEventRef/FiMIinhSumRef` to be linked (like done in AUTOSAR R3.1).

Requirements:

FIM038_Conf, FIM085_Conf, FIM083_Conf, FIM102_Conf



- ▶ Limitation on maximum number of events per FID

Description:

Per FID maximal 255 events (single Dem events and individual Dem events from event summaries) can be referenced (by `FiMInhEventRef` and `FiMInhSumRef`).

Rationale:

Also in huge systems this number of Dem events per FID is typically fully sufficient. This limitation allows for more efficient implementation.

Requirements:

`FIM089_Conf`

- ▶ Limitation on used Dem Event IDs

Description:

If `FiMEventUpdateTriggeredByDem = false`, only events that have an ID in the range of 1..8191 (1..0x1FFF) can be referenced by the FiM (by `FiMInputSumEventRef` and `FiMInhEventRef`).

Rationale:

Also in huge systems this number of Dem events is typically fully sufficient. This limitation allows for more efficient implementation.

Requirements:

`FIM083_Conf`, `FIM100_Conf`

- ▶ Limitation on used function IDs

Description:

If `FiMEventUpdateTriggeredByDem = true`, only FIDs that have an ID in the range of 1..8191 (1..0x1FFF) can be referenced by the FiM (by `FiMFunctionId`).

Rationale:

Also in huge systems this number of FIDs is typically fully sufficient. This limitation allows for more efficient implementation.

Requirements:

`FIM085_Conf`



3.3.3.6. Open-source software

The `FiM` module does not use open-source software.



4. ACG8 Diagnostic Stack user guide

4.1. Overview

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

- ▶ Diagnostics Communication Manager (`Dcm`)
- ▶ Diagnostics Event Manager (`Dem`)
- ▶ Function Inhibition Manager (`FiM`)

The diagnostic stack modules `Dem` and `Dcm` provide options that support the **BSW Distribution** feature for use in a multi-core project environment. The general concepts related to multi-core and BSW distribution in AUTOSAR, and the support provided in EB tresos AutoCore Generic are described in chapter *Multi-core support* in the EB tresos AutoCore Generic 8 documentation.

To understand the basic concepts of the diagnostic modules, see [Section 4.2, “Background information”](#).

For integration issues around the diagnostic stack, see [Section 4.3, “Integrating the diagnostic stack”](#).

For instructions on how to configure the modules, see:

- ▶ [Section 4.4, “Dcm module user guide”](#)
- ▶ [Section 4.5, “Dem module user guide”](#)
- ▶ [Section 4.6, “FiM module user guide”](#)

4.2. Background information

This chapter describes the basic concepts of the AUTOSAR diagnostic modules.

Additional background information is available in the module-specific user guides:

- ▶ for the `Dcm` module, see [Section 4.4.2, “Background information”](#)
- ▶ for the `Dem` module, see [Section 4.5.2, “Background information”](#)
- ▶ for the `FiM` module, see [Section 4.6.2, “Background information”](#)



4.2.1. Purpose of the diagnostic stack

The diagnostic stack modules supervise the proper functionality of the complete basic software stack as well as of the software components. The diagnostic stack also collects diagnostic data from software components and transmits it to the communication stack.

The setup of the diagnostic modules Dcm, Dem and Fim together with their interfaces is depicted in [Figure 4.1, “Diagnostic modules in production mode”](#).

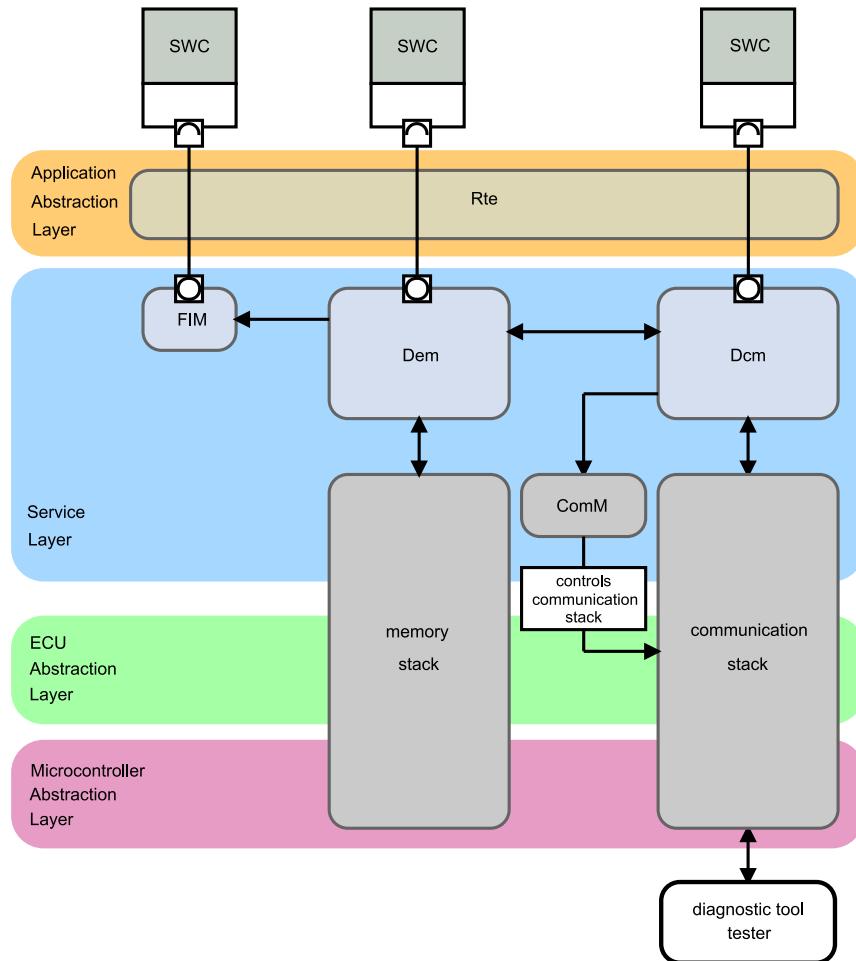


Figure 4.1. Diagnostic modules in production mode

4.2.1.1. Diagnostic stack interfaces

The interfaces depicted in the figure above are described here.



Dcm interfaces

Dcm/communication stack

To handle the service requests from the external tester, the `Dcm` uses the communication stack. The service requests are always routed via the `PduR`, the bus-specific transport protocol module and the bus-specific interface and driver module.

Dcm/ComM interface

To handle the diagnostic service requests, the `Dcm` has an interface towards the `ComM` module. Via the `ComM`, the `Dcm` requests the desired communication mode. For details about the `ComM` module, see the EB tresos AutoCore Generic documentation.

Dcm/software components

The interface towards the software components enables the `Dcm` to obtain the requested diagnostic data or to execute requested diagnostic commands.

Dcm/Dem

The `Dcm` has an interface towards the `Dem`. The `Dcm` receives data concerning diagnostic events from the `Dem`. The `Dcm` also sends commands to the `Dem` to clear or control the event settings.

Dem interfaces

Dem/memory stack

The `Dem` uses the memory stack to store event data non-volatile. For details about the memory stack, see the EB tresos AutoCore Generic Memory Stack documentation.

Dem/FiM

The `Dem` informs the `FiM` about changes of the event status. The event status reflects a malfunction. As a result, the `FiM` inhibits the execution of the software components which are dependent on this malfunction.

Dem/software components

The `Dem` has an interface towards the software components to update and/or retrieve events status information.

FiM interfaces

FiM/Dem

The `Dem` is in charge of handling detected malfunctions denoted as events and reported by monitoring functions. The `Dem` informs and updates the `FiM` upon changes of the event status in order to stop or release functionalities according to assigned dependencies.

FiM/software components

Software components (SWC) with function identifier (FID) interface query for permission to execute functionality identified by an FID at the `FiM`. The FIDs have to be provided by the software components.

4.2.1.2. Use case example

The following example describes the interactions of the diagnostic modules.



If the CAN bus monitoring software detects a short circuit between CAN lines, the software reports this event in the form of an event ID and event status to the `Dem`. Usually, an event ID is mapped to a specific Diagnostic Trouble Code (DTC). Various event IDs could be mapped to the same DTC, e.g., if all of the event IDs have the same repair procedure. Apart from the event information, the CAN bus monitoring software might report various parameters, e.g. the time when the error occurred. Some DTCs require additional information, e.g. the ECU temperature at the time of the error. This information can be defined and configured as freeze frame information or extended data in the `Dem`.

If so configured, the `Dem` module stores the detected diagnostic event in non-volatile memory via the `NvM` module. In this way, the diagnostic event is not deleted when the engine is turned off.

Later on, at the garage, the mechanic can read out the error memory via a tester tool. The tester tool sends a diagnostic request to the `Dcm`. The `Dcm` collects the necessary data (the DTC and freeze frame/extended data) from the `Dem` and generates the appropriate answer. Based on the DTC, the mechanic can retrieve the information that at some time the CAN had a short circuit (bus-off) event. The freeze frame information and extended data provide the mechanic with additional data to identify the error cause.

4.2.2. Error notification

AUTOSAR distinguishes the following two modes with regard to diagnostic communication:

1. Development mode: In this mode, usually the access to the ECU is realized with the help of a debugger. For details, refer to [Section 4.2.2.1, “Development mode”](#).
2. Production mode: In this mode, the communication to the ECU is only possible via a diagnostic tool (tester tool). Garages work with this mode only. For details, refer to [Section 4.2.2.2, “Production mode”](#).

4.2.2.1. Development mode

During development mode, all modules have enabled the **Development Error Detection** parameter and the `Det` module is included in the project. With this setting, all modules perform the following checks on the incoming API parameters:

- ▶ Check whether the module is initialized before using an API.
- ▶ Check whether the parameters are within the correct range.
- ▶ Check whether incoming pointers are set correctly (no NULL pointer).
- ▶ Perform consistency checks. For example whether the execution of a command is allowed.

In parallel to the checks of the development mode, diagnostic checks are performed. For details refer to [Section 4.2.2.2, “Production mode”](#).

You find further details about the `Det` module in the EB tresos AutoCore Generic Development Error Detection documentation.



4.2.2.2. Production mode

During production mode, all modules have disabled the **Development Error Detection** parameter to save the associated code for API parameter checks. Additionally, the `Det` module itself is removed from the configuration project. It is assumed that all issues were already solved within the integration stage in the development mode.

4.2.3. Diagnostic Trouble Code (DTC)

The Diagnostic Trouble Code (DTC) is needed to identify a diagnostic event unambiguously within a whole vehicle. The scope of a diagnostic event is limited to an ECU. The DTC is mapped to an event-ID in the `Dem` configuration.

There are emission-related DTCs, which are identical for all OEMs. Additionally there are OEM-specific DTCs.

Configure the format of the DTCs in the following way:

- ▶ ISO 14229-1, 3 Byte
- ▶ ISO 15031-6 (mandatory for emission-relevant ECUs), 2 Byte
- ▶ SAEJ1939-73
- ▶ ISO 11992-4

The `Dem` handles the DTC values for all interfaces (`Dcm` and application) within an `uint32` data type and applies the mapping according to [Figure 4.2, “Supported Diagnostic Trouble Code \(DTC\) formats”](#).

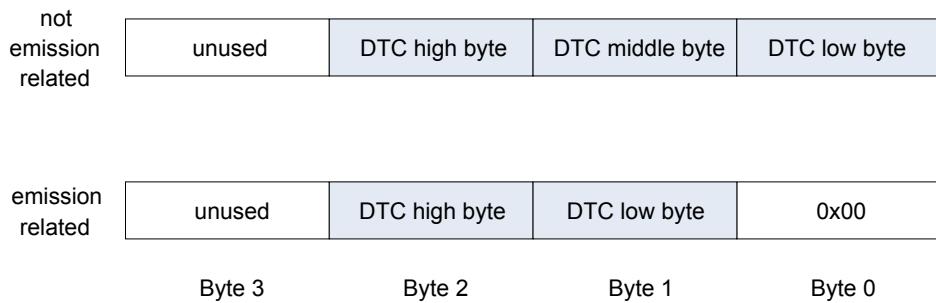


Figure 4.2. Supported Diagnostic Trouble Code (DTC) formats

4.3. Integrating the diagnostic stack

This chapter describes major integration issues of the overall diagnostic stack. For further module-specific integration notes including information on exclusive areas, memory mapping and integration requirements, see



- ▶ [Section 5.2.4, “Integration notes”](#) and [Section 4.4.4, “Dcm integration notes”](#) for the `Dcm` module
- ▶ [Section 5.3.4, “Integration notes”](#) for the `Dem` module
- ▶ [Section 5.4.3, “Integration notes”](#) for the `FIM` module

4.3.1. Dcm integration issues

Versions of the `Dcm` greater than or equal to 4.6.0 can only be integrated with versions of the EB BswM greater than or equal to 1.6.0. This is a limitation regarding backwards compatibility with the EB BswM module.

In order to fulfill the requirement Dcm768, the `Dcm` module invokes the `BswM_Dcm_ApplicationUpdated()` API that was added to the EB BswM in version 1.6.0.

To ensure correct concurrent behavior of the `Dcm` module, observe the following integration restrictions:

- ▶ If asynchronous service processing (`DcmAsyncServiceExecution`) is enabled, `Dcm_ProcessServiceA-`
`sync()` is intended to perform the service processing asynchronously related to the `Dcm_MainFunc-`
`tion()`.

WARNING**Undefined behavior of the ECU**

If you use `Dcm_ProcessServiceAsync()` from the same or higher priority context than `Dcm_MainFunction()`, the cyclic task scheduling can be delayed or violated. This might lead to undefined behavior of your ECU.

To avoid undefined behavior of your ECU, call `Dcm_ProcessServiceAsync()` from a lower priority context than `Dcm_MainFunction()`.

4.3.2. Dem integration issues

Versions of the `Dem` greater than or equal to 6.0.0 can only be integrated with versions of the EB `Dcm` greater than or equal to 4.15.0.

In order to support parallel access to the `Dem` data (see [Section 4.5.2.16, “Parallel event memory access”](#)), several `Dem` APIs used by `Dcm` are implemented according to AUTOSAR 4.3.1 to have `ClientId` as a parameter. The `Dcm` uses the updated APIs from version 4.15.0.

To ensure correct concurrent behavior of the `Dem` module, observe the following integration restrictions:

- ▶ [`Dem_Init\(\)`](#) is not intended to be used before the `Dem` NV status block is completely loaded via `NvM_-`
`ReadAll()`.

**WARNING****Undefined behavior of the ECU**

AUTOSAR states that [Dem_Init\(\)](#) shall be called during the startup phase of the ECU, after the NVRAM Manager has finished restoring NVRAM data. If you call `Dem_Init()` before `NvM_ReadAll()` is finished, the `Dem` module may use outdated or inconsistent data. This might lead to undefined behavior of your ECU.

To avoid undefined behavior of your ECU, do not call the `Dem` API function `Dem_Init()` before the NVRAM Manager has finished restoring NVRAM data.

- ▶ [Dem_PreInit\(\)](#), [Dem_Init\(\)](#) and [Dem_Shutdown\(\)](#) are typically called from the ECU State Manager and are not intended to be used in concurrence with other `Dem` functions. The following exceptions apply:
 - ▶ The following functions can be called regardless of the `Dem` initialization state because they access only unchangeable constant data:
 - ▶ [Dem_GetVersionInfo\(\)](#)
 - ▶ [Dem_GetDTCStatusAvailabilityMask\(\)](#)
 - ▶ [Dem_GetTranslationType\(\)](#)
 - ▶ [Dem_GetDTCofEvent\(\)](#) accesses only calibratable constant data, which shall not change at runtime. So it can be called regardless of the `Dem` initialization state.
 - ▶ The following functions can be called after `Dem_PreInit()`, before or during `Dem_Init()`, and also during and after `Dem_Shutdown()`:
 - ▶ [Dem_ReportErrorStatus\(\)](#)
 - ▶ [Dem_ResetEventDebounceStatus\(\)](#)
 - ▶ [Dem_MainFunction\(\)](#) only performs actions when `Dem` is initialized, but it can be called in any state without any harm.

WARNING**Undefined behavior of the ECU**

If you use the API functions [Dem_PreInit\(\)](#), [Dem_Init\(\)](#), or [Dem_Shutdown\(\)](#) in concurrence with other `Dem` functions, the `Dem` module may use uninitialized data. This might lead to undefined behavior of your ECU.

To avoid undefined behavior of your ECU, do not use the API functions `Dem_PreInit()`, `Dem_Init()`, or `Dem_Shutdown()` in concurrence with other `Dem` functions, except for the functions mentioned above.

- ▶ The initialization and shutdown API functions have to be called consecutively only once and in the following chronological order:
 1. [Dem_PreInit\(\)](#)
 2. [Dem_Init\(\)](#)



3. [Dem_Shutdown\(\)](#)

WARNING

Invalid initialization sequence

If you call the API functions `Dem_PreInit()`, `Dem_Init()` and `Dem_Shutdown()` in incorrect chronological order, or if you call any of the API functions consecutively more than once, the `Dem` module stays unaffected and a corresponding `Det` error is reported.

To avoid an invalid initialization and shutdown sequence:

- ▶ Call the API functions in the chronological order `Dem_PreInit()`, `Dem_Init()`, `Dem_Shutdown()`.
- ▶ Call each of the initialization and shutdown API functions consecutively only once.

- ▶ External DataServices callbacks, in case of C/S operations, should have a minimum run-time. This is necessary because data is collected synchronously (see [Data elements and read functions](#)) and these callbacks are called within the `Dem` master core exclusive area for optimization reasons.

Consequently, the following points have to be taken into account:

- ▶ The callbacks must be integrated and implemented to guarantee direct function calls.
- ▶ A C/S call to another core is not possible because waiting for an event could conflict with the exclusive area. If data from other cores are required, they have to be mirrored to the main core and made available via getter functions.
- ▶ To reduce the run-time, pure getter functions should be implemented that just copy pre-calculated data from global variables.
- ▶ Variant handling via calibration feature (see [Section 4.5.2.14.1, “Calibration”](#)):
 - ▶ For a list of calibratable parameters, see [DemCalibrationSupport](#).
 - ▶ Take into account all limitations that come into effect if calibration is enabled (see MC-SUPPORT in the `Dem` Basic Software Module Description).
 - ▶ If an OBD DTC value of an event is to be disabled via calibration and the event is a member of an OBD Readiness Group, [DemEventOBDReadinessGroup](#) needs to be re-calibrated consistently to `DEM_OBD_RDY_NONE`.
- ▶ Variant handling via dynamic event availability feature (see [Section 4.5.2.14.2, “Dynamic event availability”](#)):
 - ▶ The event available status is stored volatile. Therefore, the desired event available status needs to be set again within each power cycle, immediately after [Dem_Init\(\)](#).
 - ▶ An event can be disabled only if there is no entry for the event in the memory and if none of the event status bits TF, PDTC, CDTC, or WIR are set.
 - ▶ If the operation cycle already started, related event monitors must be disabled and corresponding DTC entries must be cleared before disabling the variant-specific events that are enabled via configuration.



WARNING**Inconsistent UDS status**

If used concurrently with other APIs, `Dem_SetEventAvailable()` can interrupt the event processing of these APIs. This may lead to an inconsistent UDS status for the corresponding events.

To ensure correct concurrent behavior of `Dem_SetEventAvailable()` with other APIs:

- ▶ Use `Dem_SetEventAvailable()` only to **enable** events at run-time. Disable all variant-specific events only by configuration.
 - ▶ Do not execute `Dem_SetEventAvailable()` to disable events while APIs are called that access these events. To ensure, do not call any other API at all before the event available status is completely configured. This includes APIs that process multiple events, e.g. `Dem_SetOperationCycleState()`. Specifically:
 - ▶ Do not activate related monitors before the corresponding events are disabled.
 - ▶ Do not start the `Dcm` service before the event available status is completely configured.
-

WARNING**Orphaned event entries**

An event memory entry of an event that is disabled by configuration parameter `DemAvailabilitySupport` and dynamically enabled at run-time, becomes orphaned if the event is not re-enabled in another variant. In such a case, the event entry cannot be deleted at all. The only way it can be cleared is by event displacement.

To avoid orphaned event entries:

- ▶ Always clear the corresponding DTCs before selecting another variant.
 - ▶ Enable the configuration parameter `DemClearDTCOfDisabledEvents` to allow `Dem_ClearDTC()` for `DEM_DTC_GROUP_ALL_DTCs` to clear event entries of disabled events.
-

▶ Variant handling via side allocation feature (see [Section 4.5.2.14.3, “Side allocation”](#)):

- ▶ The side allocation feature is intended to be used in a similar way as the AUTOSAR *post-build selectable* variant. That means, the DTC assigned to a variant shall not vary during the whole lifetime of the ECU respectively of the flashed software. The variant selection shall be performed only once in each power cycle, before `Dem_Init()`.

**WARNING**

It is not recommended to dynamically change a DTC at run-time. Even if this is not prohibited, such a change might impact running DTC-related `Dcm` or software component query functions. The event related to the DTC might not be found anymore or a different event might be found for the same DTC. For example, if `Dcm` paged buffering is used, the `Dcm` pads the result data with zeros if a DTC cannot be found anymore while a DTC-related query is in progress.

- ▶ The side allocation feature can only be used if a DTC is configured for the event and the DTC is not disabled by calibration.
- ▶ The side allocation feature must not be used to dynamically disable DTCs at run-time. This means the callback function shall not return 0x000000.

For additional constraints regarding the returned DTC value, see [DemCalloutDynamicDTCFnc](#).

- ▶ For performance reasons, the implementation of the function shall focus on minimal processing time because it is called every time the `Dem` needs to access the DTC configuration.
- ▶ Variant handling via DTC suppression (see [Section 4.5.2.14.4, “DTC suppression”](#)):
 - ▶ The DTC suppression status is stored volatile. Therefore, the desired suppression status needs to be set again in each power cycle, immediately after [Dem_Init\(\)](#).
 - ▶ A DTC cannot be suppressed if an event memory entry exists for the related event or combined events.

Event processing is not stopped for suppressed DTCs. Therefore, event data might be stored while the DTC is suppressed. Thus, the DTC might become visible again in the next power cycle.

To prevent the potential reappearance of suppressed DTCs in the next power cycle, it is recommended to always combine the deactivation of the monitor and the suppression of dependent DTCs.

- ▶ DTC suppression might impact running DTC-related `Dcm` or software component query functions. For example, if `Dcm` paged buffering is used, the `Dcm` pads the result data with zeros if a DTC is suppressed while a DTC-related query is in progress.
- ▶ `Dem` BSW distribution functionality (see [Section 4.5.2.13, “Support for BSW event reporting from multiple cores”](#)):
 - ▶ Each configuration that has BSW distribution enabled must have configured one master instance and at least one satellite instance. Only one partition per core must contain a `Dem` instance.
 - ▶ All `Dem` C-API calls (except [Dem_ReportErrorStatus\(\)](#)) have to be performed from modules integrated on the same core as the `Dem` master instance.
 - ▶ The `Dem` master instance must be integrated into the same core and partition as the `Dcm` module and the `NvM` module.
 - ▶ Functions that are generated via `SchM` are `Dem`-internal and must not be used externally. The functions reported via `SchM` have the following names:
 - ▶ `SchM_Receive_Dem_ReportErrorStatusRequire`



- ▶ SchM_Send_Dem_1_Core[!"num:i(\$DemSatelliteCoreId)"!]_-
 ReportErrorStatusProvide
- ▶ SchMSendReportErrorStatusToMaster
- ▶ Dem_ReceiveBswEventReportsOnMaster
- ▶ SchM_Trigger_Dem_FullInitializedProvide
- ▶ The same EventId must only be used from one monitor on one core. It must not be used from multiple cores.

WARNING**BSW distribution - satellite exclusive area**

All satellite exclusive areas use interrupt locking mechanisms because of an unsymmetrical approach/locking. A guard is implemented on one side only. Each `Dem` satellite instance has its own exclusive area to protect its local resources necessary for buffer handling in the pre-initialized phase. The `Dem` master instance exclusive area is called `SCHM DEM EXCLUSIVE AREA MASTER`. The `Dem` satellite exclusive area is called `SCHM EXCLUSIVE AREA_SatelliteCore_<SatelliteCoreId>`.

- ▶ `NvM_ReadBlock/NvM_WriteBlock(<any Dem block>)` is not intended to be used from external context.

WARNING**Undefined behavior of the ECU**

Do not trigger read/write requests from an external context for any `Dem`-designated non-volatile memory block between `Dem_Init()` and `Dem_Shutdown()`. An external trigger might block internal write processing in the `Dem`, and cause DET warnings to be reported. Consequently, the internal request is stopped because the `NvM` is busy handling the external requests for the specific memory block.

For detailed information concerning `NvM`, see the EB tresos AutoCore Generic Memory Stack documentation.

- ▶ The `Dem` service component provides one port of the `EventStatus` interface type per application-related diagnostic event.

WARNING**Only one client port for EventStatus interface type**

The `Dem` service component provides one port of the interface type `EventStatus` per application-related diagnostic event. Therefore it must only be connected to one client port.

- ▶ The `Dem` does not handle the case where more than one monitor shares a single EventId.

**WARNING****Only one client port for DiagnosticMonitor interface type**

The `Dem` service component provides one port of interface type `DiagnosticMonitor` per application-related diagnostic event. Therefore, it must only be connected to one client port.

According to [Dem154] in AUTOSAR 4.0.3 Specification of Diagnostic Event Manager, the `Dem` is not designed to be able to handle the case where more than one monitor shares a single `EventId`.

For integration of the [`Dem_MainFunction\(\)`](#) and configuration of a dedicated `DemTaskTime` parameter, consider the following aspects:

- ▶ ECU/project-specific requirements, e.g. task mapping and scheduling, processor speed, compiler options
- ▶ `Dem` configuration and related increment of processing time within the main function, e.g.:
 - ▶ required error-queue size (refer to [`DemBswErrorBufferSize`](#) parameter)
 - ▶ required processing method (asynchronous/synchronous) for SWC event memory entries (refer to [`DemExtendedDataCapture`](#) / [`DemFreezeFrameCapture`](#) parameters)
 - ▶ size, content, and handling of event memories (see [Section 4.5.2.3, “Event-related data \(event memory\)”](#)) and run-time of related `DataServices` / `StatusChanged` / `DataChanged` callbacks
 - ▶ number of events with configured `InitMonitorForEvent` callback
 - ▶ time-dependent debouncing configuration parameters (refer to [`DemDebounceTimeBasedSupport`](#) / [`DemDebounceFrequencyBasedSupport`](#))
 - ▶ fault confirmation configuration (refer to [`DemEventFailureCycleCounterThreshold`](#) parameter if [`DemResetConfirmedBitOnOverflow`](#) is enabled)
 - ▶ warning indicator configuration for BSW events (refer to [`DemIndicatorAttribute`](#) parameter)
 - ▶ OBD PID \$21/\$31 calculation (refer to [`DemOBDCentralizedPID21Handling`](#) / [`DemOBDCentralizedPID31Handling`](#) parameters)
- ▶ dynamic behavior according to the project-specific integration, e.g.:
 - ▶ maximum possible event-burst between two main function cycles via [`Dem_SetEventStatus\(\)`](#) / [`Dem_ReportErrorStatus\(\)`](#); if required, configure [`DemMaxHandledPassedEventsPerScheduling`](#) / [`DemMaxHandledFailedEventsPerScheduling`](#)
 - ▶ probability of event memory displacement processing (if enabled, refer to [`DemEventDisplacementSupport`](#) parameter)
 - ▶ maximum number of events that are cleared per main function cycle. If required, configure [`DemMaxNumberClearEventsPerCycle`](#) parameter to reduce ClearDTC processing bursts.
 - ▶ maximum number of `InitMonitorForEvent` callback functions called per main function cycle. If required, configure [`DemMaxHandledInitMonitorReenabledPerScheduling`](#) parameter to reduce `InitMonitorForEvent` callback processing bursts.



4.4. Dcm module user guide

4.4.1. Overview

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

- ▶ [Section 4.4.2, “Background information”](#) explains the concepts of the `Dcm` module.
- ▶ [Section 4.4.3, “Configuring the Dcm module”](#) provides instructions on how to configure the `Dcm`.
- ▶ [Section 4.4.4, “Dcm integration notes”](#) provides important notes regarding the integration of the `Dcm`.

4.4.2. Background information

The purpose of the Diagnostic Communication Manager module is to manage the diagnostic data flow. The `Dcm` module communicates with other basic software modules or software components to gather diagnostic information. The diagnostic information can be read by using a diagnostic tester tool, also called tester. The `Dcm` also allows the user to reprogram or update an ECU.

To gather diagnostic information via the `Dcm`, a diagnostic service request can be sent to the ECU, e.g. to retrieve DTCs from the error memory. The diagnostic service request is received and processed by the `Dcm`. After the `Dcm` completed the requested diagnostic service, the `Dcm` sends a response back to the tester.

This chapter provides information on the following topics:

- ▶ [Section 4.4.2.1, “Diagnostic protocol”](#) describes the diagnostic protocol and its components.
- ▶ [Section 4.4.2.2, “Diagnostic session”](#) describes the purpose of diagnostic sessions.
- ▶ [Section 4.4.2.3, “Security level”](#) describes the purpose of security levels.
- ▶ [Section 4.4.2.4, “Bootloader interaction”](#) describes how the `Dcm` can jump to the bootloader.
- ▶ [Section 4.4.2.5, “Diagnostic services”](#) describes diagnostic services and subfunctions. This section also describes the checks that the `Dcm` can perform before it allows the access to diagnostic services and subfunctions.
- ▶ [Section 4.4.2.6, “Diagnostic service request handling”](#) describes how the `Dcm` receives diagnostic service requests, processes them, and sends a response to the diagnostic tester tool.
- ▶ [Section 4.4.2.7, “BSW distribution”](#) describes the BSW distribution concept for the communication between the `Dcm` module and the `ComM` module when they are mapped to different partitions (cores).



4.4.2.1. Diagnostic protocol

The main communication functions of the Dcm are configured in the context of diagnostic protocols. The diagnostic protocol is an instance of processing and not a protocol in the sense of UDS or OBD. The diagnostic protocol specifies the correspondence between the communication channels of the Dcm, the available diagnostic sessions, and the supported diagnostic services on the communication channels.

To allow the Dcm to communicate with the tester, at least one protocol must contain at least one main connection. The main connection contains the reception channels. Incoming diagnostic service requests are received on these reception channels. The main connection also defines the communication between the reception channels and the transmission channel. Responses to diagnostic service requests are sent on these transmission channels.

The possible diagnostic protocol structures are depicted in [Figure 4.3, “The Dcm protocol”](#). You can configure the Dcm as follows:

- ▶ The Dcm can have one or more diagnostic protocols.
- ▶ Each diagnostic protocol can contain one or more main connections.
- ▶ Each main connection can contain one or more receive channels and one transmission channel.

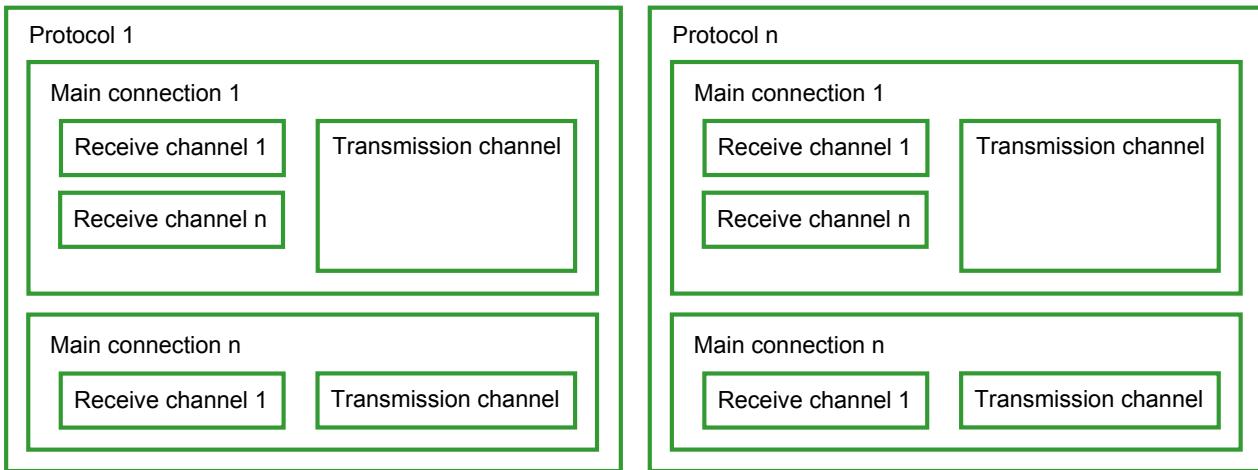


Figure 4.3. The Dcm protocol

Diagnostic service requests

Besides the communication channels, the diagnostic protocol references the configuration of the diagnostic sessions and diagnostic services that the tester can use. The diagnostic session is a state of the ECU in which diagnostic services can be executed. The diagnostic session can be switched via the tester.



4.4.2.2. Diagnostic session

The diagnostic session is a state variable of the ECU. The ECU can only be operated in one diagnostic session at one time.

The uses of the diagnostic session are the following:

- ▶ You can configure certain behavior to only be available when the ECU is within a specific diagnostic session.
- ▶ Certain actions occur when the `Dcm` switches from one diagnostic session to another diagnostic session, either conditionally or unconditionally.
- ▶ When the diagnostic session is switched to another diagnostic session, a jump to the bootloader can be performed. This behavior can be configured for each diagnostic session.
- ▶ The allowed delay to send a response to a diagnostic request is configured within the diagnostic session.

The diagnostic sessions that are available to an ECU, i.e. the set of diagnostic sessions to which the ECU can switch, are defined within the **Session Configuration List** table. Your `Dcm` configuration must contain at least one available diagnostic session. This is the default diagnostic session. The default diagnostic session is the session in which the ECU boots by default and to which the ECU returns after the S3 timer elapsed, i.e. 5 seconds after the last request was received in a non-default diagnostic session.

The following parameters are configured in a diagnostic session:

- ▶ The `DcmDspSessionLevel` parameter is a numeric session identifier that is received as the subfunction ID in the `DiagnosticSessionControl (0x10)` service.
- ▶ The `DcmDspSessionP2ServerMax` parameter is the maximum P2Server time that represents the maximum time until one of the following occurs:
 - ▶ The final response to a diagnostic service request is transmitted.
 - ▶ A first `RequestCorrectlyReceivedResponsePending (0x7F service ID 0x78)` NRC is automatically transmitted by the `Dcm`.
- ▶ The `DcmDspSessionP2StarServerMax` parameter is the maximum P2*Server time that represents the maximum time until one of the following occurs:
 - ▶ The final response to a diagnostic service request is transmitted.
 - ▶ A subsequent `RequestCorrectlyReceivedResponsePending (0x7F service ID 0x78)` NRC is automatically sent out by the `Dcm`.

This is only valid if a `RequestCorrectlyReceivedResponsePending (0x7F service ID 0x78)` NRC was already sent out.

- ▶ The `DcmDspSessionForBoot` parameter specifies whether a switch to the diagnostic session results in the following:
 - ▶ No jump to a bootloader.



- ▶ A jump to the OEM bootloader.
- ▶ A jump to the system supplier bootloader.

You can configure the maximum number of `RequestCorrectlyReceivedResponsePending` (0x7F service ID 0x78) NRCs that is sent as a result of a diagnostic service request globally.

For information on how to configure diagnostic sessions, see [Section 4.4.3.2, “Configuring diagnostic sessions”](#).

4.4.2.2.1. Protocol timers

The P2 timer represents the minimum amount of time before a first response to a diagnostic service request must be available on the bus. This first response can be a positive response or a negative response if it is provided in time. It can also be an internally generated `RequestCorrectlyReceivedResponsePending` (0x7F service ID 0x78) NRC that is sent out before the configured P2 timer elapses.

The P2* timer represents the minimum amount of time before a subsequent response to a diagnostic service request must be available on the bus. A subsequent response is any response which follows a previous `RequestCorrectlyReceivedResponsePending` (0x7F service ID 0x78) NRC. This subsequent response can be a positive or a negative response if it is provided in time. It can also be an internally generated `RequestCorrectlyReceivedResponsePending` (0x7F service ID 0x78) NRC that is sent out before the configured P2* timer elapses.

You can adjust the protocol timers for expected jitter on the bus or other delays caused by processing latency. To do this, you can configure adjustment timers for the P2 timer or P2* timer. The parameters for the adjustment timers are `DcmTimStrP2ServerAdjust` and `DcmTimStrP2StarServerAdjust`. These values are protocol-specific and are automatically subtracted from the actual timers. The adjustment timers cause a first or subsequent automatically generated `RequestCorrectlyReceivedResponsePending` (0x7F service ID 0x78) NRC to be available on the bus sooner than the P2 timer or P2* timer suggest. This is used to ensure that the responses are available on the bus before the tester's time-out elapses.

4.4.2.2. Diagnostic session switch in AUTOSAR 4.0.3 and AUTOSAR 4.0.2

You can configure a switch to another diagnostic session according to AUTOSAR 4.0.2 or AUTOSAR 4.0.3.

Rte interaction for AUTOSAR 4.0.3 or higher

From AUTOSAR 4.0.3 onwards, the `Dcm`, upon a request to switch to a diagnostic session, switches the `Dcm-DiagnosticSessionControl ModeDeclarationGroupPrototype` to the mode which corresponds to the new diagnostic session.

The `DcmModeDeclarationSupport ModeDeclarationGroup` is a Provided Mode Group with the `Dcm` as the provider. The interested parties, e.g. basic software modules such as the `BswM` or complex device dri-



vers, must register or connect to this `ModeDeclarationGroupPrototype`. This enables the basic software modules to receive notifications that its mode switched to a new value and thus either execute certain actions upon the mode switch or inquire about the current mode of this `ModeDeclarationGroupPrototype`. Therefore, you must map the `Provided Mode Group` of the `DcmDiagnosticSessionControl` `ModeDeclarationGroupPrototype` to a `Required Mode Group` of another basic software module. You do the mapping via Rte configuration.

BswM interaction for AUTOSAR 4.0.2 or lower

Up to AUTOSAR 4.0.2 the `Dcm` does the following upon a request to switch to a diagnostic session:

1. The `Dcm` calls the `BswM_Dcm_RequestSessionMode()` API with the requested diagnostic session as parameter.
2. The `Dcm` waits for the `BswM` to call the `Dcm_<SessionName>ModeEntry()` API for the requested diagnostic session.
3. Upon reception of the `Dcm_<SessionName>ModeEntry()` API call, the `Dcm` switches to the requested diagnostic session.

4.4.2.3. Security level

The ECU can be operated in at most one security level at a time. The current security level is considered the active security level. The ECU can also be operated without any security level. The state in which the ECU has no active security level is the default state.

The security levels are used as follows:

- ▶ You can configure a particular behavior so that it is only available if the ECU is within a specific security level.
- ▶ You can configure every security level to remain inaccessible:
 - ▶ for a configurable period of time after a configurable number of failed accesses
 - ▶ for a configurable period of time after boot-up
- ▶ If periodic DID transmission is configured via the service `ReadDataByPeriodicIdentifier` (0x2A), and the security level is switched, the transmission of DIDs which are not readable in the new security level is stopped.

To learn which functions are available at which security level, see the chapter on the particular function.

The ECU is switched to a security level in two operations. First, a `securitySeed` is requested from the ECU by executing a `SecurityAccess` (0x27) service request with the subfunction `RequestSeed` ((2 * The numerical identifier of the requested security level) -1). In the response, the ECU provides a sequence of bytes representing a `securitySeed` that was calculated by the application with an algorithm.



Based on the `securitySeed`, the tester then computes a `securityKey` which is sent to the ECU via a request to the `SecurityAccess` (0x27) service with the subfunction `CompareKey` (2 * The numerical identifier of the requested security level). The application receives this `securityKey` and checks its correctness with an internal algorithm. If the `securityKey` is accepted, the application allows the ECU to transition to the requested security level.

If the `securityKey` is not accepted, the response of the application depends on the default AUTOSAR interface version as configured with the `DcmDefaultASRServiceAPI` parameter.

- ▶ If the `DcmDefaultASRServiceAPI` parameter is set to `AUTOSAR_42`, the application decides which NRC is transmitted.
- ▶ If the `DcmDefaultASRServiceAPI` parameter is set to `AUTOSAR_40` or if the parameter is disabled, the Dcm sends the `InvalidKey` (0x35) NRC.

The ECU resets to the default state of no active security level if any of the following events occur:

- ▶ The ECU resets.
- ▶ The diagnostic session changes from any non-default session to the same or a different non-default session.
- ▶ The diagnostic session changes from a non-default session to the default session.

4.4.2.3.1. AttemptCounter and DelayTimer

For all security levels, you can configure an AttemptCounter. The AttemptCounter tracks the number of failed attempts to enter the requested security level. To activate the AttemptCounter, you must configure either of the following parameters with a non-zero value:

- ▶ `DcmDspSecurityDelayTime`

This parameter specifies the time after a failed security access before another access attempt is allowed. In the following, the term `DelayTimer` is used for this parameter.

- ▶ `DcmDspSecurityDelayTimeOnBoot`

This parameter specifies the time at ECU boot-up during which the Dcm does not accept a security access. In the following, the term `OnBootDelayTimer` is used for this parameter.

Access to the security level is considered to have failed if the application does not accept the `securityKey` provided.

The following events affect the AttemptCounter:

- ▶ A failed `CompareKey` operation: The application returns a `DCM_E_COMPARE_KEY_FAILED` error code to the call that implemented the `CompareKey` operation. This causes the AttemptCounter to increment. The AttemptCounter increments until it reaches the limit value that is specified with the `DcmDspSecuri-`



`tyNumAttDelay` parameter. The AttemptCounter can increment even past this limit value if the `DcmDspSecurityIncrementAttemptCounterPastLimit` parameter is set to TRUE. However, the AttemptCounter can never increment above 255 which is the maximum allowed value for the `DcmDspSecurityNumAttDelay` parameter.

- ▶ A successful `CompareKey` operation: The application returns a `DCM_OK` error code to the call that implemented the `CompareKey` operation. This causes the AttemptCounter to reset to the value of 0.
- ▶ If the `DcmDspSecurityResetAttemptCounterOnTimeout` parameter is configured to TRUE, an expiry of any started `DelayTimer` or `OnBootDelayTimer` related to the security level causes the AttemptCounter to reset to the value of 0. In this case, the amount of time which is necessary for any non-volatile store operation to be completed is added to the started `DelayTimer` or `OnBootDelayTimer` configured for the security level. This means that the security level only becomes accessible when the application finished the store operation, either successfully or unsuccessfully.

If the AttemptCounter reaches a configurable limit of failed accesses for a security level, given by the value of the `DcmDspSecurityNumAttDelay` parameter, one of the following behaviors can be specified:

- ▶ The security level becomes inaccessible for the amount of time as configured with the `DelayTimer`. The `DelayTimer` can be configured with 0. In this case, there is no delay effect. However, if the AttemptCounter needs to be saved to non-volatile memory and is configured to be reset to 0 after the `DelayTimer` elapses, a `DelayTimer` with the value of 0 adopts the amount of time which is necessary to store the AttemptCounter, either successfully or unsuccessfully. The `DelayTimer` can also be configured to be infinite, by setting the `DcmDspSecurityDelayTimeInfinite` parameter to TRUE. In this case, the security level remains permanently inaccessible until the next reboot.
- ▶ All security levels become inaccessible for the amount of time of the largest `DelayTimer`. This is enabled by configuring the `DcmDspSecurityUnifiedDelayTimer` parameter to TRUE. It means that the `DelayTimers` of all security levels are started simultaneously. The AttemptCounters are still treated separately (incremented, considered, stored) per security level. If any reset AttemptCounters need to be stored to non-volatile memory upon time-out of the unified `DelayTimer`, all security levels become accessible only when all store operations are completed, either successfully or unsuccessfully.

The AttemptCounter for a security level can be stored to non-volatile memory by configuring the `DcmDspSecurityAttemptCounterEnabled` parameter to TRUE. If storing of the AttemptCounter to non-volatile memory is enabled, every event that modifies the value of the AttemptCounter of a security level triggers a store operation. If the AttemptCounter is configured to be stored to non-volatile memory, it is reloaded from non-volatile memory at boot-up. The amount of time which is allowed for all reload operations to finish is determined by the value configured in the `DcmDspSecurityMaxAttemptCounterReadoutTime` parameter. Depending on the value retrieved, the following behavior can be specified for each security level:

Scenario A: unified `DelayTimers` are disabled (`DcmDspSecurityUnifiedDelayTimer` parameter is set to FALSE). The following applies to each individual security level:

- ▶ Scenario A1: If reloading the AttemptCounter is successful, i.e. the application returns a `DCM_OK` error code to the call that implemented the reload operation, then:



- ▶ If the `DcmDspSecurityDelayTimeOnBootOverride` parameter is set to `TRUE`, the security level becomes inaccessible for the amount of time as configured in the `OnBootDelayTimer` for the security level's `DcmDspSecurityRow` entry.
- ▶ If the `DcmDspSecurityDelayTimeOnBootOverride` parameter is set to `FALSE`, and if the value of the `AttemptCounter` which is retrieved from the application is greater than or equal to the limit configured for `DcmDspSecurityNumAttDelay`, the security level becomes inaccessible for an amount of time that is the higher value of the `DelayTimer` or `OnBootDelayTimer` in the security level's `DcmDspSecurityRow` entry. If an infinite `DelayTimer` is configured (`DcmDspSecurityDelayTimeInfinite` parameter is set to `TRUE`) the security level becomes permanently inaccessible until the next reboot.
- ▶ Scenario A2: If reloading the `AttemptCounter` is unsuccessful, i.e. the application:
 - ▶ returns a `DCM_OK` error code to the call implementing the restore operation or
 - ▶ returns an unspecified error code to the call implementing the restore operation or
 - ▶ does not finish restoring the `AttemptCounter` value of the security level within the time as specified via `DcmDspSecurityMaxAttemptCounterReadoutTime`
 the following applies:
 - ▶ The `AttemptCounter` is, by default, initialized with the value of the `DcmDspSecurityNumAttDelay` of the security level's `DcmDspSecurityRow` entry.
 - ▶ If the `DcmDspSecurityAttemptCounterReadoutFailInfiniteDelay` parameter is set to `TRUE`, the security level becomes permanently inaccessible until the next reboot.
 - ▶ If the `DcmDspSecurityAttemptCounterReadoutFailInfiniteDelay` parameter is set to `FALSE`, then:
 - ▶ If the `DcmDspSecurityDelayTimeOnBootOverride` parameter is set to `TRUE`, the security level becomes inaccessible for the amount of time as configured in the `OnBootDelayTimer` for the security level's `DcmDspSecurityRow` entry.
 - ▶ If the `DcmDspSecurityDelayTimeOnBootOverride` parameter is set to `FALSE`, and if the value of the `AttemptCounter` which is retrieved from the application is greater than or equal to the limit configured in the `DcmDspSecurityNumAttDelay`, the security level becomes inaccessible for an amount that is the higher value of the `DelayTimer` or `OnBootDelayTimer` in the security level's `DcmDspSecurityRow` entry. If the `DelayTimer` is configured as infinite (`DcmDspSecurityDelayTimeInfinite` parameter is set to `TRUE`) the security level becomes permanently inaccessible until the next reboot.

Scenario B: unified `DelayTimers` are enabled (`DcmDspSecurityUnifiedDelayTimer` parameter is set to `TRUE`). The following applies:

- ▶ Scenario B1: If reloading all `AttemptCounters` is successful, i.e. the application returns a `DCM_OK` error code to all calls implementing the restore operation for all security levels, then all security levels become



inaccessible for the highest amount of time determined for each security level. This amount of time can also be zero. In this case, no `DelayTimers` are started. This is determined, per security level, as follows:

- ▶ If the `DcmDspSecurityDelayTimeOnBootOverride` parameter is set to `TRUE`, the prospective delay for this security level is the value of its `OnBootDelayTimer` in the security level's `DcmDspSecurityRow` entry.
- ▶ If the `DcmDspSecurityDelayTimeOnBootOverride` parameter is set to `FALSE` and the value of the `AttemptCounter` which is retrieved from the application is greater than or equal to the limit configured in the `DcmDspSecurityNumAttDelay` of the security level's `DcmDspSecurityRow` entry, the prospective delay for this security level is the greatest `DelayTimer` or `OnBootDelayTimer` of the security level's `DcmDspSecurityRow` entry. If an infinite `DelayTimer` is configured (`DcmDspSecurityDelayTimeInfinite` is set to `TRUE`), all security levels become permanently inaccessible until the next reboot as this is the highest possible prospective value.
- ▶ Scenario B2: If reloading the `AttemptCounter` is unsuccessful for any security level, i.e. the application:
 - ▶ returns a `DCM_OK` error code to the call implementing the restore operation or
 - ▶ returns an unspecified error code to the call implementing the restore operation or
 - ▶ does not finish restoring the `AttemptCounter` value of the security level within the given `DcmDspSecurityMaxAttemptCounterReadoutTime` time frame
 for any of the configured security levels, the following rules apply:
 - ▶ The `AttemptCounter` of each security level for which the restore operation failed is, by default, initialized with the value of the `DcmDspSecurityNumAttDelay` parameter of the security level's `DcmDspSecurityRow` entry.
 - ▶ If the `DcmDspSecurityAttemptCounterReadoutFailInfiniteDelay` parameter is set to `TRUE`, all security levels become permanently inaccessible until the next reboot.
 - ▶ If the `DcmDspSecurityAttemptCounterReadoutFailInfiniteDelay` parameter is set to `FALSE`, then all security levels become inaccessible for the highest amount of time determined by each security level. This is determined, per security level, as follows:
 - ▶ If the `DcmDspSecurityDelayTimeOnBootOverride` parameter of the security level's `DcmDspSecurityRow` entry is set to `TRUE`, the prospective delay is the value configured for the security level's `OnBootDelayTimer` in the security level's `DcmDspSecurityRow` entry, regardless whether the security level has successfully or unsuccessfully retrieved its `AttemptCounter` value.
 - ▶ If the `DcmDspSecurityDelayTimeOnBootOverride` parameter is set to `FALSE` and the restore operation succeeded for the security level, and if the value of the `AttemptCounter` which is retrieved from the application is greater than or equal to the limit configured in the `DcmDspSecurityNumAttDelay` of the security level's `DcmDspSecurityRow` entry, the prospective delay is the greater of the values configured for the `DelayTimer` or `OnBootDelayTimer` in the security level's `DcmDspSecurityRow` entry. If an infinite `DelayTimer` is configured, i.e. `DcmDspSecurityDelayTimeInfinite` is set to `TRUE`, the prospective delay is infinite and therefore the highest possible.



- ▶ If the `DcmDspSecurityDelayTimeOnBootOverride` parameter of the security level's `DcmDspSecurityRow` entry is set to `FALSE` and the restore operation failed for the security level, the prospective delay is the greater of the values configured for the `DelayTimer` or `OnBootDelayTimer` in the security level's the `DcmDspSecurityRow` entry. If an infinite `DelayTimer` is configured (`DcmDspSecurityDelayTimeInfinite` is set to `TRUE`), all security levels become permanently inaccessible until the next reboot as this is the highest possible prospective value.

4.4.2.3.2. Application interfaces for security levels

For every configured security level, two operations are required from the application by default:

- ▶ the `GetSeed` operation, for performing the functions of the `RequestSeed` subfunction of the `SecurityAccess` (0x27) service
- ▶ the `CompareKey` operation, for performing the functions of the `SendKey` subfunction of the `SecurityAccess` (0x27) service

Additionally, if the `AttemptCounter` for a security level is configured to be stored to non-volatile memory (`DcmDspSecurityAttemptCounterEnabled` parameter is set to `TRUE`), two more operations are necessary for storing and retrieving the `AttemptCounter`:

- ▶ With the `SetSecurityAttemptCounter` operation, the application saves the value of the `AttemptCounter` to non-volatile memory.
- ▶ With the `GetSecurityAttemptCounter` operation, the `Dcm` retrieves the value of the `AttemptCounter` stored in non-volatile memory from the application.

If the `DcmDspSecurityAttemptCounterEnabled` parameter is set to `FALSE` for all configured security levels, legacy support is offered for the storage and retrieval of the `AttemptCounter` values. The legacy mechanism is activated by enabling the configuration parameter `DcmDspSecurityLegacyAttemptCounterHandling`.

For retrieving the `AttemptCounter` for a given security level from non-volatile memory, the following callout function is invoked with every `SendKey` subfunction request of the `SecurityAccess` (0x27) service:

```
uint8 Rte_DcmSecGetNumAtt(Dcm_SecLevelType SecurityLevel)
```

where `Dcm_SecLevelType SecurityLevel` is the security level for which the `AttemptCounter` is retrieved.

The return value of the function represents the value of the `AttemptCounter` retrieved from non-volatile memory.

For storing the `AttemptCounter` for a given security level to non-volatile memory, the following callout function is invoked:

```
void Rte_DcmSecSetNumAtt(Dcm_SecLevelType SecurityLevel, uint8 NumAtt)
```



where:

- ▶ `Dcm_SecLevelType SecurityLevel` is the security level for which the `AttemptCounter` is stored
- ▶ `uint8 NumAtt` is the value of the `AttemptCounter` to be stored.

This callout function is invoked as follows:

- ▶ with every failed `CompareKey` operation, in order to increment the `AttemptCounter` value
- ▶ with every successful `CompareKey` operation, in order to reset the `AttemptCounter` value to zero

NOTE
Limited functionality with legacy AttemptCounter handling


If legacy handling is used, the following limitations apply:

- ▶ `AttemptCounter` values are not restored at boot-up.
- ▶ No function related to restoring `AttemptCounters` at boot-up is available.
- ▶ No function related to `DelayTimer` initialization following the restoring of `AttemptCounters` at boot-up is available.
- ▶ The `AttemptCounter` cannot be reset after a `DelayTimer` has expired.

NOTE
No Rte configuration required with legacy AttemptCounter handling


`Rte` usage is not necessary for legacy `AttemptCounter` storage behavior. Therefore the `Rte` does not need to be configured.

Four interface types are available to implement the operations per security level. The interface type is selected via the `DcmDspSecurityUsePort` parameter of a security level's `DcmDspSecurityRow` entry with the following values:

- ▶ `USE_SECURITY_SYNCH_FNC`: synchronous function callout, requiring only one call in order to fulfill the function. See [Section 4.4.2.3.2.1, “Synchronous function callout security level operations”](#).
- ▶ `USE_SECURITY_ASYNCNCH_FNC`: asynchronous function callout, requiring a chain of calls, one per `Dcm_MainFunction()` cycle in order to fulfill the function. See [Section 4.4.2.3.2.2, “Asynchronous function callout security level operations”](#).
- ▶ `USE_SECURITY_SYNCH_CLIENT_SERVER`: synchronous operation belonging to a `ClientServerInterface`, requiring only one call in order to fulfill the function. See [Section 4.4.2.3.2.3, “Synchronous ClientServerInterface operation security level operations”](#).
- ▶ `USE_SECURITY_ASYNCNCH_CLIENT_SERVER`: asynchronous operation belonging to a `ClientServerInterface`, requiring a chain of calls, one per `Dcm_MainFunction()` cycle in order to fulfill their functions. See [Section 4.4.2.3.2.4, “Asynchronous ClientServerInterface operation security level operations”](#).

**NOTE****No interface generation per security level if legacy AttemptCounter handling is used**

If legacy AttemptCounter handling is enabled, the global interfaces `Rte_DcmSecGetNumAtt()` and `Rte_DcmSecSetNumAtt()` are used.

The actual signature of these operations further depends on the configuration of the following properties:

- ▶ the need to store the AttemptCounter value to non-volatile memory, determined by the value of the `DcmDspSecurityAttemptCounterEnabled` parameter
- ▶ the presence of the `SecurityAccessDataRecord`, selected for every security level by configuring the `DcmDspSecurityADRSIZE` parameter of the security level's `DcmDspSecurityRow` entry with its corresponding size
- ▶ the default AUTOSAR interface version, selected globally by configuring the `DcmDefaultASRServiceAPI` parameter

The following sub-chapters describe the signatures of all functions per interface type.

4.4.2.3.2.1. Synchronous function callout security level operations

The operations for the security level are implemented using synchronous callout functions. These represent direct function calls to a symbol present in the application software and configured, per security level and per operation, in the Dcm.

4.4.2.3.2.1.1. Synchronous function callout interface for the GetSeed operation

The function configured with the `DcmDspSecurityGetSeedFnc` parameter of the security level's `DcmDspSecurityRow` entry is called in order to fulfill the functionality of the `GetSeed` operation. The signature of this function depends on the value of the `DcmDspSecurityADRSIZE` parameter of the security level's `DcmDspSecurityRow` entry. The signature of this function does not depend on the value of the `DcmDefaultASRServiceAPI` parameter.

- ▶ If the `DcmDspSecurityADRSIZE` parameter is disabled for the security level's `DcmDspSecurityRow` entry, the `GetSeed` callout function shall have the following signature:

```
Std_ReturnType <GetSeedFunctionName>( uint8 * Seed, Dcm_NegativeResponseCodeType * ErrorCode )
```

where:

- ▶ `uint8 * Seed` is a pointer to a buffer where the application must write the generated `securitySeed`. This data is serialized in the diagnostic service response.
- ▶ `Dcm_NegativeResponseCodeType * ErrorCode` is a pointer at which the application may write an NRC to be transmitted in case the operation is not accepted.



The function's valid error codes are:

- ▶ E_OK: The operation has succeeded and the generated securitySeed can be sent to the tester.
- ▶ E_NOT_OK: The operation has failed. The NRC returned via the ErrorCode parameter is sent to the tester.
- ▶ If the DcmDspSecurityADRSIZE parameter is enabled a security level's DcmDspSecurityRow entry, the GetSeed callout function has the following signature:

```
Std_ReturnType <GetSeedFunctionName> ( const uint8 * SecurityAccessDataRecord,
uint8 * Seed, Dcm_NegativeResponseCodeType* ErrorCode )
```

where:

- ▶ const uint8 * SecurityAccessDataRecord is a pointer to a buffer where the securityAccessDataRecord from the RequestSeed diagnostic request is stored. This is an auxiliary parameter via which an additional level of security may be implemented in the application by checking the validity of this record and deciding whether to accept or reject the request.
- ▶ uint8 * Seed is a pointer to a buffer where the application must write the generated securitySeed. This data is serialized in the diagnostic service response.
- ▶ Dcm_NegativeResponseCodeType * ErrorCode is a pointer at which the application may write an NRC to be transmitted in case the operation is not accepted.

The function's valid error codes are:

- ▶ E_OK: The operation has succeeded and the generated securitySeed can be sent to the tester.
- ▶ E_NOT_OK: The operation has failed. The NRC returned via the ErrorCode parameter is sent to the tester.

4.4.2.3.2.1.2. Synchronous function callout interface for the CompareKey operation

The function configured in the DcmDspSecurityCompareKeyFnc parameter of the security level's DcmDspSecurityRow entry is called in order to fulfill the functionality of the CompareKey operation. The signature of this function does not depend on the value of the DcmDspSecurityADRSIZE parameter of the security level's DcmDspSecurityRow entry. The signature of this function depends on the value of the DcmDefaultASRServiceAPI parameter.

- ▶ If the DcmDefaultASRServiceAPI parameter is set to AUTOSAR_40, the CompareKey callout function shall have the following signature:

```
Std_ReturnType <CompareKeyFunctionName> ( const uint8* Key )
```

where:

- ▶ const uint8 * Key is a pointer to a buffer containing the securityKey received from the tester.



The function's valid error codes are:

- ▶ `E_OK`: The operation has succeeded. A positive response is sent to the tester and the ECU transitions to the requested security level.
- ▶ `E_NOT_OK`: The operation has failed. The `InvalidKey` (0x35) NRC is sent to the tester.
- ▶ `DCM_E_COMPARE_KEY_FAILED`: The operation has failed. The `InvalidKey` (0x35) NRC is sent to the tester and the `AttemptCounter` of the security level is incremented.
- ▶ If the `DcmDefaultASRServiceAPI` parameter is set to `AUTOSAR_42`, the `CompareKey` callout function shall have the following signature:

```
Std_ReturnType <CompareKeyFunctionName> ( const uint8* Key, Dcm_NegativeResponseCodeType * ErrorCode )
```

where:

- ▶ `const uint8 * Key` is a pointer to a buffer containing the `securityKey` received from the tester.
- ▶ `Dcm_NegativeResponseCodeType * ErrorCode` is a pointer at which the application may write an NRC to be transmitted in case the operation is not accepted.

The function's valid error codes are:

- ▶ `E_OK`: The operation has succeeded. A positive response is sent to the tester and the ECU transitions to the requested security level.
- ▶ `E_NOT_OK`: The operation has failed. The NRC returned via the `ErrorCode` parameter is sent to the tester.
- ▶ `DCM_E_COMPARE_KEY_FAILED`: The operation has failed. The `InvalidKey` (0x35) NRC is sent to the tester and the `AttemptCounter` of the security level is incremented.

4.4.2.3.2.1.3. Synchronous function callout interface for the `GetSecurityAttemptCounter` operation

The function configured in the `DcmDspSecurityGetAttemptCounterFnc` parameter of the security level's `DcmDspSecurityRow` entry is called in order to reload the value of the `AttemptCounter` for the security level from non-volatile memory. This operation is only required if the `AttemptCounter` for the security level is configured to be stored to non-volatile memory (`DcmDspSecurityAttemptCounterEnabled` parameter is set to `TRUE`). The signature of this function does not depend on the value of the `DcmDspSecurityADRSIZE` parameter of the security level's `DcmDspSecurityRow` entry. The signature of this function does not depend on the value of the `DcmDefaultASRServiceAPI` parameter.

The `GetSecurityAttemptCounter` callout function shall have the following signature:

```
Std_ReturnType <GetSecurityAttemptCounterFunctionName> ( uint8 * AttemptCounter )
```

where:



- ▶ `uint8 * AttemptCounter` is a pointer at which the application writes the retrieved value of the `AttemptCounter`.

The function's valid error codes are:

- ▶ `E_OK`: The operation has succeeded.
- ▶ `E_NOT_OK`: The operation has failed.

4.4.2.3.2.1.4. Synchronous function callout interface for the `SetSecurityAttemptCounter` operation

The function configured in the `DcmDspSecuritySetAttemptCounterFnc` parameter of the security level's `DcmDspSecurityRow` entry is called in order to store the value of the `AttemptCounter` for the security level to non-volatile memory. This operation is only required if the `AttemptCounter` for the security level is configured to be stored to non-volatile memory (`DcmDspSecurityAttemptCounterEnabled` parameter is set to `TRUE`). The signature of this function does not depend on the value of the `DcmDspSecurityADRSIZE` parameter of the security level's `DcmDspSecurityRow` entry. The signature of this function does not depend on the value of the `DcmDefaultASRServiceAPI` parameter.

The `SetSecurityAttemptCounter` callout function shall have the following signature:

```
Std_ReturnType <SetSecurityAttemptCounterFunctionName>(uint8 AttemptCounter)
```

where:

- ▶ `uint8 AttemptCounter` is the value of the `AttemptCounter` that the application shall store in non-volatile memory.

The function's valid error codes are:

- ▶ `E_OK`: The operation has succeeded.
- ▶ `E_NOT_OK`: The operation has failed.

4.4.2.3.2.2. Asynchronous function callout security level operations

The operations for the security level are implemented using asynchronous callout functions. These represent direct function calls to a symbol present in the application software and configured, per security level and per operation, in the `Dcm`. Asynchronous function calls may finish their operations after more than one call. These shall be called repeatedly until either:

- ▶ the application signals that the operation has finished by returning a "final" error:
 - ▶ `E_OK`
 - ▶ `E_NOT_OK`
 - ▶ `DCM_E_COMPARE_KEY_FAILED`, where applicable



- ▶ the `Dcm` cancels the operation as a result of either:
 - ▶ a time-out resulting from reaching the maximum allowed number of `RequestCorrectlyReceivedResponsePending` (0x78) NRCs, if the operation is called in the context of a diagnostic service request
 - ▶ a time-out resulting from reaching the maximum allowed number of calls to the operation in question, if the operation is not called in the context of a diagnostic service request
 - ▶ protocol preemption

The application interface is always called for the first invocation with an `OpStatus` value of `DCM_INITIAL`.

An application interface signals that it must be invoked once more by returning the error code `DCM_E_PENDING`. As a result, the `Dcm` shall call the application interface in the next `Dcm_MainFunction()` cycle with an `OpStatus` value of `DCM_PENDING` in order to signal such a subsequent call.

During such a chain of calls, the `Dcm` monitors the P2 and P2* timers of the diagnostic session and handles transmission of `RequestCorrectlyReceivedResponsePending` (0x7F 0x27 0x78) NRCs internally. Such an NRC transmission may also be forced by returning a `DCM_E_FORCE_RCRRP` error code where possible (signalled by the availability of this error code). In order to signal a successful transmission of such a forced NRC, the subsequent call to the application interface takes place after transmission confirmation with an `OpStatus` value of `DCM_FORCE_RCRRP_OK`.

The `Dcm` also counts the number of `RequestCorrectlyReceivedResponsePending` (0x7F 0x27 0x78) NRCs transmitted in order to cancel the application interface call chain if this number reaches the value configured for the `DcmDslDiagRespMaxNumRespPend` parameter. This is performed regardless of the reason (time-out or forced transmission) of the NRC transmission.

An application interface is signalled to cancel its operations with a call with an `OpStatus` value of `DCM_CANCEL`. This shall always signal the last call of the asynchronous call chain. The error code returned by the application to such a call is ignored.

4.4.2.3.2.2.1. Asynchronous function callout interface for the GetSeed operation

The function configured in the `DcmDspSecurityGetSeedFnc` parameter of the security level's `DcmDspSecurityRow` entry is called in order to fulfill the functionality of the `GetSeed` operation. The signature of this function depends on the value of the `DcmDspSecurityADRSIZE` parameter of the security level's `DcmDspSecurityRow` entry. The signature of this function does not depend on the value of the `DcmDefaultASRSERVICEAPI` parameter.

- ▶ If the `DcmDspSecurityADRSIZE` parameter is disabled for the security level's `DcmDspSecurityRow` entry, the `GetSeed` callout function shall have the following signature:

```
Std_ReturnType <RequestSeedFunctionName>( Dcm_OpStatusType OpStatus, uint8 * Seed, Dcm_NegativeResponseCodeType * ErrorCode )
```



where:

- ▶ `Dcm_OpStatusType` `OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this callout.
 - ▶ `DCM_PENDING`: This is a subsequent call of this callout after it returned a `DCM_E_PENDING` error code in a previous call. This allows processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_CANCEL`: This is a call for cancellation of the callout. The callout shall terminate all processing and shall reset any internal variables and states to a default value. All results of processing performed from the last call with a `DCM_INITIAL` `OpStatus` up to this call shall be considered null. The error code of this call is not internally considered by the `Dcm`.
- ▶ `uint8 * Seed` is a pointer to a buffer where the application must write the generated `securitySeed`. This data is serialized in the diagnostic service response.
- ▶ `Dcm_NegativeResponseCodeType * ErrorCode` is a pointer at which the application may write an NRC to be transmitted in case the operation is not accepted.

The function's valid error codes are:

- ▶ `E_OK`: The operation has succeeded and the generated `securitySeed` can be sent to the tester.
- ▶ `E_NOT_OK`: The operation has failed. The NRC returned via the `ErrorCode` parameter is sent to the tester.
- ▶ `DCM_E_PENDING`: Request processing is pending and shall need a further invocation of the callout function. Return this error code in order to split the processing of a request into fragments which will not be too long so as to interfere with scheduling.
- ▶ If the `DcmDspSecurityADRSIZE` parameter is enabled for the security level's `DcmDspSecurityRow` entry, the `GetSeed` callout function shall have the following signature:

```
Std_ReturnType <RequestSeedFunctionName>( const uint8 * SecurityAccess-
DataRecord, Dcm_OpStatusType OpStatus, uint8 * Seed, Dcm_NegativeResponseCode-
Type* ErrorCode )
```

where:

- ▶ `const uint8 * SecurityAccessDataRecord` is a pointer to a buffer where the `SecurityAccessDataRecord` from the `RequestSeed` diagnostic request is stored. This is an auxiliary parameter via which an additional level of security may be implemented in the application by checking the validity of this record and deciding whether to accept or reject the request.
- ▶ `Dcm_OpStatusType` `OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this callout.



- ▶ `DCM_PENDING`: This is a subsequent call of this callout after it returned a `DCM_E_PENDING` error code in a previous call. This allows processing to be broken up over more than one `Dcm_MainFunction()` cycle.
- ▶ `DCM_CANCEL`: This is a call for cancellation of the callout. The callout shall terminate all processing and shall reset any internal variables and states to a default value. All results of processing performed from the last call with a `DCM_INITIAL` `OpStatus` up to this call shall be considered null. The error code of this call is not internally considered by the `Dcm`.
- ▶ `uint8 * Seed` is a pointer to a buffer where the application must write the generated `securitySeed`. This data is serialized in the diagnostic service response.
- ▶ `Dcm_NegativeResponseCodeType * ErrorCode` is a pointer at which the application may write an NRC to be transmitted if the operation is not accepted.

The function's valid error codes are:

- ▶ `E_OK`: The operation has succeeded and the generated `securitySeed` can be sent to the tester.
- ▶ `E_NOT_OK`: The operation has failed. The NRC returned via the `ErrorCode` parameter is sent to the tester.
- ▶ `DCM_E_PENDING`: Request processing is pending and shall need a further invocation of the callout function. Return this error code in order to split the processing of a request into fragments which are not too long to interfere with scheduling.

4.4.2.3.2.2.2. Asynchronous function callout interface for the CompareKey operation

The function configured in the `DcmDspSecurityCompareKeyFnc` parameter of the security level's `DcmDspSecurityRow` entry is called in order to fulfill the functionality of the `CompareKey` operation. The signature of this function does not depend on the value of the `DcmDspSecurityADRSIZE` parameter of the security level's `DcmDspSecurityRow` entry. The signature of this function depends on the value of the `DcmDefaultASRServiceAPI` parameter.

- ▶ If the `DcmDefaultASRServiceAPI` parameter is set to `AUTOSAR_40`, the `CompareKey` callout function shall have the following signature:

```
Std_ReturnType <CompareKeyFunctionName>( const uint8* Key, Dcm_OpStatusType OpStatus )
```

where:

- ▶ `const uint8 * Key` is a pointer to a buffer containing the `securityKey` received from the tester.
- ▶ `Dcm_OpStatusType OpStatus` represents the type of the current call to the callout function:
- ▶ `DCM_INITIAL`: This is the initial call of this callout.



- ▶ **DCM_PENDING:** This is a subsequent call of this callout after it returned a `DCM_E_PENDING` error code in a previous call. This allows processing to be broken up over more than one `Dcm_MainFunction()` cycle.
- ▶ **DCM_CANCEL:** This is a call for cancellation of the callout. The callout shall terminate all processing and shall reset any internal variables and states to a default value. All results of processing performed from the last call with a `DCM_INITIAL` OpStatus up to this call shall be considered null. The error code of this call is not internally considered by the `Dcm`.

The function's valid error codes are:

- ▶ `E_OK`: The operation has succeeded. A positive response is sent to the tester and the ECU transitions to the requested security level.
- ▶ `E_NOT_OK`: The operation has failed. The `InvalidKey` (0x35) NRC is sent to the tester.
- ▶ `DCM_E_COMPARE_KEY_FAILED`: The operation has failed. The `InvalidKey` (0x35) NRC is sent to the tester and the AttemptCounter of the security level is incremented.
- ▶ `DCM_E_PENDING`: Request processing is pending and shall need a further invocation of the callout function. Return this error code in order to split the processing of a request into fragments which are not too long to interfere with scheduling.
- ▶ If the `DcmDefaultASRServiceAPI` parameter is set to `AUTOSAR_42`, the `CompareKey` callout function shall have the following signature:

```
Std_ReturnType <CompareKeyFunctionName> ( const uint8* Key, Dcm_OpStatusType
OpStatus, Dcm_NegativeResponseCodeType * ErrorCode )
```

where:

- ▶ `const uint8 * Key` is a pointer to a buffer containing the `securityKey` received from the tester.
- ▶ `Dcm_OpStatusType OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this callout.
 - ▶ `DCM_PENDING`: This is a subsequent call of this callout after it returned a `DCM_E_PENDING` error code in a previous call. This allows processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_CANCEL`: This is a call for cancellation of the callout. The callout shall terminate all processing and shall reset any internal variables and states to a default value. All results of processing performed from the last call with a `DCM_INITIAL` OpStatus up to this call shall be considered null. The error code of this call is not internally considered by the `Dcm`.
- ▶ `Dcm_NegativeResponseCodeType * ErrorCode` is a pointer at which the application may write an NRC to be transmitted in case the operation is not accepted.

The function's valid error codes are:



- ▶ `E_OK`: The operation has succeeded. A positive response is sent to the tester and the ECU transitions to the requested security level.
- ▶ `E_NOT_OK`: The operation has failed. The NRC returned via the `ErrorCode` parameter is sent to the tester.
- ▶ `DCM_E_COMPARE_KEY_FAILED`: The operation has failed. The `InvalidKey` (0x35) NRC is sent to the tester and the `AttemptCounter` of the security level is incremented.
- ▶ `DCM_E_PENDING`: Request processing is pending and shall need a further invocation of the callout function. Return this error code in order to split the processing of a request into fragments which are not too long to interfere with scheduling.

4.4.2.3.2.2.3. Asynchronous function callout interface for the `GetSecurityAttemptCounter` operation

The function configured in the `DcmDspSecurityGetAttemptCounterFnc` parameter of the security level's `DcmDspSecurityRow` entry is called in order to reload the value of the `AttemptCounter` for the security level from non-volatile memory. This operation is only required if the `AttemptCounter` for the security level is configured to be stored to non-volatile memory (`DcmDspSecurityAttemptCounterEnabled` parameter is set to TRUE). The signature of this function does not depend on the value of the `DcmDspSecurityADRSIZE` parameter of the security level's `DcmDspSecurityRow` entry. The signature of this function does not depend on the value of the `DcmDefaultASRServiceAPI` parameter.

The `GetSecurityAttemptCounter` callout function shall have the following signature:

```
Std_ReturnType <GetSecurityAttemptCounterFunctionName> ( Dcm_OpStatusType OpStatus,
                                                       uint8 * AttemptCounter )
```

where:

- ▶ `Dcm_OpStatusType OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this callout.
 - ▶ `DCM_PENDING`: This is a subsequent call of this callout after it has returned a `DCM_E_PENDING` error code in a previous call. This allows processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_CANCEL`: This is a call for cancellation of the callout. The callout shall terminate all processing and shall reset any internal variables and states to a default value. All results of processing performed from the last call with a `DCM_INITIAL` `OpStatus` up to this call shall be considered null. The error code of this call is not internally considered by the `Dcm`.
- ▶ `uint8 * AttemptCounter` is a pointer at which the application writes the retrieved value of the `AttemptCounter`.

The function's valid error codes are:



- ▶ `E_OK`: The operation has succeeded.
- ▶ `E_NOT_OK`: The operation has failed.
- ▶ `DCM_E_PENDING`: Request processing is pending and shall need a further invocation of the callout function. Return this error code in order to split the processing of a request into fragments which are not too long to interfere with scheduling.

4.4.2.3.2.2.4. Asynchronous function callout interface for the `SetSecurityAttemptCounter` operation

The function configured in the `DcmDspSecuritySetAttemptCounterFnc` parameter of the security level's `DcmDspSecurityRow` entry is called in order to store the value of the `AttemptCounter` for the security level to non-volatile memory. This operation is only required if the `AttemptCounter` for the security level is configured to be stored to non-volatile memory (`DcmDspSecurityAttemptCounterEnabled` parameter is set to `TRUE`). The signature of this function does not depend on the value of the `DcmDspSecurityADRSIZE` parameter of the security level's `DcmDspSecurityRow` entry. The signature of this function does not depend on the value of the `DcmDefaultASRServiceAPI` parameter.

The `SetSecurityAttemptCounter` callout function shall have the following signature:

```
Std_ReturnType <SetSecurityAttemptCounterFunctionName>(Dcm_OpStatusType OpStatus,uint8 AttemptCounter)
```

where:

- ▶ `Dcm_OpStatusType` `OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this callout.
 - ▶ `DCM_PENDING`: This is a subsequent call of this callout after it returned a `DCM_E_PENDING` error code in a previous call. This allows processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_FORCE_RCRRP_OK`: This is a subsequent call of this callout after it returned a `DCM_E_FORCE_RCRRP` error code in a previous call. A call with this `OpStatus` value occurs after the positive confirmation of the transmission of the requested `RequestCorrectlyReceivedResponsePending` (0x7F 0x27 0x78) NRC. This allows processing to be broken up over more than one `Dcm_MainFunction()` cycle and, at the same time, an immediate transmission of a `RequestCorrectlyReceivedResponsePending` (0x7F 0x27 0x78) NRC.
 - ▶ `DCM_CANCEL`: This is a call for cancellation of the callout. The callout shall terminate all processing and shall reset any internal variables and states to a default value. All results of processing performed from the last call with a `DCM_INITIAL` `OpStatus` up to this call shall be considered null. The error code of this call is not internally considered by the `Dcm`.
- ▶ `uint8` `AttemptCounter` is the value of the `AttemptCounter` that the application shall store in non-volatile memory.



The function's valid error codes are:

- ▶ E_OK: The operation has succeeded.
- ▶ E_NOT_OK: The operation has failed.
- ▶ DCM_E_PENDING: Request processing is pending and shall need a further invocation of the callout function. Return this error code in order to split the processing of a request into fragments which are not too long to interfere with scheduling.
- ▶ DCM_E_FORCE_RCRRP: Request processing is pending and shall need a further invocation of the callout function. Additionally, the application requests the immediate transmission of a RequestCorrectlyReceivedResponsePending (0x7F 0x27 0x78) NRC.

4.4.2.3.2.3. Synchronous ClientServerInterface operation security level operations

The operations for the security level are implemented via calls to operations on RequiredPorts implementing the ClientServerInterface:

`ClientServerInterface SecurityAccess_<SecurityLevel>`

where `SecurityLevel` is the short name configured for the security level's `DcmDspSecurityRow` entry.

The operations:

- ▶ GetSeed
- ▶ CompareKey

are always present for the `SecurityAccess_<SecurityLevel> ClientServerInterface`. Additionally, if the `AttemptCounter` for the security level is configured to be stored to non-volatile memory (`DcmDspSecurityAttemptCounterEnabled` parameter is set to `TRUE`), this interface shall also contain the operations:

- ▶ GetSecurityAttemptCounter
- ▶ SetSecurityAttemptCounter

For every security level implementing its operations using this mechanism, the following RequiredPort is generated:

`SecurityAccess_<SecurityLevel>`

where `SecurityLevel` is the short name configured for the security level's `DcmDspSecurityRow` entry.

The available codes of the `ClientServerInterface SecurityAccess_<SecurityLevel>` are:

```
PossibleErrors { E_NOT_OK = 1, E_COMPARE_KEY_FAILED = 11 };
```



4.4.2.3.2.3.1. Synchronous ClientServerInterface operations for the GetSeed operation

The `GetSeed` operation of the `SecurityAccess_<SecurityLevel>` `ClientServerInterface` is invoked in order to fulfill the functionality of the `GetSeed` operation. The signature of this operation depends on the value of the `DcmDspSecurityADRSIZE` parameter of the security level's `DcmDspSecurityRow` entry. The signature of this operation does not depend on the value of the `DcmDefaultASRServiceAPI` parameter.

- ▶ If the `DcmDspSecurityADRSIZE` parameter is disabled for the security level's `DcmDspSecurityRow` entry, the `GetSeed` operation shall have the following signature:

```
GetSeed( OUT uint8 Seed[<DcmDspSecuritySeedSize>], OUT NegativeResponseCodeType ErrorCode, ERR{E_NOT_OK} );
```

where:

- ▶ `OUT uint8 Seed[<DcmDspSecuritySeedSize>]` is a pointer to a buffer where the application must write the generated `securitySeed`. This buffer shall have the maximum size given as the value of the `DcmDspSecuritySeedSize` parameter for the current security level. This data is serialized in the diagnostic service response.
- ▶ `OUT NegativeResponseCodeType ErrorCode` is a pointer at which the application may write an NRC to be transmitted if the operation is not accepted.

The operation's valid error codes are:

- ▶ `E_OK`: The operation has succeeded and the generated `securitySeed` can be sent to the tester.
- ▶ `E_NOT_OK`: The operation has failed. The NRC returned via the `ErrorCode` parameter is sent to the tester.
- ▶ If the `DcmDspSecurityADRSIZE` parameter is enabled for the security level's `DcmDspSecurityRow` entry, the `GetSeed` callout function shall have the following signature:

```
GetSeed( IN uint8 SecurityAccessDataRecord[<DcmDspSecurityADRSIZE>], OUT uint8 Seed[<DcmDspSecuritySeedSize>], OUT NegativeResponseCodeType ErrorCode, ERR{E_NOT_OK} );
```

where:

- ▶ `IN uint8 SecurityAccessDataRecord[<DcmDspSecurityADRSIZE>]` is a pointer to a buffer where the `SecurityAccessDataRecord` from the `RequestSeed` diagnostic request is stored. This buffer shall have the maximum size given as the value of the `DcmDspSecurityADRSIZE` of the security level's `DcmDspSecurityRow` entry. This is an auxiliary parameter via which an additional level of security may be implemented in the application by checking the validity of this record and deciding whether to accept or reject the request.
- ▶ `OUT uint8 Seed[<DcmDspSecuritySeedSize>]` is a pointer to a buffer where the application must write the generated `securitySeed`. This buffer shall have the maximum size given as the value



of the `DcmDspSecuritySeedSize` parameter for the current security level. This data is serialized in the diagnostic service response.

- ▶ OUT `NegativeResponseCodeType ErrorCode` is a pointer at which the application may write an NRC to be transmitted if the operation is not accepted.

The operation's valid error codes are:

- ▶ `E_OK`: The operation has succeeded and the generated `securitySeed` can be sent to the tester.
- ▶ `E_NOT_OK`: The operation has failed. The NRC returned via the `ErrorCode` parameter is sent to the tester.

4.4.2.3.2.3.2. Synchronous ClientServerInterface operations for the CompareKey operation

The `CompareKey` operation of the `SecurityAccess_<SecurityLevel>` ClientServerInterface is invoked in order to fulfill the functionality of the `CompareKey` operation. The signature of this operation does not depend on the value of the `DcmDspSecurityADRSIZE` parameter of the security level's `DcmDspSecurityRow` entry. The signature of this operation depends on the value of the `DcmDefaultASRServiceAPI` parameter.

- ▶ If the `DcmDefaultASRServiceAPI` parameter is set to `AUTOSAR_40`, the `CompareKey` operation shall have the following signature:

```
CompareKey( IN uint8 Key[<DcmDspSecurityKeySize>], ERR{E_NOT_OK, E_COMPARE_KEY_FAILED} );
```

Where:

- ▶ `IN uint8 Key[<DcmDspSecurityKeySize>]` Is a pointer to a buffer containing the `securityKey` received from the tester. This buffer shall have the maximum size given as the value of the `DcmDspSecurityKeySize` parameter for the current security level.

The operation's valid error codes are:

- ▶ `E_OK`: The operation has succeeded. A positive response is sent to the tester and the ECU transitions to the requested security level .
- ▶ `E_NOT_OK`: The operation has failed. The `InvalidKey` (0x35) NRC is sent to the tester
- ▶ `DCM_E_COMPARE_KEY_FAILED`: The operation has failed. The `InvalidKey` (0x35) NRC is sent to the tester and the `AttemptCounter` of the security level is incremented.
- ▶ If the `DcmDefaultASRServiceAPI` parameter is set to `AUTOSAR_42`, the `CompareKey` operation shall have the following signature:

```
CompareKey( IN uint8 Key[<DcmDspSecurityKeySize>], OUT Dcm_NegativeResponseCodeType ErrorCode, ERR{E_NOT_OK, E_COMPARE_KEY_FAILED} );
```

Where:



- ▶ IN uint8 Key[<DcmDspSecurityKeySize>] Is a pointer to a buffer containing the securityKey received from the tester. This buffer shall have the maximum size given as the value of the DcmDspSecurityKeySize parameter for the current security level.
- ▶ OUT Dcm_NegativeResponseCodeType ErrorCode is a pointer at which the application may write an NRC to be transmitted in case the operation is not accepted.

The operation's valid error codes are:

- ▶ E_OK: The operation has succeeded. A positive response is sent to the tester and the ECU transitions to the requested security level .
- ▶ E_NOT_OK: The operation has failed. The NRC returned via the ErrorCode parameter is sent to the tester.
- ▶ DCM_E_COMPARE_KEY_FAILED: The operation has failed. The InvalidKey (0x35) NRC is sent to the tester and the AttemptCounter of the security level is incremented.

4.4.2.3.2.3.3. Synchronous ClientServerInterface operations for the GetSecurityAttemptCounter operation

The GetSecurityAttemptCounter operation of the SecurityAccess_<SecurityLevel> ClientServerInterface is invoked in order to reload the value of the AttemptCounter for the security level from non-volatile memory. This operation is only required if the AttemptCounter for the security level is configured to be stored to non-volatile memory (i.e., the DcmDspSecurityAttemptCounterEnabled parameter is set to TRUE). The signature of this operation does not depend on the value of the DcmDspSecurityADRSIZE parameter of the security level's DcmDspSecurityRow entry. The signature of this operation does not depend on the value of the DcmDefaultASRServiceAPI parameter.

The GetSecurityAttemptCounter operation shall have the following signature:

```
GetSecurityAttemptCounter( OUT uint8 AttemptCounter, ERR{E_NOT_OK} );
```

Where:

- ▶ OUT uint8 AttemptCounter is a pointer at which the application writes the retrieved value of the AttemptCounter.

The operation's valid error codes are:

- ▶ E_OK: The operation has succeeded.
- ▶ E_NOT_OK: The operation has failed.



4.4.2.3.2.3.4. Synchronous ClientServerInterface operations for the SetSecurityAttemptCounter operation

The `SetSecurityAttemptCounter` operation of the `SecurityAccess_<SecurityLevel>` ClientServerInterface is invoked in order to store the value of the AttemptCounter for the security level to non-volatile memory. This operation is only required if the AttemptCounter for the security level is configured to be stored to non-volatile memory (i.e., the `DcmDspSecurityAttemptCounterEnabled` parameter is set to TRUE). The signature of this operation does not depend on the value of the `DcmDspSecurityADRSIZE` parameter of the security level's `DcmDspSecurityRow` entry. The signature of this operation does not depend on the value of the `DcmDefaultASRServiceAPI` parameter.

The `SetSecurityAttemptCounter` operation shall have the following signature:

```
SetSecurityAttemptCounter( IN uint8 AttemptCounter, ERR{E_NOT_OK} );
```

Where:

- ▶ `uint8 AttemptCounter` is the value of the AttemptCounter that the application shall store in non-volatile memory

The function's valid error codes are:

- ▶ `E_OK`: The operation has succeeded.
- ▶ `E_NOT_OK`: The operation has failed.

4.4.2.3.2.4. Asynchronous ClientServerInterface operation security level operations

The operations for the security level are implemented via calls to operations on RequiredPorts implementing the ClientServerInterface:

```
ClientServerInterface SecurityAccess_<SecurityLevel>
```

where `SecurityLevel` is the short name configured for the security level's `DcmDspSecurityRow` entry.

Asynchronous operations may finish their operations after more than one call. These shall be called repeatedly until either:

- ▶ the application signals that the operation has finished by returning a "final" error:
 - ▶ `E_OK`
 - ▶ `E_NOT_OK`
 - ▶ `DCM_E_COMPARE_KEY_FAILED`, where applicable
- ▶ the Dcm cancels the operation as a result of either:
 - ▶ a time-out resulting from reaching the maximum allowed number of `RequestCorrectlyReceivedResponsePending` (0x78) NRCs, if the operation is called in the context of a diagnostic service request



- ▶ a time-out resulting from reaching the maximum allowed number of calls to the operation in question, if the operation is not called in the context of a diagnostic service request
- ▶ protocol preemption

The operation is always called for the first invocation with an `OpStatus` value of `DCM_INITIAL`.

An operation signals that it must be invoked once more by returning the error code `DCM_E_PENDING`. As a result, the `Dcm` shall call the operation in the next `Dcm_MainFunction()` cycle with an `OpStatus` value of `DCM_PENDING` in order to signal such a subsequent call.

During such a chain of calls, the `Dcm` monitors the P2 and P2* timers of the diagnostic session and handles transmission of `RequestCorrectlyReceivedResponsePending` (0x7F 0x27 0x78) NRCs internally. Such an NRC transmission may also be forced by returning a `DCM_E_FORCE_RCRRP` error code where possible (signalled by the availability of this error code). In order to signal a successful transmission of such a forced NRC, the subsequent call to the operation takes place after transmission confirmation with an `OpStatus` value of `DCM_FORCE_RCRRP_OK`.

The `Dcm` also counts the number of `RequestCorrectlyReceivedResponsePending` (0x7F 0x27 0x78) NRCs transmitted in order to cancel the operation call chain if this number reaches the value configured for the `DcmDslDiagRespMaxNumRespPend` parameter. This is performed regardless of the reason (time-out or forced transmission) of the NRC transmission.

An operation is signalled to cancel its operations with a call with an `OpStatus` value of `DCM_CANCEL`. This shall always signal the last call of the asynchronous call chain. The error code returned by the application to such a call is ignored.

The operations:

- ▶ `GetSeed`
- ▶ `CompareKey`

are always present for the `SecurityAccess_<SecurityLevel>` `ClientServerInterface`. Additionally, if the `AttemptCounter` for the security level is configured to be stored to non-volatile memory (`DcmDspSecurity-AttemptCounterEnabled` parameter is set to `TRUE`), this interface shall also contain the operations:

- ▶ `GetSecurityAttemptCounter`
- ▶ `SetSecurityAttemptCounter`

For every security level that implements its operations with this mechanism, the following `RequiredPort` is generated:

`SecurityAccess_<SecurityLevel>`

where `SecurityLevel` is the short name configured for the security level's `DcmDspSecurityRow` entry.

The available codes of the `ClientServerInterface` `SecurityAccess_<SecurityLevel>` are:



```
PossibleErrors { E_NOT_OK = 1, DCM_E_PENDING = 10, E_COMPARE_KEY_FAILED = 11 };
```

Additionally, if the `AttemptCounter` for the security level is configured to be stored to non-volatile memory (`DcmDspSecurityAttemptCounterEnabled` parameter is set to `TRUE`), the following error code is added:

```
PossibleErrors { DCM_E_FORCE_RCRP = 12, };
```

4.4.2.3.2.4.1. Asynchronous ClientServerInterface operations for the GetSeed operation

The `GetSeed` operation of the `SecurityAccess_<SecurityLevel>` `ClientServerInterface` is invoked in order to fulfill the functionality of the `GetSeed` operation. The signature of this operation depends on the value of the `DcmDspSecurityADRSIZE` parameter of the security level's `DcmDspSecurityRow` entry. The signature of this operation does not depend on the value of the `DcmDefaultASRServiceAPI` parameter.

- ▶ If the `DcmDspSecurityADRSIZE` parameter is disabled for the security level's `DcmDspSecurityRow` entry, the `GetSeed` operation shall have the following signature:

```
GetSeed( IN Dcm_OpStatusType OpStatus, OUT uint8
Seed[<DcmDspSecuritySeedSize>], OUT NegativeResponseCodeType ErrorCode,
ERR{E_NOT_OK, DCM_E_PENDING} );
```

where:

- ▶ IN `Dcm_OpStatusType` `OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this operation.
 - ▶ `DCM_PENDING`: This is a subsequent call of this operation after it returned a `DCM_E_PENDING` error code in a previous call. This allows processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_CANCEL`: This is a call for cancellation of the operation. The operation shall terminate all processing and shall reset any internal variables and states to a default value. All results of processing performed from the last call with a `DCM_INITIAL` `OpStatus` up to this call shall be considered null. The error code of this call is not internally considered by the `Dcm`.
- ▶ OUT `uint8 Seed[<DcmDspSecuritySeedSize>]` is a pointer to a buffer where the application must write the generated `securitySeed`. This buffer shall have the maximum size given as the value of the `DcmDspSecuritySeedSize` parameter for the current security level. This data is serialized in the diagnostic service response.
- ▶ OUT `NegativeResponseCodeType ErrorCode` is a pointer at which the application may write an NRC to be transmitted in case the operation is not accepted.

The operation's valid error codes are:

- ▶ `E_OK`: The operation has succeeded and the generated `securitySeed` can be sent to the tester.



- ▶ `E_NOT_OK`: The operation has failed. The NRC returned via the `ErrorCode` parameter is sent to the tester.
- ▶ `DCM_E_PENDING`: Request processing is pending and shall need a further invocation of the operation. Return this error code in order to split the processing of a request into fragments which are not too long to interfere with scheduling.
- ▶ If the `DcmDspSecurityADRSIZE` parameter is enabled for the security level's `DcmDspSecurityRow` entry, the `GetSeed` operation shall have the following signature:

```
GetSeed( IN uint8 SecurityAccessDataRecord[<DcmDspSecurityADRSIZE>], IN Dcm_OpStatusType OpStatus, OUT uint8 Seed[<DcmDspSecuritySeedSize>], OUT NegativeResponseCodeType ErrorCode, ERR{E_NOT_OK, DCM_E_PENDING} );
```

where:

- ▶ `IN uint8 SecurityAccessDataRecord[<DcmDspSecurityADRSIZE>]` is a pointer to a buffer where the `SecurityAccessDataRecord` from the `RequestSeed` diagnostic request is stored. This buffer shall have the maximum size given as the value of the `DcmDspSecurityADRSIZE` parameter of the security level's `DcmDspSecurityRow` entry. This is an auxiliary parameter via which an additional level of security may be implemented in the application by checking the validity of this record and deciding whether to accept or reject the request.
- ▶ `IN Dcm_OpStatusType OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this operation.
 - ▶ `DCM_PENDING`: This is a subsequent call of this operation after it returned a `DCM_E_PENDING` error code in a previous call. This allows processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_CANCEL`: This is a call for cancellation of the operation. The operation shall terminate all processing and shall reset any internal variables and states to a default value. All results of processing performed from the last call with a `DCM_INITIAL` `OpStatus` up to this call shall be considered null. The error code of this call is not internally considered by the `Dcm`.
- ▶ `OUT uint8 Seed[<DcmDspSecuritySeedSize>]` is a pointer to a buffer where the application must write the generated `securitySeed`. This buffer shall have the maximum size given as the value of the `DcmDspSecuritySeedSize` parameter for the current security level. This data is serialized in the diagnostic service response.
- ▶ `OUT NegativeResponseCodeType ErrorCode` is a pointer at which the application may write an NRC to be transmitted if the operation is not accepted.

The operation's valid error codes are:

- ▶ `E_OK`: The operation has succeeded and the generated `securitySeed` can be sent to the tester.
- ▶ `E_NOT_OK`: The operation has failed. The NRC returned via the `ErrorCode` parameter is sent to the tester.



- ▶ **DCM_E_PENDING:** Request processing is pending and shall need a further invocation of the operation. Return this error code in order to split the processing of a request into fragments which are not too long to interfere with scheduling.

4.4.2.3.2.4.2. Asynchronous ClientServerInterface operations for the CompareKey operation

The `CompareKey` operation of the `SecurityAccess_<SecurityLevel>` ClientServerInterface is invoked in order to fulfill the functionality of the `CompareKey` operation. The signature of this operation does not depend on the value of the security level's `DcmDspSecurityADRSIZE` parameter of the `DcmDspSecurityRow` entry. The signature of this operation depends on the value of the `DcmDefaultASRServiceAPI` parameter.

- ▶ If the `DcmDefaultASRServiceAPI` parameter is set to `AUTOSAR_40`, the `CompareKey` operation shall have the following signature:

```
CompareKey( IN uint8 Key[<DcmDspSecurityKeySize>], IN Dcm_OpStatusType OpStatus, ERR{E_NOT_OK, E_PENDING, E_COMPARE_KEY_FAILED} );
```

where:

- ▶ IN `uint8 Key[<DcmDspSecurityKeySize>]` is a pointer to a buffer containing the `securityKey` received from the tester. This buffer shall have the maximum size given as the value of the `DcmDspSecurityKeySize` parameter for the current security level.
- ▶ IN `Dcm_OpStatusType OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this operation.
 - ▶ `DCM_PENDING`: This is a subsequent call of this operation after it returned a `DCM_E_PENDING` error code in a previous call. This allows processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_CANCEL`: This is a call for cancellation of the operation. The operation shall terminate all processing and shall reset any internal variables and states to a default value. All results of processing performed from the last call with a `DCM_INITIAL` `OpStatus` up to this call shall be considered null. The error code of this call is not internally considered by the `Dcm`.

The operation's valid error codes are:

- ▶ `E_OK`: The operation has succeeded. A positive response is sent to the tester and the ECU transitions to the requested security level.
- ▶ `E_NOT_OK`: The operation has failed. The `InvalidKey` (0x35) NRC is sent to the tester.
- ▶ `DCM_E_PENDING`: Request processing is pending and shall need a further invocation of the operation. Return this error code in order to split the processing of a request into fragments which are not too long to interfere with scheduling.
- ▶ `DCM_E_COMPARE_KEY_FAILED`: The operation has failed. The `InvalidKey` (0x35) NRC is sent to the tester and the `AttemptCounter` of the security level is incremented.



- ▶ If the `DcmDefaultASRServiceAPI` parameter is set to `AUTOSAR_42`, the `CompareKey` operation shall have the following signature:

```
CompareKey( IN uint8 Key[<DcmDspSecurityKeySize>], IN Dcm_OpStatusType OpStatus,
            OUT Dcm_NegativeResponseType ErrorCode, ERR{E_NOT_OK, E_PENDING, E_COMPARE_KEY_FAILED}); ;
```

where:

- ▶ `IN uint8 Key[<DcmDspSecurityKeySize>]` is a pointer to a buffer containing the `securityKey` received from the tester. This buffer shall have the maximum size given as the value of the `DcmDspSecurityKeySize` parameter for the current security level.
- ▶ `IN Dcm_OpStatusType OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this operation.
 - ▶ `DCM_PENDING`: This is a subsequent call of this operation after it has returned a `DCM_E_PENDING` error code in a previous call. This allows processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_CANCEL`: This is a call for cancellation of the operation. The operation shall terminate all processing and shall reset any internal variables and states to a default value. All results of processing performed from the last call with a `DCM_INITIAL` `OpStatus` up to this call shall be considered null. The error code of this call is not internally considered by the `Dcm`.
- ▶ `OUT Dcm_NegativeResponseType ErrorCode` is a pointer at which the application may write an NRC to be transmitted if the operation is not accepted.

The operation's valid error codes are:

- ▶ `E_OK`: The operation has succeeded. A positive response is sent to the tester and the ECU transitions to the requested security level.
- ▶ `E_NOT_OK`: The operation has failed. The NRC returned via the `ErrorCode` parameter is sent to the tester.
- ▶ `DCM_E_PENDING`: Request processing is pending and shall need a further invocation of the operation. Return this error code in order to split the processing of a request into fragments which are not too long to interfere with scheduling.
- ▶ `DCM_E_COMPARE_KEY_FAILED`: The operation has failed. The `InvalidKey` (0x35) NRC is sent to the tester and the `AttemptCounter` of the security level is incremented.

4.4.2.3.2.4.3. Asynchronous ClientServerInterface operations for the GetSecurityAttemptCounter operation

The `GetSecurityAttemptCounter` operation of the `SecurityAccess_<SecurityLevel>` ClientServerInterface is invoked in order to reload the value of the `AttemptCounter` for the security level from non-



volatile memory. This operation is only required if the AttemptCounter for the security level is configured to be stored to non-volatile memory (`DcmDspSecurityAttemptCounterEnabled` parameter is set to TRUE). The signature of this operation does not depend on the value of the `DcmDspSecurityADRSIZE` parameter of the security level's `DcmDspSecurityRow` entry. The signature of this operation does not depend on the value of the `DcmDefaultASRServiceAPI` parameter.

The `GetSecurityAttemptCounter` operation shall have the following signature:

```
GetSecurityAttemptCounter( IN Dcm_OpStatusType OpStatus, OUT uint8 AttemptCounter,  
ERR{E_NOT_OK, E_PENDING} );
```

where:

- ▶ IN `Dcm_OpStatusType` `OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this operation.
 - ▶ `DCM_PENDING`: This is a subsequent call of this operation after it returned a `DCM_E_PENDING` error code in a previous call. This allows processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_CANCEL`: This is a call for cancellation of the operation. The operation shall terminate all processing and shall reset any internal variables and states to a default value. All results of processing performed from the last call with a `DCM_INITIAL` `OpStatus` up to this call shall be considered null. The error code of this call is not internally considered by the `Dcm`.
- ▶ OUT `uint8` `AttemptCounter` is a pointer at which the application writes the retrieved value of the `AttemptCounter`.

The operation's valid error codes are:

- ▶ `E_OK`: The operation has succeeded.
- ▶ `E_NOT_OK`: The operation has failed.
- ▶ `DCM_E_PENDING`: Request processing is pending and shall need a further invocation of the operation. Return this error code in order to split the processing of a request into fragments which are not too long to interfere with scheduling.

4.4.2.3.2.4.4. Asynchronous ClientServerInterface operations for the `SetSecurityAttemptCounter` operation

The `SetSecurityAttemptCounter` operation of the `SecurityAccess_{SecurityLevel}` ClientServerInterface is invoked in order to store the value of the `AttemptCounter` for the security level to non-volatile memory. This operation is only required if the `AttemptCounter` for the security level is configured to be stored to non-volatile memory (`DcmDspSecurityAttemptCounterEnabled` parameter is set to TRUE). The signature of this operation does not depend on the value of the `DcmDspSecurityADRSIZE` parameter of the



security level's `DcmDspSecurityRow` entry. The signature of this operation does not depend on the value of the `DcmDefaultASRServiceAPI` parameter.

The `SetSecurityAttemptCounter` operation shall have the following signature:

```
SetSecurityAttemptCounter( IN Dcm_OpStatusType OpStatus, IN uint8 AttemptCounter,
ERR{E_NOT_OK, DCM_E_PENDING, DCM_E_FORCE_RCRP} );
```

where:

- ▶ `IN Dcm_OpStatusType OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this operation.
 - ▶ `DCM_PENDING`: This is a subsequent call of this operation after it returned a `DCM_E_PENDING` error code in a previous call. This allows processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_CANCEL`: This is a call for cancellation of the operation. The operation shall terminate all processing and shall reset any internal variables and states to a default value. All results of processing performed from the last call with a `DCM_INITIAL` `OpStatus` up to this call shall be considered null. The error code of this call is not internally considered by the `Dcm`.
- ▶ `uint8 AttemptCounter` is the value of the `AttemptCounter` that the application shall store in non-volatile memory.

The operation's valid error codes are:

- ▶ `E_OK`: The operation has succeeded.
- ▶ `E_NOT_OK`: The operation has failed.
- ▶ `DCM_E_PENDING`: Request processing is pending and shall need a further invocation of the operation. Return this error code in order to split the processing of a request into fragments which are not too long to interfere with scheduling.
- ▶ `DCM_E_FORCE_RCRP`: Request processing is pending and shall need a further invocation of the operation. Additionally, the application requests the immediate transmission of a `RequestCorrectlyReceivedResponsePending` (0x7F 0x27 0x78) NRC.

NOTE



No hierarchical relationship between security levels

There is no hierarchical relationship between different security levels. The functionalities of a security level are not automatically available in another security level of a numerically lower or higher value. There is no "higher" or "lower" security level. Such conditions must be configured explicitly.

The security levels that are available to an ECU, i.e. the set of security levels to which the ECU can switch, are defined in the **Security Levels** table. By default, the ECU boots up without security level, i.e. in the `LOCKED` state.

The following parameters are configured in a security level:



- ▶ The `DcmDspSecurityLevel` parameter is a numeric security level identifier. In order to transition to this security level, the subfunction IDs to be used for the subfunctions of the `SecurityAccess` (0x27) service are determined as follows:
 - ▶ $2 * \text{SecurityLevel} - 1$ for the `RequestSeed` subfunction corresponding to the security level
 - ▶ $2 * \text{SecurityLevel}$ for the `SendKey` subfunction corresponding to the security level
- ▶ the application interfaces which are used in order to provide the functionality of the security level
- ▶ the `AttemptCounter` limit value and `DelayTimer` values for the security level
- ▶ forced `DelayTimer` values

For information on how to configure security levels, see [Section 4.4.3.3, “Configuring security levels”](#).

4.4.2.4. Bootloader interaction

Bootloader interaction in the `Dcm` is handled mainly by the diagnostic session management functionality. You can configure the `Dcm` to trigger a jump to the bootloader and to process a jump from the bootloader with parameters that were stored before the reset. To enable this behavior, configure the following:

- ▶ At least one diagnostic session for the jump to the bootloader.
- ▶ The application interface to store/restore the `ProgrammingConditions`.

To jump to the bootloader upon switch to the diagnostic session, the `DcmEcuReset ModeDeclarationGroupPrototype` is switched to the requested bootloader reset mode and then switching it to the `EXECUTE` mode.

The jump to the bootloader upon switch to the diagnostic session is executed by the following:

- ▶ If the AUTOSAR 4.0.2 interaction via `BswM` is used: the reset functionality that is implemented in the `BswM` module is executed. This contains calling the `BswM_Dcm_RequestResetMode()` API for the requested bootloader reset mode and then calling the same `BswM_Dcm_RequestResetMode()` API for the `DCM_RESET_EXECUTION` mode.
- ▶ If the AUTOSAR 4.0.3 interaction via `Rte` is used: the `DcmEcuReset ModeDeclarationGroupPrototype` is switched to the requested bootloader reset mode and then switched to the `EXECUTE` mode.

4.4.2.4.1. Notification of the ECU after the reset

If a jump to the bootloader occurs, the positive response to the `DiagnosticSessionControl` (0x10) service that triggered the jump is sent to the tester after the ECU restarts. This positive response notifies the tester when the ECU is ready to process new commands and ensures synchronization. In certain cases, the ECU needs to transmit a `RequestCorrectlyReceivedResponsePending` (0x7F 0x10 0x78) NRC before it executes the reset. This behavior prevents the tester from timing out because of a lack of responses while the reset is executed. This behavior is controlled by the `DcmSendRespPendOnTransToBoot` parameter.



4.4.2.4.2. Application interfaces for storing the ProgrammingConditions

The ECU can store pre-reset conditions, e.g. data that is necessary to transmit a positive response to the `DiagnosticSessionControl` (0x10) service that triggered the jump to the bootloader. For this, a set of variables called `ProgrammingConditions` are used. These variables are written by the `Dcm` by calling the `Dcm_SetProgConditions()` API and are read back after the reboot by calling the `Dcm_GetProgConditions()` API. The functionality that stores these `ProgrammingConditions` in non-volatile memory is implemented in the application code, i.e. in the software component or complex device driver. Therefore, you do not need to configure this feature. However, the code of these two APIs must be present in the application.

4.4.2.4.3. Bootloader interaction in AUTOSAR 4.0.3 and AUTOSAR 4.0.2

You can configure the jump to the bootloader according to AUTOSAR 4.0.2 or AUTOSAR 4.0.3.

Rte interaction for AUTOSAR 4.0.3 or higher

A diagnostic session that triggers a jump to/from the bootloader is basically handled by the `Rte` as a regular switch to another diagnostic session. Therefore, the same configuration rules apply as for switching the diagnostic session. The only difference is that besides a switch from the `DcmDiagnosticSessionControl` `ModeDeclarationGroupPrototype` to the mode corresponding to the requested diagnostic session, the `ModeDeclarationGroupPrototype` `DcmEcuReset` is also switched to a mode which corresponds to the kind of jump that is configured for the diagnostic session.

You can configure the `Dcm` to either jump to the manufacturer bootloader or the supplier bootloader:

- ▶ If the `Dcm` is configured to jump to the manufacturer bootloader, the `ModeDeclarationGroupPrototype` `DcmEcuReset` is switched to the `JUMPTOBOOTLOADER` mode.
- ▶ If the `Dcm` is configured to jump to the system supplier bootloader, the `ModeDeclarationGroupPrototype` `DcmEcuReset` is switched to the `JUMPTOSYSSUPPLIERBOOTLOADER` mode.

This first switch of the `ModeDeclarationGroupPrototype` `DcmEcuReset` is then used by basic software modules as the `BswM` or complex device drivers to prepare a jump to the bootloader. Therefore, if this behavior must be executed by a basic software module or a complex device driver to prepare a jump to the bootloader, the `Rte` must be configured. The `Dcm` provides a `Provided Mode Group` for the `DcmEcuReset` `ModeDeclarationGroupPrototype`, similarly to the `DcmDiagnosticSessionControl` `ModeDeclarationGroupPrototype`. A participating basic software module or complex device driver is also connected to such a `Provided Mode Group` by the `Rte`.

BswM interaction for AUTOSAR 4.0.2 or lower

If the AUTOSAR 4.0.2 interaction via `BswM` is used, a diagnostic session that triggers a jump to/from the bootloader is basically handled by the `BswM` as a regular switch to another diagnostic session. Therefore, the same configuration rules as for switching to another diagnostic session apply. The only difference is that besides the call to the `BswM_Dcm_RequestSessionMode()` API to request the switch to another diagnostic session, the `BswM_Dcm_RequestResetMode()` API is called.



This first call is used in order for the `BswM` to prepare a jump to the bootloader. Therefore, if this behaviour must be executed by a basic software module or a complex device driver to prepare a jump to the bootloader, the same steps as for configuring the regular switch to a diagnostic session must be performed again to configure the `BswM` to perform the necessary actions.

You can configure the `Dcm` to either jump to the manufacturer bootloader or the supplier bootloader:

- ▶ If the `Dcm` is configured to jump to the manufacturer bootloader, the `BswM_Dcm_RequestResetMode()` API is called with a value of `DCM_BOOTLOADER_RESET` for the `RequestedMode` parameter.
- ▶ If the `Dcm` is configured to jump to the system supplier bootloader, the `BswM_Dcm_RequestResetMode()` API is called with a value of `DCM_SS_BOOTLOADER_RESET` for the `RequestedMode` parameter.

4.4.2.5. Diagnostic services

The `Dcm` provides a number of UDS and OBD diagnostic services that are internally-managed, i. e. the service handler is provided in the EB tresos Diagnostic Stack. The internally-managed services are listed in [Section 2.3.1.2, “UDS services”](#) and [Section 2.3.1.3, “OBD services”](#). If you require diagnostic services that are not internally-managed, you can create externally-managed diagnostic services or subfunctions. If a diagnostic service is internally managed but a subfunction is not, you can configure an externally-managed subfunction handler to be called by an internally-managed diagnostic service handler.

For information on how to configure diagnostic services, see [Section 4.4.3.4, “Configuring diagnostic services”](#).

4.4.2.5.1. Diagnostic service access

You can configure a diagnostic service or subfunction to be only accessible under specific circumstances. The following list contains all checks that can be performed before allowing access to a diagnostic service. The checks are performed in the following order:

1. Check manufacturer-specific access conditions, see [Section 4.4.2.5.2, “Manufacturer-specific or supplier-specific access verification”](#).

2. Check if the diagnostic service is run in an allowed diagnostic session.

If a request to a diagnostic service is received in a diagnostic session in which the requested service is not allowed to run, a `ServiceNotSupportedInActiveSession` (0x7F) NRC is sent back to the tester.

3. Check if the diagnostic service is run at an allowed security level.

If a request to a diagnostic service is received at a security level at which the requested diagnostic service is not allowed, a `SecurityAccessDenied` (0x33) NRC is sent back to the tester.

4. Check if the diagnostic service meets mode rules.

If a request to a diagnostic service is received and the evaluation of the mode condition fails, a NRC is sent back to the tester. The NRC is defined by the mode condition.



5. Check supplier-specific access conditions, see [Section 4.4.2.5.2, “Manufacturer-specific or supplier-specific access verification”](#).

6. Check if the subfunction is supported by the diagnostic service.

If a request to a subfunction is received in a diagnostic session in which the requested subfunction is not supported, a `ServiceNotSupportedInActiveSession` (0x7F) NRC is sent back to the tester.

7. Check if the subfunction is run at an allowed security level.

If a request to a subfunction is received at a security level at which the requested subfunction is not allowed, a `SecurityAccessDenied` (0x33) NRC is sent back to the tester.

8. Check if the subfunction meets mode rules.

If a request to a subfunction is received and the evaluation of the mode condition fails, a NRC is sent back to the tester. The NRC is defined by the mode condition.

NOTE

Conditions for subfunction-related checks



All subfunction-related checks are only performed under the following conditions:

- ▶ The parent diagnostic service is internally managed.
- ▶ In the parent diagnostic service, the `DcmDsdSidTabSubfuncAvail` parameter is enabled.

Access of subfunctions

The diagnostic sessions in which a subfunction is available must be a subset of the diagnostic sessions under which the parent diagnostic service is available. If no diagnostic sessions are configured for the parent diagnostic service, the diagnostic service is by default available in any diagnostic session. Therefore, any diagnostic session is allowed to be configured for its subfunctions. The diagnostic sessions which are referenced for the subfunction specify in which diagnostic sessions this subfunction is allowed to run. If no diagnostic sessions are configured for the subfunction, it is allowed to run in the same diagnostic sessions as the parent diagnostic service. The same method applies to the configuration of the security levels.

4.4.2.5.2. Manufacturer-specific or supplier-specific access verification

The `Dcm` allows you to configure manufacturer-specific checks. These conditions are checked before the other access checks for the diagnostic service, i.e. allowed diagnostic session, mode condition, security level, and subfunction checks. The `Dcm` also provides a mechanism that notifies the application when the positive or negative response for service processing is transmitted to the bus. This mechanism is performed through `RequiredPorts` that implements the interface `ClientServerInterface ServiceRequestNotification`.

This interface provides the indication operation and the confirmation operation.

Indication operation



The indication operation has the following structure:

```
Indication(
    IN uint8 SID,
    IN uint8 RequestData[<Data size>],
    IN uint16 DataSize,
    IN uint8 ReqType,
    IN uint16 SourceAddress,
    ,
    OUT NegativeResponseType* ErrorCode
    ,
    ERR{E_REQUEST_NOT_ACCEPTED, E_NOT_OK});
```

The indication operation allows the application to inspect the following data:

- ▶ Service ID (SID)
- ▶ The request message, beginning with the first data byte (if any) after the service ID (requestData, dataSize).
- ▶ The type of the request addressing, i.e. physical or functional (ReqType).
- ▶ The tester source address that corresponds to the `DcmDslProtocolRxTesterSourceAddr` parameter that is configured for the main connection with the RxPdul on which the request was received (SourceAddress).

Based on this indication operation, the application can make one of the following decisions:

- ▶ Accept the request.
- ▶ Reject the request and transmit an adequate NRC.
- ▶ Reject the request with no answer.
- ▶ Modify the request message if needed.

Confirmation operation

The indication operation has the following structure:

```
Confirmation(
    IN uint8 SID,
    IN uint8 ReqType,
    IN uint16 SourceAddress,
    IN ConfirmationStatusType ConfirmationStatus
    ,
    ERR{E_NOT_OK});
```



The confirmation operation is called after the response to the diagnostic service was transmitted, whether it is suppressed or not. The confirmation operation informs the application about the following data:

- ▶ Which service ID transmitted a response (SID).
- ▶ For which type of request this response was transmitted, i.e. physical or functional (ReqType).
- ▶ The tester source address that corresponds to the `DcmDslProtocolRxTesterSourceAddr` parameter that is configured for the main connection with the RxPduld on which the request was received (SourceAddress).
- ▶ The result of the transmission of the service response, i.e. if it was successful or unsuccessful (ConfirmationStatus).

The `Dcm` also allows you to configure supplier-specific checks. The supplier-specific checks are performed the same way as the manufacturer-specific checks. However, the supplier-specific checks are performed after the other access checks for the diagnostic service, i.e. allowed diagnostic session, mode condition, security level, but before the checks for the subfunctions. The supplier-specific checks are performed before the request itself is dispatched to the diagnostic service processor, i.e. before the service handler functionality is invoked. It also provides a notification mechanism similar to the manufacturer-specific notification. This is handled through `RequiredPorts` that implement the same `ServiceRequestNotification.ClientServerInterface` interface as in the manufacturer case. The only differences are:

- ▶ The supplier indication operations are called after all other diagnostic service-related checks.
- ▶ The supplier indication operations are called before all other subfunction-related checks.
- ▶ The supplier confirmation operations are called after the manufacturer confirmation operations.

4.4.2.5.3. Disable diagnostic services while driving

If diagnostic services need to be disabled under some conditions, for example while the car is driving, one of the following features can be used:

- ▶ the manufacturer/supplier-specific verification
- ▶ the mode rule verification for diagnostic services

For configuration advice, see [Section 4.4.3.4.4.8, “Disabling writing memory diagnostic services while driving”](#).

4.4.2.5.4. Negative and positive response suppression

Suppression of positive responses is handled internally for diagnostic service handlers, in accordance with ISO14229-1. For diagnostic services that support subfunctions, the most significant bit of the subfunction ID is the `suppressPosRspMsgIndicationBit`. If this bit is set in the request, the positive response is suppressed. The processing in this case is still handled the same way as for unsuppressed responses. However, the response is not transmitted.



If a `RequestCorrectlyReceivedResponsePending` (0x7F ServiceID 0x78) NRC is transmitted while the request is processed, negative or positive responses are no longer suppressed.

NRCs for functional requests can be suppressed according to ISO14229-1:2013 or ISO14229-1:2006. The `DcmNRCPolicyISOVersion` parameter defines the ISO version that is applied.

In ISO14229_2006, the following NRCs are suppressed if they are received on a functional RxPduld:

- ▶ `ServiceNotSupported` (0x11)
- ▶ `SubFunctionNotSupported` (0x12)
- ▶ `RequestOutOfRange` (0x31)

In ISO14229_2013, the following NRCs are suppressed if they are received on a functional RxPduld:

- ▶ `ServiceNotSupported` (0x11)
- ▶ `SubFunctionNotSupported` (0x12)
- ▶ `RequestOutOfRange` (0x31)
- ▶ `SubfunctionNotSupportedInActiveSession` (0x7E)
- ▶ `ServiceNotSupportedInActiveSession` (0x7F)

For internally-managed diagnostic services:

Positive response suppression is handled internally if the internally-managed service handler is configured with subfunction support.

For externally-managed diagnostic services:

Positive response suppression for externally-managed diagnostic services is handled by the externally-managed service handler.

4.4.2.5.5. Internally managed diagnostic service handlers and subfunction handlers

The `Dcm` allows you to configure your own diagnostic service handlers. This allows you to use diagnostic services and subfunctions that are either not internally managed or are customized according to specific project requirements.

For most diagnostic services that support subfunctions according to ISO14229-1, you can use your own subfunction handler and call it within an internally-managed service handler. In this case, the internally-managed service handler serves as a dispatcher for internally and externally-managed subfunction handlers.

4.4.2.5.6. Asynchronous service processing via separate task

You can use asynchronous service processing to increase the amount of data that is processed. A single invocation of a service handler function must return within very constrained time limits because of the periodic



calls to the `Dcm_MainFunction()` API. Therefore, if very tight scheduling requirements are imposed, the execution of a diagnostic service that processes large amounts of data may not be completed before the next call of the `Dcm_MainFunction()` API. This leads to a call to the `Dcm_MainFunction()` API while the previous call has not yet returned. Asynchronous service processing allows to call the service handler function via a task different than the task that calls the `Dcm_MainFunction()` API. The `Dcm_MainFunction()` API does not call the actual service handler every `DcmTaskTime` cycle. Instead, the `Dcm_MainFunction()` API interrogates the asynchronous service handler task about the state of the current run of the service handler function. If a run of the service handler is still not finished, the `Dcm_MainFunction()` API behaves just like the service handler has instantly returned a `DCM_E_PENDING` error code. If a run of the service handler is finished, the `Dcm_MainFunction()` API receives the result, i.e. an error code, of this run from the asynchronous service handler task. The `Dcm_MainFunction()` API then behaves like it called the service handler function and received this result.

Asynchronous service processing is selectable per diagnostic service. It is realized by generating an `InternalTriggerOccurredEvent` and mapping a `RunnableEntity` to be run when this event is triggered. If asynchronous service processing is enabled for a service handler, this event is triggered every time the service handler function needs to run one more time, instead of issuing a direct call. The `RunnableEntity` invokes an API that calls the service handler function from the context of a different task. The `Dcm_MainFunction()` then keeps track of the state of this service handler execution step and can obtain the results of the execution once the `RunnableEntity` finishes the call to the service handler function.

If asynchronous service handling is enabled for a service handler, the `Dcm` generates in its basic software module description a `BswInternalTriggerOccuredEvent` event named `BswInternalTriggerOccuredEvent_Async`. You must map this `BswInternalTriggerOccuredEvent` event to a task that has a lower priority than the task to which the `BswTimingEvent TimingEvent_MainFunction` of the `Dcm` is mapped. The reason for this is that the asynchronous service handler must run in a task of a lower priority than the task that runs the `Dcm_MainFunction()` API that sets the `BswInternalTriggerOccuredEvent` event. This way, the service handler is allowed to run in the low priority task decoupled from the task of higher priority. Otherwise the service handler function would run in a higher priority task that cannot be preempted by the task that calls the `Dcm_MainFunction()` API. This would lead to both nullifying the advantage of asynchronous service handling and a disruption of the `Dcm`'s timing behaviour, as the time base for the module is the frequency in which the `Dcm_MainFunction()` API is called. This frequency is every `DcmTaskTime` seconds.

Besides this restriction in priority, an additional restriction applies to the task to which the `BswInternalTriggerOccuredEvent` event is mapped and, therefore, runs the asynchronous diagnostic service handlers. This restriction only applies to basic tasks.

If you use a basic task, the maximum number of activations that are entered in the `OsTaskActivation` parameter must be at least 2. The rationale for this restriction is that the asynchronous service processor task may continue to run for a short time after the service processing is completed and the response to the service request is sent to the tester. This behavior occurs because this task is preempted before it can terminate. If this behavior occurs, the tester can further request another service from the `Dcm`. If this second service is also configured to be executed asynchronously the asynchronous service processor task is activated before it is terminated. This leads to an error if the maximum number of activations is 1.



4.4.2.5.7. ReadScalingDataByIdentifier (0x24) UDS service

With the UDS service `ReadScalingDataByIdentifier` (0x24), the client can request scaling information of a record identified by a data identifier from the server.

The `Dcm` module offers an internal implementation of the UDS service `ReadScalingDataByIdentifier` (0x24). This service is implemented according to the requirements of the following standards:

- ▶ AUTOSAR 4.0.3
- ▶ AUTOSAR CP R20-11
- ▶ ISO_14229-1:2013
- ▶ ISO_14229-1:2020: Regarding the specified negative response codes (NRCs) stated in *Table 210 - Supported negative response codes*, the following deviation applies:

The `Dcm` does not support the NRC `authenticationRequired` (0x34) because the UDS service `Authentication` (0x29) is currently not supported.

NOTE**`Dcm` behavior for dynamically defined DIDs, range DIDs, and DIDs that reference other DIDs**

The `Dcm` does not support the service request `ReadScalingDataByIdentifier` (0x24) for the following DIDs:

- ▶ a dynamically defined data identifier, i.e. for a DID value that has the configuration parameter `DcmDspDidDynamicallyDefined` enabled,
- ▶ a range data identifier, i.e. for a DID value that is configured via the configuration parameter `DcmDspDidRange` and not via the configuration parameter `DcmDspDidIdentifier`.

Consequently, the `Dcm` transmits a negative response with an NRC equal to the value 0x31 (`requestOutOfRange`) if the service `ReadScalingDataByIdentifier` (0x24) is requested for a dynamically defined DID or for a range DID.

If the service `ReadScalingDataByIdentifier` (0x24) is requested for a DID that contains reference(s) to other DID(s), i.e. for a DID that has at least one valid reference to another DID configured in the configuration container `DcmDspDidRef`, the `Dcm` simply ignores the referenced DID(s). That means the `Dcm` does not invoke the `Xxx_GetScalingInformation()` callback for the referenced DID(s).

For more information on the UDS service `ReadScalingDataByIdentifier` (0x24), see the following specifications:

- ▶ Specification of Diagnostic Communication Manager, AUTOSAR 4.0.3, section *7.4.2.7 UDS Service 0x24 - ReadScalingDataByIdentifier*



- ▶ Specification of Diagnostic Communication Manager, AUTOSAR CP R20-11, section 7.6.2.7 Service `0x24 - ReadScalingDataByIdentifier`
- ▶ ISO14229-1:2013, section 10.4 `ReadScalingDataByIdentifier (0x24)` service
- ▶ ISO14229-1:2020, section 11.4 `ReadScalingDataByIdentifier (0x24)` service

See [Section 4.4.3.4.6, “Configuring the `ReadScalingDataByIdentifier \(0x24\)` UDS service”](#) for configuration advice.

4.4.2.5.8. SecurityAccess (0x27) UDS service

The `Dcm` module offers an internal implementation of the `SecurityAccess (0x27)` UDS service. This service is implemented according to the requirements of the following standards:

- ▶ AUTOSAR 4.0.3
- ▶ AUTOSAR 4.3 with regards to the following:
 - ▶ Security `DelayTimer` and `AttemptCounter` functionality
 - ▶ Selectably, interface compatibility for the `Xxx_GetSeed()` and `Xxx_CompareKey()` interfaces
- ▶ ISO14229-1:2006
- ▶ ISO14229-1:2013 with regards to the following aspects:
 - ▶ Security `DelayTimer` and `AttemptCounter` functionality
 - ▶ Negative Response Code (NRC) order

Additionally, synchronous function callouts and `ClientServerInterface` operations can be used for the implementation of `SecurityAccess (0x27)` service-related operations. For this, the following additional interface types are provided in the `DcmDspSecurityUsePort` interface selection parameter:

- ▶ `USE_SYNCH_CLIENT_SERVER`
- ▶ `USE_SYNCH_FNC`

With regards to the security `DelayTimer` and `AttemptCounter` functionality, the following extensions are provided:

- ▶ The ability to specify infinite `DelayTimer` values upon reaching the `AttemptCounter` limit
- ▶ The ability to specify a unified `DelayTimer`. This allows to start the `DelayTimer` for all security levels by starting one `DelayTimer`.
- ▶ Selectable permanent locking of security levels upon failed retrieval of `AttemptCounter` value stored in non-volatile memory
- ▶ Selectable start of security level `DelayTimers` on boot-time
- ▶ Selectable reset of the security level's `AttemptCounter` upon `DelayTimer` expiration



For more information on the `SecurityAccess` (0x27) UDS service, see the following specifications:

- ▶ AUTOSAR 4.0.3, section *7.4.2.8 UDS Service 0x27 - SecurityAccess*
- ▶ AUTOSAR 4.3, section *7.5.2.8 Service 0x27 - SecurityAccess*
- ▶ ISO14229-1:2006, section *9.4 SecurityAccess (27 hex) service*
- ▶ ISO14229-1:2013, section *9.4 SecurityAccess (0x27) service*

4.4.2.5.9. CommunicationControl (0x28) UDS service

The `Dcm` module offers an internal implementation of the `CommunicationControl` (0x28) UDS service. This service is implemented according to the requirements of the following standards:

- ▶ AUTOSAR 4.0.3 with enabled `ModeDeclaration` for `Rte` interaction
- ▶ Alternatively, you can use the AUTOSAR 4.0.2 legacy `BswM` interaction instead of AUTOSAR 4.0.3 `Rte` interaction
- ▶ AUTOSAR 4.3 with regards to the support for the following subfunctions:
 - ▶ `enableRxAndDisableTxWithEnhancedAddressInformation` (0x04)
 - ▶ `enableRxAndTxWithEnhancedAddressInformation` (0x05)
- ▶ ISO14229-1:2006
- ▶ ISO14229-1:2013, specifically the following subfunctions:
 - ▶ `enableRxAndDisableTxWithEnhancedAddressInformation` (0x04)
 - ▶ `enableRxAndTxWithEnhancedAddressInformation` (0x05)

The `Dcm` supports internal subfunction handler implementations for the following subfunctions:

- ▶ `enableRxAndTx` (0x00)
- ▶ `enableRxAndDisableTx` (0x01)
- ▶ `disableRxAndEnableTx` (0x02)
- ▶ `disableRxAndTx` (0x03)
- ▶ `enableRxAndDisableTxWithEnhancedAddressInformation` (0x04) (only FP2)
- ▶ `enableRxAndTxWithEnhancedAddressInformation` (0x05) (only FP2)

For more information about the UDS service `UDS Service CommunicationControl` (0x28) and its included subfunctions, see the following specifications:

- ▶ AUTOSAR 4.0.3, section *7.4.2.22 UDS Service 0x28 – CommunicationControl*
- ▶ AUTOSAR 4.3, section *7.5.2.9 Service 0x28 - CommunicationControl*
- ▶ ISO14229-1:2006, section *9.5 CommunicationControl (28 hex) service*



- ▶ ISO14229-1:2013, section 9.5 *CommunicationControl (0x28) service*

4.4.2.5.10. RoutineControl (0x31) UDS service

A `diagnostic routine` is a routine (a defined sequence of steps) that is embedded in an electronic control unit and that may be started by a server upon a request from the client.

The `RoutineControl (0x31)` UDS service is used by the client to execute a diagnostic routine and obtain any relevant results.

Typical usage of diagnostic routines may include functionality such as:

- ▶ erasing memory
- ▶ resetting or learning adaptive data
- ▶ running a self-test
- ▶ overriding the normal server control strategy
- ▶ controlling a server value to change over time including predefined sequences

The `Dcm` module offers an internal implementation of the `RoutineControl (0x31)` UDS service. This service is implemented according to the requirements of the following standards:

- ▶ AUTOSAR 4.0.3
- ▶ AUTOSAR 4.3 regarding interface compatibility

This means that the following interface operations:

- ▶ `Xxx_Start()`
- ▶ `Xxx_Stop()`
- ▶ `Xxx_RequestResults()`

can optionally be used in compliance with AUTOSAR 4.3. specifications regarding:

- ▶ operation parameters
- ▶ operation possible error codes
- ▶ ISO14229-1:2006
- ▶ ISO14229-1:2013 with regards to the Negative Response Code (NRC) order, with the following deviations:
 - ▶ The `Dcm` performs mode condition check in addition to ISO 14229-1/2013 specifications.
 - ▶ The `Dcm` passes the responsibility of the `routineControlOptionRecord` validity check to the application.

Additional features:



- ▶ The configuration parameter `DcmDspRoutineVariableLengthInBytes` allows you to configure the `currentDataLength` (as specified in AUTOSAR 4.0.3) or `variableDataLength` (as specified in AUTOSAR 4.3) to be represented either in **bits** or in **bytes**.
- ▶ The `Dcm` offers the possibility to use application-defined interfaces for each unsupported (i.e. unconfigured or unused) `DcmDspRoutineIdentifier`. The `Dcm` calls these interfaces instead of transmitting a negative response.

For information on how to configure these application-defined interfaces, see [Section 4.4.3.4.9.4, “Configuring application-defined interfaces for unsupported `DcmDspRoutineIdentifiers`”](#).

- ▶ The `Dcm` offers the possibility to use an application-defined interface in order to insert the `RoutineInfo` byte in the diagnostic response as defined in ISO 14229-1/2013.

For information on how to configure this application-defined interface, see [Section 4.4.3.4.9.5, “Configuring the application-defined interface for inserting the `RoutineInfo` byte in the diagnostic response as defined in ISO 14229-1/2013”](#).

For more information on the `RoutineControl` (0x31) UDS service, see the following specifications:

- ▶ Specification of Diagnostic Communication Manager, AUTOSAR 4.0.3, section *7.4.2.13 UDS Service 0x31 - RoutineControl*
- ▶ Specification of Diagnostic Communication Manager, AUTOSAR 4.3, section *7.5.2.14 Service 0x31 - RoutineControl*
- ▶ ISO14229-1:2006, section *13.2 RoutineControl (31 hex) service*
- ▶ ISO14229-1:2013, section *13.2 RoutineControl (0x31) service*

4.4.2.5.11. Endianness interpretation for DID signals

The `Dcm` module offers endianness interpretation for `DcmDspData` with sender/receiver access for `ReadDataByIdentifier` (0x22) and `WriteDataByIdentifier` (0x2E) UDS services. The endianness interpretation is implemented according to the requirements of the following standards:

- ▶ AUTOSAR 4.2.1 for DID signals with bit types, atomic types and data array types
- ▶ AUTOSAR 4.3 for DID signals with atomic types and data array types. The signal size and signal start position must be given in bits instead of bytes:
 - ▶ `DcmDspDataSize` configures the signal size and must be the size of the atomic type or a multiple of array element size in case of data array type
 - ▶ `DcmDspDidDataPos` configures the signal start position and must be a multiple of 8 bit

**TIP****It is recommended to use signal configuration according to AUTOSAR 4.3**

The mapping of big-endian signals in the DID is easier to configure and to maintain as byte-aligned signals than as bit-aligned signals.

Chapter 2.4 in OSEK/VDX Communication Version 3.0.3 defines a different mapping of big-endian signals and little-endian signals in the DID relative to `DcmDspDidDataPos`.

- ▶ Big-endian signals start at `DcmDspDidDataPos` and grow towards lower addresses.
- ▶ Little-endian signals start at `DcmDspDidDataPos` and grow towards higher addresses.

Opaque signals start at `DcmDspDidDataPos` and grow towards higher addresses.

NOTE**Calculation of signal start position**

When signals with different endianness belong to the same DID, the calculation of the signal start position must take into account the endianness-specific mapping in the DID to avoid signal overlapping. There is no check for signal overlapping. `DcmDspDataEndianness` and `DcmDspDataDefaultEndianness` are not applicable to byte and byte array signal data, but also in this case they determine how the signal is mapped relative to signal start position.

[Figure 4.4, “Byte-aligned signals”](#) shows a byte-aligned little-endian signal and a byte-aligned big-endian signal. The signals have the same size and are mapped to the same location in the DID. The little-endian signal starts at bit position 16. The big-endian signal starts at bit position 40.

		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Low Address	Byte 0	7	6	5	4	3	2	1	0
	Byte 1	15	14	13	12	11	10	9	8
Element 1		23	22	21	20	19	18	17	16
Low Byte	Byte 2	7	6	5	4	3	2	1	0
Element 1		31	30	29	28	27	26	25	24
High Byte	Byte 3	15	14	13	12	11	10	9	8
Element 2		39	38	37	36	35	34	33	32
Low Byte	Byte 4	7	6	5	4	3	2	1	0
Element 2		47	46	45	44	43	42	41	40
High Byte	Byte 5	15	14	13	12	11	10	9	8
	Byte 6	55	54	53	52	51	50	49	48
High Address	Byte 7	63	62	61	60	59	58	57	56

Little-endian signal with 2x16 bit array type at start position 16 in DID

		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Low Address	Byte 0	7	6	5	4	3	2	1	0
	Byte 1	15	14	13	12	11	10	9	8
Element 1		23	22	21	20	19	18	17	16
High Byte	Byte 2	15	14	13	12	11	10	9	8
Element 1		31	30	29	28	27	26	25	24
Low Byte	Byte 3	7	6	5	4	3	2	1	0
Element 2		39	38	37	36	35	34	33	32
High Byte	Byte 4	15	14	13	12	11	10	9	8
Element 2		47	46	45	44	43	42	41	40
Low Byte	Byte 5	7	6	5	4	3	2	1	0
	Byte 6	55	54	53	52	51	50	49	48
High Address	Byte 7	63	62	61	60	59	58	57	56

Big-endian signal with 2x16 bit array type at start position 40 in DID

Figure 4.4. Byte-aligned signals

The most significant bits (MSBs) of bit-aligned signals are truncated or filled up if the signal size does not match the signal data type size. The signal fill bits are calculated as:



- ▶ ones (negative sign extension), if signal endianness is set to little-endian or big-endian, signal type is set to signed type and most significant data bit is one
- ▶ zeros (positive sign extension) otherwise

NOTE**Definition of the MSBs of bit-aligned opaque signals**

The byte significance is not known. The bits at the high address are truncated or filled up with zeros.

Bit-aligned signals with data array type always have a signal size equal to or less than the size of the data array type.

NOTE**Pitfall of bit-aligned signals with data array type**

The endianness of the signal defines which element is truncated or filled.

- ▶ The MSBs of the first element of a big-endian signal are truncated or filled up.
- ▶ The MSBs of the last element of a little-endian signal are truncated or filled up.
- ▶ The MSBs of the last element of an opaque signal are truncated or filled up.

[Figure 4.5, “Bit-aligned signals”](#) shows a bit-aligned little-endian and a bit-aligned big-endian signal.

		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Low Address	Byte 0	7	6	5	4	3	2	1	0
	Byte 1	15	14	13	12	11	10	9	8
	Byte 2	23	22	21	20	19	18	17	16
Low Byte	Byte 2	3	2	1	0				
	Byte 3	31	30	29	28	27	26	25	24
High Byte	Byte 3	11	10	9	8	7	6	5	4
Truncation / Fill Bits	Byte 4	39	38	37	36	35	34	33	32
	Byte 5	18	17	16	15	14	13	12	
	Byte 6	47	46	45	44	43	42	41	40
	Byte 7	55	54	53	52	51	50	49	48
High Address	Byte 7	63	62	61	60	59	58	57	56

19 bit little-endian signal with 16 bit data type at start position 20 in DID

		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Low Address	Byte 0	7	6	5	4	3	2	1	0
	Byte 1	15	14	13	12	11	10	9	8
	Byte 2	23	22	21	20	19	18	17	16
Truncation / Fill Bits	Byte 2								
Element 1	Byte 3	31	30	29	28	27	26	25	24
Element 1	Byte 3	11	10	9	8	7	6	5	4
Element 1	Byte 4	39	38	37	36	35	34	33	32
Low Byte	Byte 4	3	2	1	0	15	14	13	12
Element 2	Byte 5	47	46	45	44	43	42	41	40
High Byte	Byte 5	11	10	9	8	7	6	5	4
Element 2	Byte 6	55	54	53	52	51	50	49	48
Low Byte	Byte 6	3	2	1	0				
High Address	Byte 7	63	62	61	60	59	58	57	56

30 bit big-endian signal with 2x 16 bit array type at start position 52 in DID

Figure 4.5. Bit-aligned signals

Data is filled up:

- ▶ when the signal is read and signal size is larger than signal data type size.

In the example on the left side, the signal is little-endian, signal size is 19 bit and signal data type size is 16 bit. Bits 16 to 18 of the signal are filled up. The fill bit depends on the value of bit 15 and if data type is signed or unsigned.



- ▶ when the signal is written and signal size is smaller than signal data type size.

In the example on the right side, the signal is big-endian, signal size is 30 bit and signal data array type size is 32 bit. Bits 14 and 15 of the first data element are filled up. The fill bit depends on the value of bit 13 of the first data element and if data array type is signed or unsigned.

Data is truncated:

- ▶ when the signal is written and the signal size is larger than the signal data type size.

In the example on the left side, the signal is little-endian, signal size is 19 bit and data type size is 16 bit. Bits 16 to 18 of the signal are truncated.

- ▶ when the signal is read and the signal size is smaller than the signal data type size.

In the example on the right side, the signal is big-endian, signal size is 30 bit and signal data array type size is 32 bit. Bits 14 and 15 of the first data element are truncated.

4.4.2.5.12. User-defined functions for unsupported DIDs

The `Dcm` module offers the possibility to call a user-defined function for each unsupported data identifier (DID) for `ReadDataByIdentifier` (0x22), `WriteDataByIdentifier` (0x2E) and `InputOutputControl` (0x2F) UDS services, instead of transmitting a negative response. This behavior is activated via the `DcmDspDidEnableDefaultInterfaces` configuration flag.

The functions are described for the following requests:

- ▶ [UDS service ReadDataByIdentifier \(0x22\)](#)
- ▶ [UDS service WriteDataByIdentifier \(0x2E\)](#)
- ▶ [UDS service InputOutputControlByIdentifier \(0x2F\) with resetToDefault](#)
- ▶ [UDS service InputOutputControlByIdentifier \(0x2F\) with returnControlToEcu](#)
- ▶ [UDS service InputOutputControlByIdentifier \(0x2F\) with freezeCurrentState](#)
- ▶ [UDS service InputOutputControlByIdentifier \(0x2F\) with shortTermAdjustment](#)
- ▶ [Transition to the default session](#)

The permitted return values for each callout function are the following. If the returned value is different from the values specified below, a DET error is reported.

- ▶ `E_OK`: Request was successful.
- ▶ `E_NOT_OK`: Request was not successful.
- ▶ `DCM_E_PENDING`: Request is not yet finished. Further call(s) required to finish.



4.4.2.5.12.1. UDS service ReadDataByIdentifier (0x22)

On reception of the UDS service `ReadDataByIdentifier` (0x22), the `Dcm` module checks for every requested DID if the DID is supported. If the requested DID is not supported and the configuration parameter `DcmDspDidEnableDefaultInterfaces` is set to `TRUE`, the `Dcm` calls the application function `Dcm_DefaultDIDRead()`. A DID is considered as not supported if all of the following conditions apply:

- ▶ The DID is not configured as `DcmDspDid`, or the DID is configured as `DcmDspDid` but it is set as unused, i.e. `DcmDspDidUsed` is set to `FALSE`.
- ▶ The DID is not in the range of a configured `DcmDspDidRange`.
- ▶ The DID is in the range of 0xF400 to 0xF4FF (OBD DID), and the least significant byte is not an "availability PID/OBDMID/InfoType" according to the "availability IDs" concept described in AUTOSAR 4.0.3.

The application function `Dcm_DefaultDIDRead()` shall have the following signature:

```
Std_ReturnType Dcm_DefaultDIDRead ( Dcm_OpStatusType OpStatus, Dcm_DIDIDType DID,
uint8 *DestinationBuffer, uint32 *Length, Dcm_NegativeResponseCodeType *Nrc )
```

where:

- ▶ `Dcm_OpStatusType` `OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this callout.
 - ▶ `DCM_PENDING`: This is a subsequent call of this callout after it returned a `DCM_E_PENDING` error code in a previous call. It allows the processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_CANCEL`: Final call for the cancellation of the operation. The `Dcm` ignores the return value for this call.
- ▶ `Dcm_DIDIDType` `DID` is the data identifier.
- ▶ `uint8 *DestinationBuffer` is a pointer to the location in the output buffer where DID data is written.
- ▶ `uint32 *Length` is a pointer at which the application shall write the amount of data, in bytes, that was written to the `DestinationBuffer` pointer.
- ▶ `Dcm_NegativeResponseCodeType` `*Nrc` is a pointer at which the application may write an NRC to be transmitted if the operation is not accepted.

**NOTE****ReadDataByIdentifier****► DID request in range of DcmDspDidRange**

If the requested DID is not configured but is in the range defined by `DcmDspDidRange`, the function `Dcm_DefaultDIDRead()` will not be called. The handling of the DID is performed by the predefined function `DcmDspDidRangeReadDidFnc()`.

► Handling overflows

The application does not receive any information on the size of the buffer that is provided. It must ensure that no overflow occurs.

4.4.2.5.12.2. UDS service WriteDataByIdentifier (0x2E)

On reception of the UDS service `WriteDataByIdentifier` (0x2E), the `Dcm` module checks for every requested (DID) if the DID is supported. If the requested DID is not supported, and the configuration parameter `DcmDspDidEnableDefaultInterfaces` is set to `TRUE`, the `Dcm` calls the application function `Dcm_DefaultDIDWrite()`. A DID is considered as not supported if all of the following conditions apply:

- The DID is not configured as `DcmDspDid`, or the DID is configured as `DcmDspDid` but it is set as unused, i.e. `DcmDspDidUsed` is set to `FALSE`).
- The DID is not in the range of a configured `DcmDspDidRange`.

The application function `Dcm_DefaultDIDWrite()` shall have the following signature:

```
Std_ReturnType Dcm_DefaultDIDWrite ( Dcm_OpStatusType OpStatus, Dcm_DIDIDType DID,
uint8 *SourceBuffer, uint32 Length, Dcm_NegativeResponseCodeType *Nrc )
```

where:

- `Dcm_OpStatusType` `OpStatus` represents the type of the current call to the callout function:
 - `DCM_INITIAL`: This is the initial call of this callout.
 - `DCM_PENDING`: This is a subsequent call of this callout after it returned a `DCM_E_PENDING` error code in a previous call. It allows the processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - `DCM_CANCEL`: Final call for the cancellation of the operation. The `Dcm` ignores the return value for this call.
- `Dcm_DIDIDType` `DID` is the data identifier.
- `uint8 *SourceBuffer` is a pointer to the location in the input buffer where DID data is read from.
- `uint32 Length` is the valid length, in bytes, of the `SourceBuffer`. It represents the length, in bytes, of the received `dataRecord` from the `WriteDataByIdentifier` (0x2E) service request.
- `Dcm_NegativeResponseCodeType *Nrc` is a pointer at which the application may write an NRC to be transmitted if the operation is not accepted.



4.4.2.5.12.3. UDS service InputOutputControlByIdentifier (0x2F) with resetToDefault

On reception of the UDS service `InputOutputControlByIdentifier` (0x2F) with `InputOutputControlParameter` equal to `resetToDefault` for every requested DID, the `Dcm` module checks if the DID is supported. If the requested DID is not supported and the configuration parameter `DcmDspDidEnableDefaultInterfaces` is set to `TRUE`, the `Dcm` calls the application function `Dcm_DefaultDIDResetToDefault()` with the corresponding DID from the request. A DID is considered as not supported if all of the following conditions apply:

- ▶ The DID is not configured as `DcmDspDid`, or the DID is configured as `DcmDspDid` but it is set as unused, i.e. `DcmDspDidUsed` is set to `FALSE`).
- ▶ The DID is not in the range of a configured `DcmDspDidRange`.

The application function `Dcm_DefaultDIDResetToDefault()` shall have the following signature:

```
Std_ReturnType Dcm_DefaultDIDResetToDefault ( Dcm_OpStatusType OpStatus, Dcm_DIDIDType DID, Dcm_NegativeResponseCodeType *Nrc )
```

where:

- ▶ `Dcm_OpStatusType OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this callout.
 - ▶ `DCM_PENDING`: This is a subsequent call of this callout after it returned a `DCM_E_PENDING` error code in a previous call. This allows the processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_CANCEL`: Final call for cancellation of operation. The `Dcm` ignores the return value for this call.
- ▶ `Dcm_DIDIDType DID` is the data identifier.
- ▶ `Dcm_NegativeResponseCodeType *Nrc` is a pointer at which the application may write an NRC to be transmitted if the operation is not accepted.

4.4.2.5.12.4. UDS service InputOutputControlByIdentifier (0x2F) with returnControlToEcu

On reception of the UDS service `InputOutputControlByIdentifier` (0x2F) with `InputOutputControlParameter` equal to `returnControlToEcu` for every requested DID, the `Dcm` module checks if the DID is supported. If the requested DID is not supported and the configuration parameter `DcmDspDidEnableDefaultInterfaces` is set to `TRUE`, the `Dcm` calls the application function `Dcm_DefaultDIDReturnControlToECU()` with the corresponding DID from the request. A DID is considered as not supported if all of the following conditions apply:

- ▶ The DID is not configured as `DcmDspDid`, or the DID is configured as `DcmDspDid` but it is set as unused, i.e. `DcmDspDidUsed` is set to `FALSE`.
- ▶ The DID is not in the range of a configured `DcmDspDidRange`.



The application function `Dcm_DefaultDIDReturnControlToECU()` shall have the following signature:

```
Std_ReturnType Dcm_DefaultDIDReturnControlToECU ( Dcm_OpStatusType OpStatus,
Dcm_DIDIDType DID, Dcm_NegativeResponseCodeType *Nrc )
```

where:

- ▶ `Dcm_OpStatusType OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this callout.
 - ▶ `DCM_PENDING`: This is a subsequent call of this callout after it returned a `DCM_E_PENDING` error code in a previous call. This allows the processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_CANCEL`: Final call for the cancellation of the operation. The `Dcm` ignores the return value for this call.
- ▶ `Dcm_DIDIDType DID` is the data identifier.
- ▶ `Dcm_NegativeResponseCodeType *Nrc` is a pointer at which the application may write an NRC to be transmitted if the operation is not accepted.

4.4.2.5.12.5. UDS service InputOutputControlByIdentifier (0x2F) with freezeCurrentState

On reception of the UDS service `InputOutputControlByIdentifier` (0x2F) with `InputOutputControlParameter` equal to `freezeCurrentState` for every requested (DID), the `Dcm` module checks if the DID is supported. If the requested DID is not supported and the configuration parameter `DcmDspDidEnableDefaultInterfaces` is set to TRUE, the `Dcm` calls the application function `Dcm_DefaultDIDFreezeCurrentState()` with the corresponding DID from the request. A DID is considered as not supported if all of the following conditions apply:

- ▶ The DID is not configured as `DcmDspDid` or the DID is configured as `DcmDspDid` but it is set as unused, i.e. `DcmDspDidUsed` set to FALSE.
- ▶ The DID is not in the range of a configured `DcmDspDidRange`.

The function shall have the following signature:

```
Std_ReturnType Dcm_DefaultDIDFreezeCurrentState ( Dcm_OpStatusType OpStatus,
Dcm_DIDIDType DID, Dcm_NegativeResponseCodeType *Nrc )
```

where:

- ▶ `Dcm_OpStatusType OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this callout.
 - ▶ `DCM_PENDING`: This is a subsequent call of this callout after it returned a `DCM_E_PENDING` error code in a previous call. This allows the processing to be broken up over more than one `Dcm_MainFunction()` cycle.



- ▶ `DCM_CANCEL`: Final call for the cancellation of the operation. The `Dcm` ignores the return value for this call.
- ▶ `Dcm_DIDIDType DID` is the data identifier.
- ▶ `Dcm_NegativeResponseCodeType *Nrc` is a pointer at which the application may write an NRC to be transmitted if the operation is not accepted.

4.4.2.5.12.6. UDS service `InputOutputControlByIdentifier (0x2F)` with `shortTermAdjustment`

On reception of the UDS service `InputOutputControlByIdentifier (0x2F)` with `InputOutputControlParameter` equal to `shortTermAdjustment` for every requested DID, the `Dcm` module checks if the DID is supported. If the requested DID is not supported and the configuration parameter `DcmDspDidEnableDefaultInterfaces` is set to `TRUE`, the `Dcm` calls the application function `Dcm_DefaultDIDShortTermAdjustment()` with the corresponding DID from the request. A DID is considered as not supported if all of the following conditions apply:

- ▶ The DID is not configured as `DcmDspDid`, or the DID is configured as `DcmDspDid` but it is set as unused, i.e. `DcmDspDidUsed` is set to `FALSE`.
- ▶ The DID is not in the range of a configured `DcmDspDidRange`.

The application function `Dcm_DefaultDIDShortTermAdjustment()` shall have the following signature:

```
Std_ReturnType Dcm_DefaultDIDShortTermAdjustment ( Dcm_OpStatusType OpStatus,
Dcm_DIDIDType DID, uint8 *controlState, uint32 Length, Dcm_NegativeResponseCodeType *Nrc )
```

where:

- ▶ `Dcm_OpStatusType OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this callout.
 - ▶ `DCM_PENDING`: This is a subsequent call of this callout after it returned a `DCM_E_PENDING` error code in a previous call. This allows the processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_CANCEL`: Final call for the cancellation of the operation. The `Dcm` ignores the return value for this call.
- ▶ `Dcm_DIDIDType DID` is the data identifier.
- ▶ `uint8 *controlState` is a pointer to the location in the input buffer where the `controlState` (containing the `controlOptionRecord` and `controlMask`) is located.
- ▶ `uint32 Length` is the valid length, in bytes, of the `controlState`. It represents the length, in bytes, of the received `controlOptionRecord` and `controlMask` from the `InputOutputControlByIdentifier`.



- ▶ `Dcm_NegativeResponseCodeType *Nrc` is a pointer at which the application may write an NRC to be transmitted if the operation is not accepted.

4.4.2.5.12.7. Transition to the default session

On a session transition to the default session, either from default session or from non-default session, the `Dcm` stops all the control in progress. This means that while `DcmDspReturnControlToEcuTimeout` (counted from the moment the `DefaultSession` was entered) has not elapsed yet, the `Dcm` does the following:

- ▶ The `Dcm` calls the `ReturnControlToECU` operation for signals of all configured and controllable DIDs (i.e. DIDs that reference a `DcmDspDidInfo` that has any operations within the `DcmDspDidControl` container enabled).
- ▶ If the parameter `DcmDspDidEnableDefaultInterfaces` is set to `TRUE`, the `Dcm` invokes the `Dcm_DefaultDIDReturnControlToECUForAll()` application function.

The `Dcm_DefaultDIDReturnControlToECUForAll()` application function shall have the following signature:

```
Std_ReturnType Dcm_DefaultDIDReturnControlToECUForAll ( Dcm_OpStatusType OpStatus )
```

where:

- ▶ `Dcm_OpStatusType OpStatus` represents the type of the current call to the callout function:
 - ▶ `DCM_INITIAL`: This is the initial call of this callout.
 - ▶ `DCM_PENDING`: This is a subsequent call of this callout after it returned a `DCM_E_PENDING` error code in a previous call. This allows the processing to be broken up over more than one `Dcm_MainFunction()` cycle.
 - ▶ `DCM_CANCEL`: Final call for the cancellation of the operation. The `Dcm` ignores the return value for this call.

4.4.2.5.13. OBD service Request Vehicle Information (0x09)

The purpose of the OBD service Request Vehicle Information (0x09) is to enable the external test equipment to request vehicle-specific information such as the Vehicle Identification Number (VIN) and Calibration IDs. Some of this information may be required by regulations, and some should be reported in a standard format if supported by the vehicle manufacturer.

The `Dcm` module offers an internal implementation of the OBD service Request Vehicle Information (0x09). This service is implemented according to the requirements of the following standards:

- ▶ AUTOSAR 4.0.3
- ▶ AUTOSAR 4.4.0



- ▶ ISO15031-5:2011
- ▶ J1979-DA:2014

For more information on the Request Vehicle Information (0x09) OBD service, see the following specifications:

- ▶ Specification of Diagnostic Communication Manager, AUTOSAR 4.0.3, section 7.4.4.4 *OBD Service \$09 - Request Vehicle Information*
- ▶ Specification of Diagnostic Communication Manager, AUTOSAR 4.4.0, section 7.5.3.9 *Service \$09 - Request Vehicle Information*
- ▶ ISO15031-5:2011, section *Service 0x09 - Request vehicle information*

For more information on the supported INFOTYPES that can be used in the request message for the OBD Service \$09, see the following specifications:

- ▶ J1979-DA:2014, section *TABLE A1 - SUPPORTED PID/OBDMID/TID/INFOTYPE DEFINITION*

Additional features:

- ▶ The Dcm offers the possibility to send a Service Not Supported NRC (0x11) for not supported OBD services. ISO15031-5:2011, section *Table 11 - Proper response from server/ECU for ISO 15765-4 protocol*, specifies that, if OBD service Request Vehicle Information (0x09) is requested and the service is not supported, the Dcm shall issue no response. This feature enables sending the Service Not Supported NRC (0x11) by configuring the parameter `DcmDs1ObdNRCResponseSupression`.

For configuration advice, see [Section 4.4.3.4.10, “Configuring the Request Vehicle Information \(0x09\) OBD service”](#).

4.4.2.6. Diagnostic service request handling

The tester sends diagnostic service requests to the Dcm module. The Dcm processes the service requests and sends a response back to the tester.

The Dcm communicates with other basic software modules or software components. This allows the Dcm to connect to the bus on which the service request is sent and to gather the diagnostic information. The following list explains a typical data flow for processing diagnostic service requests:

1. The diagnostic service request is sent from the tester to the communication stack.
2. The PduR module routes the diagnostic service request from the transport layer module of the communication stack, e.g. CanTp, to the Dcm.

Global PDUs in the EcuC module are used to resolve the routing path between Dcm and PduR.

3. The ComM module specifies if the Dcm can receive data from the communication stack and send responses.



4. The `Dcm` module gathers diagnostic information, e.g. from the `Dem` module or software components.
5. The `PduR` module routes the diagnostic information from the `Dcm` to the communication stack.
6. The communication stack sends the diagnostic information back to the tester.

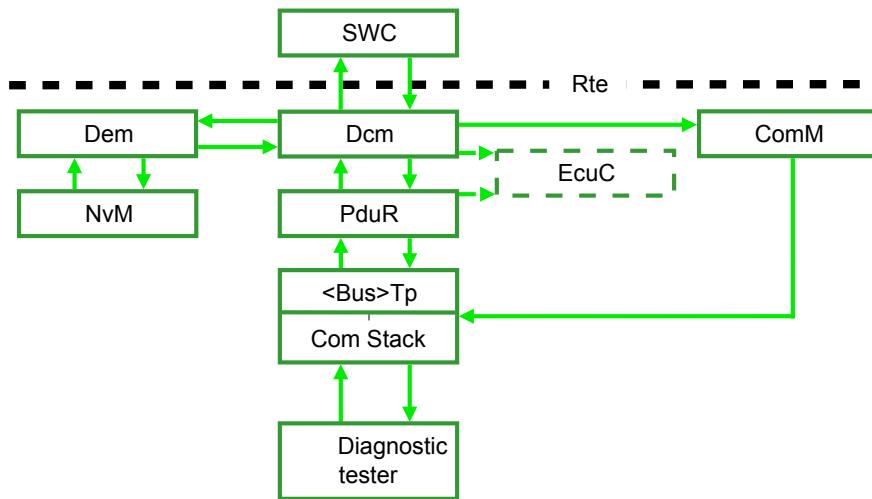


Figure 4.6. Diagnostic service request handling

Symbol	Description
Solid arrow	Communication path
Dashed arrow	Dependency

For information on how to configure the diagnostic service request handling, see [Section 4.4.3.5, “Configuring the diagnostic service request handling”](#).

4.4.2.6.1. Request processing

To understand how the `Dcm` works, you must clarify the following notions:

- ▶ The `Dcm` receives requests on one or several reception channels. These reception channels are called RxPduIDs.
- ▶ The `Dcm` processes requests in the context of a diagnostic protocol.
- ▶ The `Dcm` transmits the response that results from processing the request on a transmission channel. This transmission channel is called TxPduID.

An RxPduID always belongs to a main connection and a main connection always belongs to a diagnostic protocol. A request is processed by the `Dcm` if the following requirements are met:

- ▶ The requested diagnostic service is allowed in the current diagnostic session.



- ▶ The requested diagnostic service is allowed at the current security level.
- ▶ If a mode rule is used: the configured mode rule for the diagnostic service is fulfilled.
- ▶ The requested diagnostic service is not a concurrent `TesterPresent (0x3E 0x80)` request, i.e. a `TesterPresent (0x3E)` request with the `SuppressPositiveResponse (0x80)` bit set that arrives on a functional RxPduld. Such requests are processed in the lower layers and do not get dispatched to their intended protocol.

If subfunctions are used, the following requirements are checked after the diagnostic service requirements:

- ▶ The requested subfunction is supported by the diagnostic service.
- ▶ The requested subfunction is allowed at the current security level.
- ▶ If a mode rule is used: the configured mode rule for the subfunction is fulfilled.

If these requirements are met, the request is processed on the protocol to which the RxPduld's main connection belongs. After the `Dcm` processed the request, the resulting response is transmitted on the TxPduld that belongs to the RxPduld's main connection.

The `Dcm` can also transmit data that was not requested by a diagnostic service request. The following data transmissions are not requested:

- ▶ PeriodicDID readout that is configured via the `ReadDataByPeriodicIdentifier (0x2A)` diagnostic service.
- ▶ Event-triggered response transmission that is configured via the `ResponseOnEvent (0x86)` diagnostic service.

For PeriodicDID readout and event-triggered response transmission, special configuration parameters are used to control which diagnostic protocol is used to process the response and, if required, on which TxPduld the PeriodicDID or event-triggered response is transmitted.

4.4.2.6.2. Dcm buffers

All requests and responses that the `Dcm` processes are stored in buffers. The `Dcm` uses three types of buffers:

- ▶ General-purpose buffers
 - Receive regular requests and store the computed responses.
- ▶ Default reception buffers
 - Special two-byte buffers that are used for the following purposes:
 - ▶ Receive potential concurrent `TesterPresent (0x3E 0x80)` requests.
 - ▶ Receive the first data byte of a request that is rejected with a `BusyRepeatRequest (0x21)` NRC.
- ▶ NRC buffers



Special three-byte buffers that are used to store NRCs.

Every configured TxPduld has its own NRC buffer and every RxPduld has an own default buffer. Out of these buffers, you can only configure general-purpose buffers. Every configured RxPduld uses a general-purpose buffer. The RxPdulds reference the general-purpose buffer to be used for reception. The diagnostic protocol references a general-purpose buffer to store data that needs to be transmitted. You can use general-purpose buffers simultaneously for both reception and transmission. The general-purpose buffers are shared resources that can be used by multiple RxPdulds and multiple protocols to store data. When a request is received, the allocated general-purpose buffer of the RxPduld is reserved for the reception if it is not already used by another process. If the buffer is already used, then the reception is either:

- ▶ Ignored if the `DcmDs1DiagRespOnSecondDeclinedRequest` parameter is set to FALSE.
- ▶ Replied to with a `BusyRepeatRequest` (0x21) NRC if the `DcmDs1DiagRespOnSecondDeclinedRequest` parameter is set to TRUE.

If a reception is successful, a general-purpose buffer needs to be reserved to process the response. If the buffer is already used by another process, the reception is either:

- ▶ Ignored if the `DcmDs1DiagRespOnSecondDeclinedRequest` parameter is set to FALSE.
- ▶ Replied to with a `BusyRepeatRequest` (0x21) NRC if the `DcmDs1DiagRespOnSecondDeclinedRequest` parameter is set to TRUE.

This behavior does not occur if the buffer is used for a reception to which the response needs to be given. This applies if the same buffer is used for both the request reception and the transmission of the response.

4.4.2.6.3. Communication with other basic software modules

The `Dcm` communicates with the communication stack by protocol data units (PDUs). PDUs are routed by the PDU Router module, the `PduR`. To enable a basic software module to receive and transmit PDUs, you must link the `PduR` with the basic software modules with which it communicates.

The `Dcm` does not require information about which communication bus is providing the data. The `Dcm` just receives data on its reception channels which are specified by numeric RxPdulds. The `Dcm` then transmits data on transmission channels that are specified by numeric TxPdulds or numeric TxPduld/ConfirmationTxPduld pairs.

Data reception

The AUTOSAR communication stack specifies for modules conformant to the PDU communication stack that for reception of data, a basic software module must use an RxPduld. Therefore, the `Dcm` receives data on RxPdulds. The RxPdulds can be configured to be interpreted as physical or functional RxPduld. Physical RxPdulds represent unicast RxPdulds. Functional RxPdulds represent multicast RxPdulds.

A two-byte default buffer is provided internally for each RxPduld. The reason for this is that if requests need to be rejected with a `BusyRepeatRequest` (0x21) NRC, at least the first byte of the request needs to be



received by the lower layer. If the request is rejected because the general-purpose buffer is busy, this buffer cannot be used to store the first byte of information. Requests that are rejected based on priority arbitration are stored in the default buffer. If the RxPduld is of a functional type and the incoming request is two bytes long, the request is a potential concurrent `TesterPresent` (0x3E 0x80) request which must always be accepted. In this case, this request is stored in the default buffer. An RxPduld always has its own default buffer.

The numeric RxPduld IDs are used by the lower-layer basic software modules as handle IDs to identify the communication channel for reception sequences. The reception sequences are completed by calls to the following reception APIs:

- ▶ `Dcm_StartOfReception()`
- ▶ `Dcm_CopyRxData()`
- ▶ `Dcm_TpRxIndication()`

The `Dcm` defines the RxPduld since the handle IDs are defined by the callee, i.e. the module that is called by the API. The calling module obtains these RxPduld IDs when the ECU software is generated.

Data transmission

The AUTOSAR communication stack specifies that for data transmission, a basic software module must use a pair of TxPduld and ConfirmationTxPduld. This pair is required because data transmission is completed in two phases:

- ▶ The transmitting module initiates the transmission.
- ▶ The lower layer module copies the data and ends the transmission.

The Handle IDs that are used in API calls are defined by the callee. Therefore, to initiate a transmission, the `PduR` is the callee. This means that the TxPduld is defined by the `PduR`. To copy data and end the transmission, the callee is the `Dcm`. Therefore, the ConfirmationTxPduld is defined by the `Dcm`. The `Dcm` defines a pair of TxPduld and ConfirmationTxPduld per main connection. Out of these, the `Dcm` defines the value of the ConfirmationTxPduld.

A protocol can only handle one request at a time. Therefore, regardless of how many main connections a protocol has, a single general-purpose transmission buffer is needed to store the response. The transmission buffer must be large enough to contain all data that is expected from a diagnostic service response.

Each TxPduld has an internal NRC buffer that is used to store NRCs. The reasons for this are the following:

- ▶ The NRC buffer enables the TxPduld to send `busyRepeatRequest` (0x21) NRCs when the diagnostic protocol is busy with other processes.
- ▶ If the processing of the response takes more than the configured P2 timer or P2* timer for the current diagnostic session, a `RequestCorrectlyReceivedResponsePending` (0x7F service ID 0x78) needs to be transmitted to the tester. If this NRC used the general-purpose buffer, it would corrupt data that is already present in the general-purpose buffer.



The `Dcm` uses the numeric TxPduIDs to identify the communication channel for initiating transmission sequences. The transmission sequences are completed by calling the `PduR_DcmTransmit()` transmission API.

Each one of the following elements represents a transmission channel:

- ▶ `DcmDslProtocolTx` (one per main connection)
- ▶ `DcmDslPeriodicConnection`
- ▶ `DcmDslResponseOnEvent`

The handle IDs are defined by the callee, i.e. the module that is called by the API. In this case, the `PduR` is the callee. The `Dcm` obtains the numeric TxPduIDs that are used to initialize the transmission sequences when you generate the ECU software.

The transmission sequences are continued by the lower layer module by calling the following transmission APIs:

- ▶ `Dcm_CopyTxData()`
- ▶ `Dcm_TpTxConfirmation()`

These APIs are called by using a corresponding handle ID. The handle ID is the `ConfirmationTxPduld` and is defined by the called module, the `Dcm`.

Every instance of the previous transmission-related elements contains another numeric parameter:

- ▶ `DcmDslTxConfirmationPduId` for the `DcmDslProtocolTx`
- ▶ `DcmDslPeriodicTxConfirmationPduId` for the `DcmDslPeriodicConnection`
- ▶ `DcmDslRoeTxConfirmationPduId` for the `DcmDslResponseOnEvent`

This numeric parameter is known as the `ConfirmationTxPduld`. It corresponds to the TxPduID that identifies the transmission channel. The calling module obtains these `ConfirmationTxPduld`s when you generate the ECU software. TxPduIDs and `ConfirmationTxPduld` are always present in 1:1 pairs. Every member of these pairs is unique within a module, i.e. one TxPduID cannot be a member of two pairs as well as one `ConfirmationTxPduld` cannot be a member of more than one such pair.

For detailed information concerning `PduR`, see the EB tresos AutoCore Generic COM Services documentation.

4.4.2.6.4. Application diagnostic requests

The `Dcm` offers the option for the application to directly issue diagnostic requests.

The request from the application is treated like a request received from the lower layers. Internally, the request is enqueued until it can be sent as a result of reception connection availability, buffer availability, and priority arbitration.

To enable this feature, the `DcmDslApplicationDiagnosticRequests` container must be enabled.



When enabled, diagnostic requests coming from the application are supported. The application may generate a diagnostic service request on one of the configured reception PDU IDs (`DcmDslProtocolRxPduId`) on the `Dcm` using the provided API `Dcm.InjectDiagnosticRequest()`.

The parameter `DcmDslApplicationDiagnosticRequestBufferSize` specifies the buffer size reserved for the application request.

For a description of the application interface for this feature, see [Section 4.4.3.5.5, “Enabling diagnostic requests from the application”](#).

4.4.2.6.5. EcuC configuration

The PDUs that enable the communication between two basic software components are configured in the `EcuC` module. The PDUs are used to resolve which handle IDs are used when the communication stack API of a basic software module is called. Handle IDs are used to connect a transmission channel to the intended reception channel. The module that receives an API call defines the handle IDs that are used in this API call.

The `Dcm` defines the handle ID that is required to connect the destination path of the `PduR` to the reception channel of the `Dcm`. Therefore, the `PduR`, which conveys the request, by default does not have information to which reception channel of the `Dcm` it should convey the request. To establish this connection, global PDUs are configured in the `EcuC` module. The destination path of the `PduR` and the reception channel of the `Dcm` are then referenced to the same PDU. This allows the `PduR` to obtain the handle ID of the required reception channel of the `Dcm`. This handle ID is used to call the reception APIs of the `Dcm` to fulfill a reception sequence on this path. Global PDUs are used the same way when the `Dcm` transmits a response to the `PduR`.

For detailed information concerning `EcuC`, see the EB tresos AutoCore Generic Base documentation.

4.4.2.6.6. Communication modes

Every RxPduld has a ComM channel assigned to it. The `ComM` module controls if the RxPduld can receive diagnostic service requests and transmit responses.

If the ComM channel is in `NoCommunicationMode`:

- ▶ Requests cannot be received on the RxPduld.
- ▶ Responses to a request that was received on the RxPduld cannot be sent. This also applies to event-triggered responses and periodic transmission.

If the ComM channel is in `SilentCommunicationMode`:

- ▶ Requests can be received on the RxPduld.
- ▶ Responses to a request that was received on the RxPduld cannot be sent. This also applies to event-triggered responses and periodic transmission.



If the ComM channel is in `FullCommunicationMode`:

- ▶ Requests can be received on the RxPduld.
- ▶ Responses to a request can be sent on the RxPduld.

The `DcmDslEnableSetActiveDiagnosis` parameter controls if the `Dcm` can request the `ComM` to go to the `FullCommunicationMode` or maintain it. This behavior can be configured for the RxPduld's associated ComM channel when it receives a request in an internally-managed diagnostic session.

For detailed information concerning `ComM`, see the EB tresos AutoCore Generic Mode Management documentation.

4.4.2.6.7. Page buffering

Page buffering is used when a diagnostic service response is too large for the transmission buffer of the protocol. Page buffering is supported for selected subfunctions of the diagnostic service `ReadDTCInformation` (0x19). The `Dcm` gradually retrieves information from the `Dem` module until the buffer is filled and transmits the data. After all data is transmitted, the rest of the data is retrieved in the same way.

4.4.2.7. BSW distribution

For the general concept of BSW distribution, see chapter *BSW distribution support* in the EB tresos AutoCore Generic documentation.

You can enable or disable the BSW distribution support for `Dcm`.

When BSW distribution support is disabled, the `Dcm` always uses direct function calls because `Dcm` and `ComM` are mapped to the same partition (core).

When BSW distribution support is enabled, the `Dcm` always uses `SchM_Call()` operations. You must ensure that BSW distribution support is also enabled in the `ComM` module. Depending on the project use case, `Dcm` and `ComM` can be mapped to the same partition (core) or to different partitions (cores). In order to support both cases, the calls between `Dcm` and `ComM` are handled by using `SchM_Call()` operations.

The `Dcm` BSW distribution support is enabled with the configuration parameter [`DcmComMMultiCoreSupport`](#). If this parameter is enabled, the generated basic software module description includes all required and provided client-server entries. These entries are generated based on the number of configured channels. You must also enable the BSW distribution support for the `ComM` module that is used in your project. This allows you to connect the required client-server connections in the `Rte`.

The calling context of the `SchM_Call()` operations can be the context of an interrupt. Ensure that the `Rte/BSW` event `IntTriggerOccurredEvent_HighestPrioFunction` of the `Rte` configuration is mapped to the task with the highest priority.



For information on how to configure the `Dcm` BSW distribution, see [Section 4.4.3.6, “Configuring BSW distribution”](#).

4.4.3. Configuring the Dcm module

This chapter provides you with information on how to configure specific components of the `Dcm` module. Depending on the status of your project, you may not need to configure all components that are described in this chapter.

To understand how to configure the `Dcm` module, you must be familiar with the basic concepts of the `Dcm`. For detailed information on the `Dcm` concepts, see [Section 4.4.2, “Background information”](#).

If you do not have much experience in the Diagnostic Stack or in EB tresos Studio, see [Chapter 6, “ACG8 Diagnostic Stack tutorial”](#). This tutorial provides a short overview of the Diagnostic Stack. The tutorial also provides detailed instructions that allow you to configure a first basic Diagnostic Stack configuration.

4.4.3.1. Overview

This chapter provides information on how to configure the `Dcm` module and consists of the following sections:

- ▶ [Section 4.4.3.2, “Configuring diagnostic sessions”](#) explains how to configure a new diagnostic session.

This chapter is useful if you must configure regular diagnostic sessions that are specified in the ISO 14229 1 standard. This chapter is also useful if you must configure manufacturer-specific or supplier-specific diagnostic sessions that are not specified in this standard. If all diagnostic sessions that you require for your project were already provided, e.g. via system description file, you can skip this chapter.

- ▶ [Section 4.4.3.3, “Configuring security levels”](#) explains how to configure a new security level.

This chapter is useful if you must configure security levels and related behavior such as `AttemptCounters` and `DelayTimers`. If all security levels that you require for your project were already provided, e.g. via system description file, you can skip this chapter.

- ▶ [Section 4.4.3.4, “Configuring diagnostic services”](#) explains how to configure internally-managed or externally-managed diagnostic services and subfunctions.

This chapter is useful if you must configure internally-managed diagnostic services that are supported in the ACG8 Diagnostic Stack. This chapter is also useful if you must configure externally-managed diagnostic services, i. e. manufacturer-specific or supplier-specific diagnostic services. If all diagnostic services that you require for your project were already provided, e.g. via system description file, you can skip this chapter.

- ▶ [Section 4.4.3.5, “Configuring the diagnostic service request handling”](#) explains how to configure a basic request handling for the `Dcm`. The request handling of the `Dcm` consists of receiving a diagnostic service request from a diagnostic tool, processing the request, and transmitting a response to the diagnostic tool.



To enable the request handling, you configure diagnostic buffers, the communication between the `Dcm` and the `PduR`, the communication mode handling, and the diagnostic protocol.

- ▶ [Section 4.4.3.6, “Configuring BSW distribution”](#) explains how to configure the `Dcm` for BSW distribution.

If BSW distribution support is enabled, the direct function calls between `Dcm` and `ComM` are replaced by `SchM_Call()` operations.

TIP



Diagnostic sessions and diagnostic services in recommended configurations

The ACG8 Diagnostic Stack comes with the recommended configurations `DcmRecConfigurationBasic` and `DcmRecConfigurationDfl` for the `Dcm` module. The following components of the `Dcm` are already configured if you use these recommended configurations:

- ▶ The diagnostic sessions that are specified in the ISO 14229 1:2013 standard.
- ▶ Most of the internally-managed diagnostic services of ACG8 Diagnostic Stack.

You can select these recommended configurations when you add the `Dcm` module to your project.

4.4.3.2. Configuring diagnostic sessions

This chapter explains how to configure a diagnostic session. The configuration consists of the following steps:

- ▶ [Section 4.4.3.2.1, “Configuring diagnostic session handling for different AUTOSAR versions”](#)
- ▶ [Section 4.4.3.2.2, “Creating diagnostic sessions”](#)
- ▶ [Section 4.4.3.2.3, “Modifying protocol timings”](#)
- ▶ [Section 4.4.3.2.4, “Configuring bootloader interaction”](#)

4.4.3.2.1. Configuring diagnostic session handling for different AUTOSAR versions

In this section, you configure mode declaration support to make the `Dcm` compatible with different AUTOSAR versions.

For more information on the differences between the AUTOSAR versions, see [Section 4.4.2.2.2, “Diagnostic session switch in AUTOSAR 4.0.3 and AUTOSAR 4.0.2”](#) and [Section 4.4.2.4.3, “Bootloader interaction in AUTOSAR 4.0.3 and AUTOSAR 4.0.2”](#).

Required modules

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: `BswM` or `SchM/Rte`



Configuring the mode declaration support

Step 1

To configure the `Dcm` according to AUTOSAR 4.0.3 or higher, enable the `DcmModeDeclarationSupport` parameter. The mode handling of the `Dcm` is handled via the `Rte`.

Step 2

To configure the `Dcm` according to AUTOSAR 4.0.2 or lower, clear the `DcmModeDeclarationSupport` parameter. The mode handling of the `Dcm` is handled via the `BswM`.

4.4.3.2.2. Creating diagnostic sessions

In this section, you create a diagnostic session and configure the protocol timers and the `DiagnosticSessionControl` (0x10) diagnostic service. The `DiagnosticSessionControl` service enables the `Dcm` to switch to a diagnostic session. Depending on the AUTOSAR version for your configuration, you must configure either the `Rte` or the `BswM` module to enable switching to another diagnostic session.

For more information on diagnostic sessions, see [Section 4.4.2.2, “Diagnostic session”](#).

Required features:

- ▶ Keep track of active non-default sessions
- ▶ Support of user-defined sessions not specified in ISO (optional)

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: `Rte` or `BswM`

NOTE**Creating non-ISO diagnostic sessions**

Diagnostic sessions that are not specified in ISO 14229 1:2013 are configured and internally handled the same way as diagnostic sessions that are specified in this standard.



Configuring the `Dcm`

Step 1

To add a diagnostic session, add an entry to the **Session Configuration List** table.

Step 2

For the `DcmDspSessionLevel` parameter of the new diagnostic session, enter a number that is not already used for another diagnostic session.

Step 3

To configure the maximum number of RequestCorrectlyReceivedResponsePending (0x7F service ID 0x78) NRCs, configure the `DcmDslDiagRespMaxNumRespPend` parameter.

Step 4

If the maximum P2/P2*Server time-outs should be used in the positive response for the `DiagnosticSessionControl` (0x10) service, enable the `DcmDspSessionControlTiming` parameter.

TIP**Positive response format according to ISO 14229**

ISO 14229-1:2006 does not specify that the maximum P2/P2*Server time-outs are used in the positive response for the `DiagnosticSessionControl` (0x10) service.

ISO 14229-1:2013 specifies that the maximum P2/P2*Server time-outs are used in the positive response for the `DiagnosticSessionControl` (0x10) service.

Step 5

To configure a `DiagnosticSessionControl` (0x10) service handler:

Step 5.1

In the **Diagnostic Services** table, add an entry.

Step 5.2

For the `DcmDsdSidTabServiceId` parameter of the new entry, enter 0x10.

Step 6

To enable the ECU to switch to this diagnostic session:

Step 6.1

In the **SubService Configuration** table, add a new entry.

Step 6.2

In the `DcmDsdSubServiceId` parameter, enter the diagnostic session level that you entered in the `DcmDspSessionLevel` parameter.



Configuring the Rte (AUTOSAR 4.0.3 or higher)

Step 1

To configure the Rte, connect all **Required Mode Groups** to the **Provided Mode Groups** in the Rte Editor.



Configuring the BswM (AUTOSAR 4.0.2 or lower)

In this section, you create a BswM action that calls the `Dcm_<SessionName>ModeEntry()` API when the Dcm requests a switch to the corresponding diagnostic session. You must configure the BswM according to the following figure:

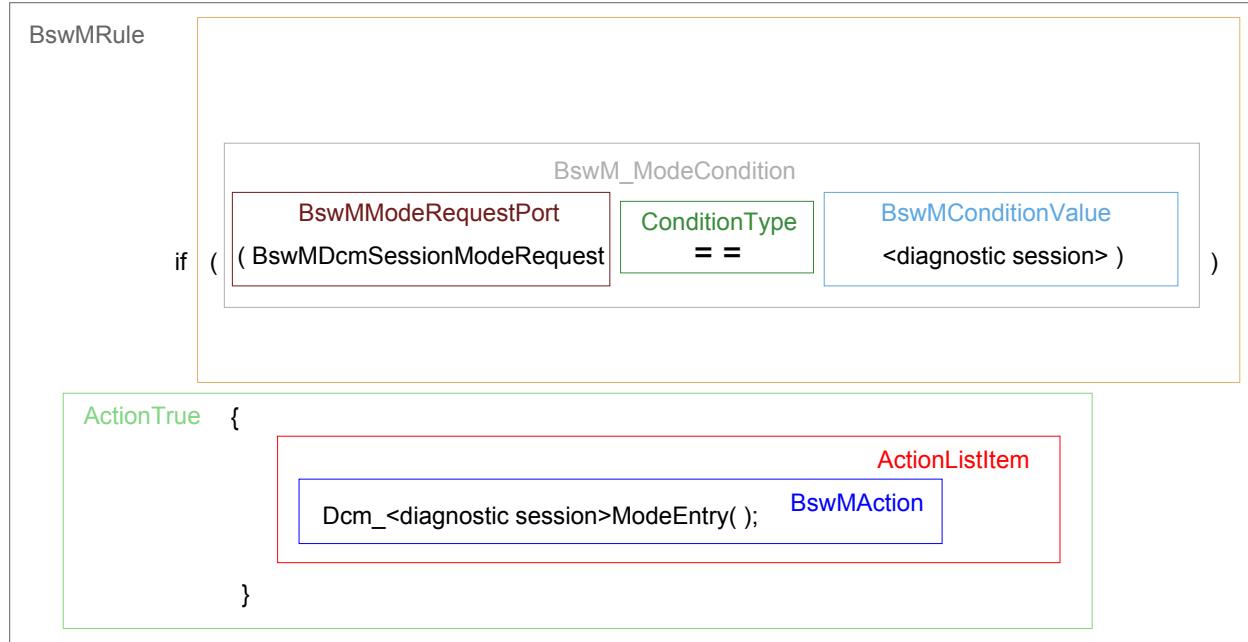


Figure 4.7. BswM configuration for a diagnostic session switch

The following instruction describes in more detail how to configure the BswM.

Step 1

In the **Configuration** table, add an entry.

Step 2

The BswMAction requests the BswM module to switch to the diagnostic session. To configure the BswMAction:

Step 2.1

In the **BswMAction** table, add an entry.

Step 2.2

In the **BswMAvailableActions** parameter, select **BswMUserCallout**.

Step 2.3

In the **BswMUserCalloutFunction** parameter, enter **Dcm_<SessionName>ModeEntry()**.

This function acknowledges to the Dcm module the transition to the new diagnostic session.

Step 2.4

In the **BswMActionList** table, add an entry.

Step 2.5

In the **BswMActionListExecution** parameter, select **BSWM_CONDITION**.

The BswMActions that are contained in the BswMActionList are only executed when specific conditions are met. The conditions are configured in the later steps.



Step 2.6

In the **BswMActionListItem** table, add an entry.

Step 2.7

In the **BswMActionListItemRef** parameter, select your **BswMAction**.

Step 3

The **BswMModeRequestPort** enables the **Dcm** to notify the **BswM** about the switch request. To configure the **BswMModeRequestPort**:

Step 3.1

In the **BswMModeRequestPort** table, add an entry.

Step 3.2

In the **BswMRequestProcessing** parameter, select either **BSWM_IMMEDIATE** or **BSWM_DEFERRED**.

Step 3.3

In the **BswMModeRequestSource** parameter, select **BswMDcmSessionModeRequest**.

Step 4

The **BswMModeCondition** specifies the conditions under which the **BswMAction** is executed. To configure the **BswMModeCondition**:

Step 4.1

In the **BswMModeCondition** table, add an entry.

Step 4.2

In the **BswMConditionType** parameter, select **BSWM_EQUALS**.

Step 4.3

In the **BswMConditionMode** parameter, select your **BswMModeRequestPort**.

Step 4.4

In the **BswMConditionValue** parameter, select **BswMBswMode**.

Step 4.5

In the **BswMBswRequestedMode** parameter, enter the name of your diagnostic session.

Step 5

The **BswMLogicalExpression** allows to evaluate **BswMModeConditions** as part of a logical expression. To configure the **BswMLogicalExpression**:

Step 5.1

In the **BswMLogicalExpression** table, add an entry.

Step 5.2

Clear the **BswMLogicalOperator** parameter.

Step 5.3

In the **BswMArgumentRef** table, add an entry.

Step 5.4

In the **BswMArgument** parameter, select your **BswMModeCondition**.

Step 6

The **BswMRule** combines the **BswMLogicalExpression** and the **BswMAction** to define what action is executed under what conditions. To configure the **BswMRule**:



Step 6.1

In the **BswMRule** table, add an entry.

Step 6.2

In the `BswMRuleInitState` parameter, select `BSWM_FALSE`.

Step 6.3

In the `BswMRuleExpressionRef` parameter, select your `BswMLLogicalExpression`.

Step 6.4

Select the `BswMRuleTrueActionList` parameter.

The configured behavior is executed when the `Dcm` requests a switch to the `BSWM` mode to the desired session, i.e. when the configured `BswMRule` becomes TRUE.

Step 6.5

In the `BswMRuleTrueActionList` parameter, select your `BswMActionList`.

4.4.3.2.3. Modifying protocol timings

In this section, you adjust the P2 timer and the P2* timer of the diagnostic session.

NOTE



Limitations of time measurement

The `Dcm` measures time in terms of the number of calls to the `Dcm_MainFunction()` API multiplied with the `DcmTaskTime` parameter. The `DcmTaskTime` parameter configures the period between two successive calls to the `Dcm_MainFunction()` API. Therefore, the `Dcm` cannot measure time with a higher resolution. Configure the `DcmDspSessionP2ServerMax` and `DcmDspSessionP2StarServerMax` parameters to be a multiple of the `DcmTaskTime` parameter. If project requirements result in a value that is not a multiple of the `DcmTaskTime` parameter, use the first multiple of the `DcmTaskTime` lower than or equal to the value which results from project requirements.

For more information on protocol timers, see [Section 4.4.2.2.1, “Protocol timers”](#).

Required feature: Allow to modify protocol timings

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: none



Configuring the `Dcm`

Step 1

Enter the `P2ServerMax` time in the `DcmDspSessionP2ServerMax` parameter of the diagnostic session.

**Step 2**

Enter the P2StarServerMax time in the `DcmDspSessionP2StarServerMax` parameter of the diagnostic session.

Step 3

To adjust the P2/P2* timers, configure the `DcmTimStrP2ServerAdjust` and `DcmTimStrP2StarServerAdjust` parameters of the diagnostic protocol.

NOTE**Immediate transmission of RequestCorrectlyReceivedResponsePending**

If the protocol timers and adjustment timers are configured according to the following calculation, a `RequestCorrectlyReceivedResponsePending` (0x7F service ID 0x78) NRC is sent immediately:

P2 timer - P2 adjustment timer $\leq 2 * \text{DcmTaskTime}$

4.4.3.2.4. Configuring bootloader interaction

In this section, you configure the `Dcm` to switch to the bootloader when the ECU switches to a defined diagnostic session. Depending on the AUTOSAR version for your configuration, you must configure either the `Rte` or the `BswM` module to enable the bootloader interaction.

For more information on the bootloader interaction, see [Section 4.4.2.4, “Bootloader interaction”](#).

Required feature: Bootloader interaction (jump to/from)

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: `Rte` or `BswM`



Configuring the Dcm

Step 1

In the `DcmDspSessionForBoot` parameter of the diagnostic session, select either `DCM_OEM_BOOT` or `DCM_SYS_BOOT`.

The actual jump logic is implemented in the software component or complex device driver.

Step 2

If you want to send a `RequestCorrectlyReceivedResponsePending` (0x7F 0x10 0x78) NRC to the tester before the reset functionality is executed, enable the `DcmSendRespPendOnTransToBoot` parameter.



Configuring the Rte (AUTOSAR 4.0.3 or higher)

Step 1

Connect all **Required Mode Groups** to the **Provided Mode Groups**.

The ModeDeclarationGroupPrototype DcmEcuReset is switched to the EXECUTE Mode. On this switch, the basic software module or complex device driver that monitors the DcmEcuReset ModeDeclarationGroupPrototype performs the switch to the bootloader.



Configuring the BswM (AUTOSAR 4.0.2 or lower)

To configure the BswM to execute a switch to the bootloader, two BswMActions are required. The first BswMAction calls the function of the application that prepares the switch to the bootloader. The second BswMAction calls the `BswM_Dcm_RequestResetMode()` API that executes the switch to the bootloader. You must configure the BswM according to the following figures:

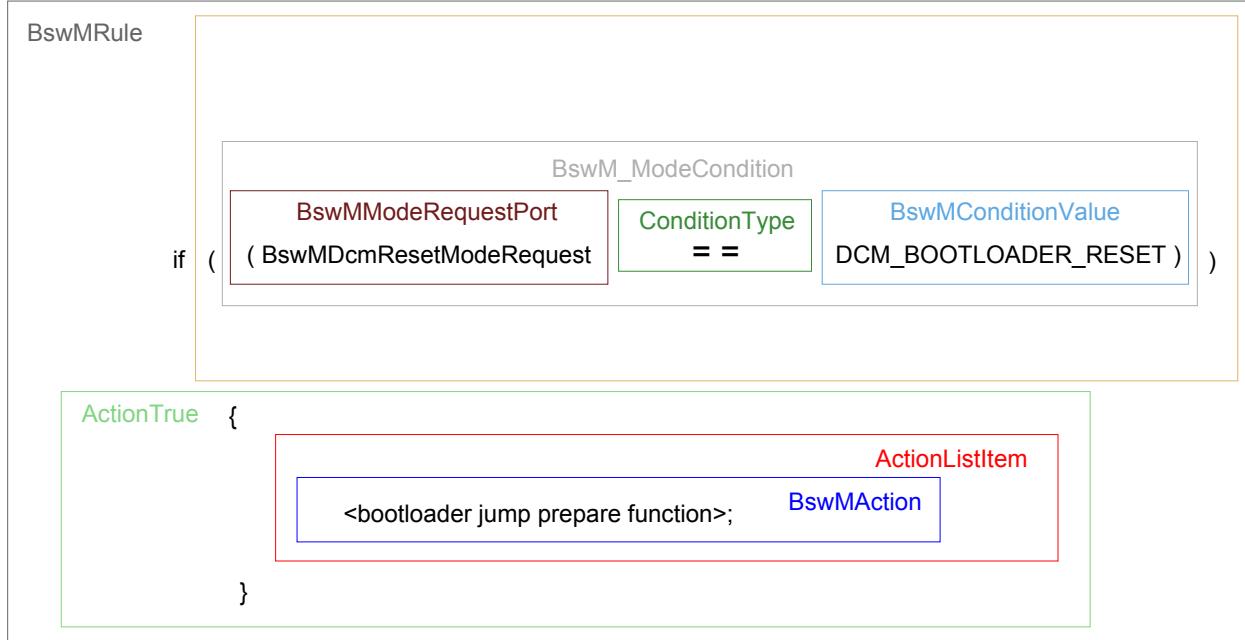


Figure 4.8. BswM configuration to prepare the bootloader jump.

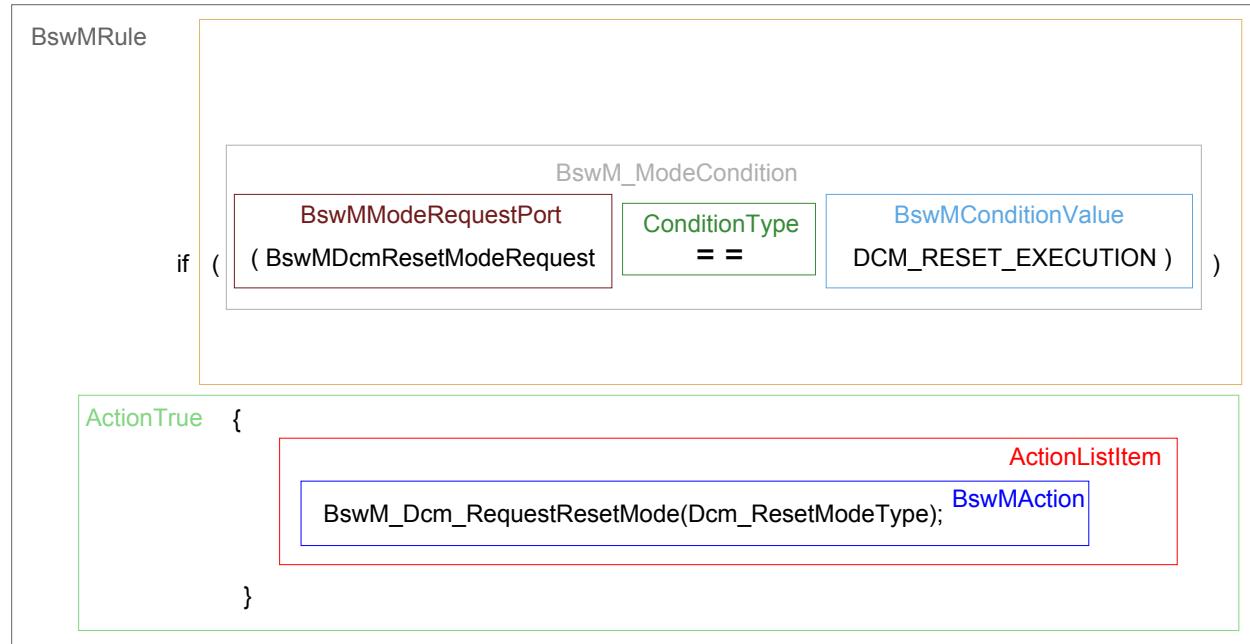


Figure 4.9. BswM configuration to execute the bootloader jump.

The following instruction describes in more detail how to configure the BswM.

Step 1

In the **Configuration** table, add an entry.

Step 2

The BswMAction requests the BswM module to switch to the reset mode. To configure the BswMAction:

Step 2.1

In the **BswMAction** table, add an entry.

Step 2.2

In the `BswMAvailableActions` parameter, select `BswMUserCallout`.

Step 2.3

In the `BswMUserCalloutFunction` parameter, enter the name of the function that prepares the jump.

Step 2.4

In the **BswMActionList** table, add an entry.

Step 2.5

In the `BswMActionListExecution` parameter, select `BSWM_CONDITION`.

Step 2.6

The BswMActions contained in the `BswMActionList` are executed only when the configured conditions for executing them are true. To configure these conditions:

Step 2.7

In the **BswMActionListItem** table, add an entry.



Step 2.8

In the `BswMActionListItemRef` parameter, select your `BswMAction`.

Step 3

The `BswMModeRequestPort` enables the `Dcm` to notify the `BswM` about the switch request. To configure the `BswMModeRequestPort`:

Step 3.1

In the **`BswMModeRequestPort`** table, add an entry.

Step 3.2

In the `BswMRequestProcessing` parameter, select either `BSWM_IMMEDIATE` or `BSWM_DEFERRED`.

Step 3.3

In the `BswMModeRequestSource` parameter, select `BswMDcmSessionModeRequest`.

Step 4

The `BswMModeCondition` specifies the conditions under which the `BswMAction` is executed. To configure the `BswMModeCondition`:

Step 4.1

In the **`BswMModeCondition`** table, add an entry.

Step 4.2

In the `BswMConditionType` parameter, select `BSWM_EQUALS`.

Step 4.3

In the `BswMConditionMode` parameter, select your `BswMModeRequestPort`.

Step 4.4

In the `BswMConditionValue` parameter, select `BswMBswMode`.

Step 4.5

In the `BswMBswRequestedMode` parameter, enter either `DCM_BOOTLOADER_RESET` for a jump to the OEM bootloader or `DCM_SS_BOOTLOADER_RESET` for a jump to the system supplier bootloader.

The required setting depends on the value configured in the `DcmDspSessionForBoot` parameter of the diagnostic session that triggers the jump.

Step 5

The `BswMLogicalExpression` allows to evaluate `BswMModeConditions` as part of a logical expression. To configure the `BswMLogicalExpression`:

Step 5.1

In the **`BswMLogicalExpression`** table, add an entry.

Step 5.2

Clear the `BswMLogicalOperator` parameter.

Step 5.3

In the **`BswMArgumentRef`** table, add an entry.

Step 5.4

In the `BswMArgument` parameter, select your `BswMModeCondition`.



Step 6

The **BswMRule** combines the **BswMLLogicalExpression** and the **BswMAction** to define what action is executed under which conditions. To configure the **BswMRule**:

Step 6.1

In the **BswMRule** table, add an entry.

Step 6.2

In the **BswMRuleInitState** parameter, select **BSWM_FALSE**.

Step 6.3

In the **BswMRuleExpressionRef** parameter, select your **BswMLLogicalExpression**.

Step 6.4

Enable the **BswMRuleTrueActionList** parameter.

The configured behavior is executed when the **Dcm** requests a switch to the **BSWM** mode to the desired session, i.e. when the configured **BswMRule** becomes true.

Step 6.5

In the **BswMRuleTrueActionList** parameter, select your **BswMActionList**.

You created the **BswMAction** that prepares the switch to the bootloader.

Step 7

To configure the second **BswMAction** that executes the switch to the bootloader, perform the same steps that are listed above with the following differences:

- ▶ In the **BswMUserCalloutFunction** parameter, enter `BswM_Dcm_RequestResetMode()`. This function executes the switch to the bootloader.
- ▶ In the **BswMBswRequestedMode** parameter, select `DCM_RESET_EXECUTION`.

You created the **BswMAction** that executes the switch to the bootloader.

4.4.3.3. Configuring security levels

This chapter explains how to configure security levels. The configuration consists of the following steps:

- ▶ [Section 4.4.3.3.1, “Configuring security level interfaces for different AUTOSAR versions”](#)
- ▶ [Section 4.4.3.3.2, “Creating security levels”](#)
- ▶ [Section 4.4.3.3.3, “Configuring DelayTimers and AttemptCounters”](#)

4.4.3.3.1. Configuring security level interfaces for different AUTOSAR versions

In this section, you configure interface signatures for security levels to make the **Dcm** compatible with different AUTOSAR versions.



For more information on the differences between interfaces for security level operations for different AUTOSAR versions, see [Section 4.4.2.3.2, “Application interfaces for security levels”](#).

Required modules

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: `NvM` and `Rte` if you use ClientServerInterfaces



Configuring AUTOSAR interface versions

Step 1

To configure the `Dcm` to use AUTOSAR 4.0.3 interface signatures, configure the `DcmDefaultASRServiceAPI` parameter to `AUTOSAR_40`.

Step 2

To configure the `Dcm` to use AUTOSAR 4.2 or higher interface signatures, configure the `DcmDefaultASRServiceAPI` parameter to `AUTOSAR_42`.

NOTE



Impact of the AUTOSAR interface version

The selection of the default AUTOSAR interface version via the `DcmDefaultASRServiceAPI` parameter affects the signature of all generated `Dcm` interfaces, except for specific generated interfaces for which an override parameter is specified.

4.4.3.3.2. Creating security levels

In this section, you create a security level. The transition to the security levels configured here can be achieved via the `SecurityAccess` (0x27) service.

For more information on security levels, see [Section 4.4.2.3, “Security level”](#).

Required features:

- ▶ Manage security level

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: `Rte` if you use ClientServerInterfaces



Configuring the `Dcm`

Step 1

To add a security level, add an entry to the **Security Levels** table.

**Step 2**

For the `DcmDspSecurityLevel` parameter of the new security level, enter a number that is not already used for another security level.

Step 3

To enable the ECU to switch to this security level, configure the `SecurityAccess (0x27)` UDS service as described in [Section 4.4.3.4.7, “Configuring the `SecurityAccess \(0x27\)` UDS service”](#).

Step 4

To select the interface via which the operations of the security level are implemented, configure the `DcmDspSecurityUsePort` parameter of the new security level.

Step 5

If a `securityAccessDataRecord` is expected in the `RequestSeed` subfunction request for this security level:

- ▶ Enable the `DcmDspSecurityADRSIZE` parameter.
- ▶ Configure the size of the expected `securityAccessDataRecord` as the value of the `DcmDspSecurityADRSIZE` parameter.

Step 6

If the chosen interface (value of the `DcmDspSecurityUsePort` parameter) is `USE_SECURITY_SYNCH_FNC` or `USE_SECURITY_ASYNC_FNC`:

Step 6.1

Enable the `DcmDspSecurityGetSeedFnc` parameter.

Step 6.2

Configure the symbol name of the callout function used for the `GetSeed` operation as the value of the `DcmDspSecurityGetSeedFnc` parameter.

Step 6.3

Enable the `DcmDspSecurityCompareKeyFnc` parameter.

Step 6.4

Configure the symbol name of the callout function used for the `CompareKey` operation as the value of the `DcmDspSecurityCompareKeyFnc` parameter.

If the chosen interface (value of the `DcmDspSecurityUsePort` parameter) is `USE_SECURITY_SYNCH_CLIENT_SERVER` or `USE_SECURITY_ASYNC_CLIENT_SERVER`, no other steps are needed in order to configure the `Dcm`.

**Configuring the Rte****Step 1**

To configure the `Rte`, connect all `SecurityAccess <SecurityLevel>` `RequiredPorts` generated by the `Dcm` for its security levels to the corresponding `ProvidedPorts` provided by the application.



4.4.3.3.3. Configuring DelayTimers and AttemptCounters

In this section, you configure `DelayTimer` and `AttemptCounter` behavior for security levels.

For more information on `AttemptCounters` and `DelayTimers`, see [Section 4.4.2.3.1, “AttemptCounter and DelayTimer”](#).

The configuration consists of the following elements:

- ▶ Global configuration of `AttemptCounter` and `DelayTimer` behavior. This includes:
 - ▶ Configuration of unified `DelayTimer` behavior: [Section 4.4.3.3.1, “Configuring the Dcm”](#)
 - ▶ Configuration of store and load time-outs for `AttemptCounters`: [Section 4.4.3.3.2, “Configuring store and load time-outs for AttemptCounters”](#)
 - ▶ Configuration of infinite `DelayTimers` upon `AttemptCounter` retrieval failure: [Section 4.4.3.3.3, “Configuring infinite DelayTimers upon AttemptCounter retrieval failure”](#)
 - ▶ Configuration of `AttemptCounter` resetting upon `DelayTimer` expiration: [Section 4.4.3.3.4, “Configuring AttemptCounter resetting upon DelayTimer expiration”](#)
 - ▶ Configuration of `AttemptCounter` incrementation limiting: [Section 4.4.3.3.5, “Configuring AttemptCounter incrementation limiting”](#)
 - ▶ Configuration of legacy `AttemptCounter` handling: [Section 4.4.3.3.6, “Configuring legacy AttemptCounter handling”](#)
- ▶ Per security level configuration of `AttemptCounter` and `DelayTimer` behavior. This includes:
 - ▶ Configuration of a security level `DelayTimer`: [Section 4.4.3.3.7, “Configuring a security level DelayTimer”](#)
 - ▶ Configuration of infinite `DelayTimers`: [Section 4.4.3.3.8, “Configuring infinite DelayTimers”](#)
 - ▶ Configuration of `OnBootDelayTimers`: [Section 4.4.3.3.9, “Configuring OnBootDelayTimers”](#)
 - ▶ Configuration of forced `OnBootDelayTimer` starting: [Section 4.4.3.3.10, “Configuring forced On-BootDelayTimers”](#)
 - ▶ Configuration of a security level `AttemptCounter` limit: [Section 4.4.3.3.11, “Configuring a security level's AttemptCounter limit”](#)
 - ▶ Configuration of security level `AttemptCounter` storage to non-volatile memory: [Section 4.4.3.3.12, “Configuring a security level's AttemptCounter to be stored to non-volatile memory”](#)

4.4.3.3.1. Configuring the Dcm

In this section, you configure unified `DelayTimers`, i.e. you apply any started `DelayTimer` to all configured security levels.

Required modules



- ▶ Required diagnostic stack modules: Dcm



Configuring unified DelayTimers

Step 1

To configure the Dcm to use unified DelayTimers, configure the DcmDspSecurityUnifiedDelayTimer parameter to TRUE.

4.4.3.3.3.2. Configuring store and load time-outs for AttemptCounters

In this section, you configure:

- ▶ the maximum amount of time allowed for the Dcm to store the value of an AttemptCounter to non-volatile memory when resetting the AttemptCounter upon DelayTimer expiry
- ▶ the maximum amount of time allowed for the Dcm to retrieve the value of all AttemptCounters from non-volatile memory upon start-up

Required modules

- ▶ Required diagnostic stack modules: Dcm



Configuring the Dcm

Step 1

Configure the maximum amount of time allowed for the Dcm to store the value of an AttemptCounter to non-volatile memory when resetting the AttemptCounter upon DelayTimer expiry as the value of the DcmDspSecurityMaxAttemptCounterWriteTime parameter.

**NOTE****Resetting the AttemptCounter upon expiry of a DelayTimer**

This time-out only applies to storing the value of the AttemptCounter if the AttemptCounter resets upon `DelayTimer` expiry. In all other circumstances (resetting upon successful security level switch and incrementation), the store operation is invoked in the context of a `SecurityAccess (0x27)` service request. In this case, the maximum amount of time available for the store operation to succeed is determined by the following:

- ▶ the P2 and P2* time-outs configured for the current active session, coupled with
- ▶ the number of `RequestCorrectlyReceivedResponsePending (0x7F 0x27 0x78)` NRCs sent, which can be configured as explained in [Section 4.4.3.2.2, “Creating diagnostic sessions”](#).

Step 2

Configure the maximum amount of time allowed for the `Dcm` to retrieve the value of an AttemptCounter from non-volatile memory upon start-up as the value of the `DcmDspSecurityMaxAttemptCounterReadoutTime` parameter.

4.4.3.3.3. Configuring infinite DelayTimers upon AttemptCounter retrieval failure

In this section, you configure security levels to become permanently inaccessible if retrieving the value of their AttemptCounters from non-volatile memory upon start-up fails.

Required modules

- ▶ Required diagnostic stack modules: `Dcm`



Configuring the Dcm

Step 1

To configure security levels to become permanently inaccessible if retrieving the value of their AttemptCounters from non-volatile memory upon start-up fails, configure the `DcmDspSecurityAttemptCounterReadoutFailInfiniteDelay` parameter to `TRUE`.

**NOTE****Permanent inaccessibility applicability**

A security level for which the retrieval of the AttemptCounter from non-volatile memory failed and thus (if configured) becomes permanently inaccessible is actually inaccessible until the next reboot. Upon reboot, the AttemptCounter retrieval is performed again and, if successful, causes the security level to become accessible again.

NOTE**Permanent inaccessibility prioritization**

Permanent inaccessibility upon AttemptCounter retrieval failure overrides any configured forced OnBootDelayTimer (see [Section 4.4.3.3.3.10, “Configuring forced OnBootDelay-Timers”](#)).

4.4.3.3.4. Configuring AttemptCounter resetting upon DelayTimer expiration

In this section, you configure the AttemptCounter security levels to be reset to zero when their started DelayTimer elapses.

Required modules

- ▶ Required diagnostic stack modules: Dcm



Configuring the Dcm

Step 1

To configure the AttemptCounter security levels to be reset to zero when their started DelayTimer elapses, configure the `DcmDspSecurityResetAttemptCounterOnTimeout` parameter to TRUE.

NOTE**Storage of a reset AttemptCounter upon expiry of a DelayTimer**

If this behavior is configured and AttemptCounters are stored to non-volatile memory as described in [Section 4.4.2.3.1, “AttemptCounter and DelayTimer”](#), also configure the maximum amount of time available for the storage of a reset AttemptCounter as described in: [Section 4.4.3.2.2, “Creating diagnostic sessions”](#).

4.4.3.3.5. Configuring AttemptCounter incrementation limiting

In this section, you enable the AttemptCounters of security levels to increment past the limit configured in their respective `DcmDspSecurityNumAttDelay` parameters.

Required modules



- Required diagnostic stack modules: Dcm



Configuring the Dcm

Step 1

To configure the AttemptCounter of security levels to continue incrementing even past the limit configured in their respective `DcmDspSecurityNumAttDelay` parameters, configure the `DcmDspSecurityIncrementAttemptCounterPastLimit` parameter to TRUE.

NOTE

Maximum AttemptCounter value



Regardless whether AttemptCounters are configured to increment past their limit values, the maximum value that an AttemptCounter can reach is 255.

4.4.3.3.6. Configuring legacy AttemptCounter handling

In this section, you configure non-volatile AttemptCounter storage using the legacy `Rte_DcmSecGet/SetNumAtt()` interfaces.

Required modules

- Required diagnostic stack modules: Dcm



Configuring the Dcm

Step 1

Disable non-volatile storage of all AttemptCounters. To do this, configure all `DcmDspSecurityAttemptCounterEnabled` parameters belonging to all `DcmDspSecurityRow` entries to FALSE. For more information, see [Section 4.4.3.3.12, “Configuring a security level's AttemptCounter to be stored to non-volatile memory”](#).

Step 2

Configure the `DcmDspSecurityLegacyAttemptCountersHandling` parameter to TRUE.

**NOTE****Limitations of legacy AttemptCounter storage behavior**

If legacy handling is used, the following limitations apply:

- ▶ AttemptCounter values are not restored at boot-up.
- ▶ No function related to restoring AttemptCounters at boot-up is available.
- ▶ No function related to DelayTimer initialization following the restoring of AttemptCounters at boot-up is available.
- ▶ The AttemptCounter cannot be reset after a DelayTimer has expired.

4.4.3.3.7. Configuring a security level DelayTimer

In this section, you configure the `DelayTimer` value for a security level.

Required modules

- ▶ Required diagnostic stack modules: `Dcm`



Configuring the Dcm

Step 1

To configure the value of the `DelayTimer` for a security level, enter the desired value in the `DcmDspSecurityDelayTime` parameter of the security level's `DcmDspSecurityRow` entry.

**NOTE****Zero-value delay timers**

If a security level's `DelayTimer` is configured to zero and the `AttemptCounter` for the security level is not stored to non-volatile memory (the `DcmDspSecurityAttemptCounterEnabled` parameter of the security level's `DcmDspSecurityRow` entry is configured to `FALSE`), or resetting `AttemptCounters` upon `DelayTimer` expiry is disabled (the `DcmDspSecurityResetAttemptCounterOnTimeout` parameter is configured to `FALSE`), time-out of the `DelayTimer` is instantaneous, i.e. the security level does not become inaccessible upon reaching the `AttemptCounter`.

If a security level's `DelayTimer` is configured to zero and the `AttemptCounter` for the security level is stored to non-volatile memory (the `DcmDspSecurityAttemptCounterEnabled` parameter of the security level's `DcmDspSecurityRow` entry is configured to `TRUE`), and resetting `AttemptCounters` upon `DelayTimer` expiry is enabled (the `DcmDspSecurityResetAttemptCounterOnTimeout` parameter is configured to `TRUE`), time-out of the `DelayTimer` does not necessarily make the security level accessible instantaneously. The security level becomes accessible only once the reset `AttemptCounter` has been stored to non-volatile memory. This makes a `DelayTimer` configured to zero to appear to have a non-zero value, since the security level will actually remain inaccessible for the amount of time necessary for the `AttemptCounter` storing operation to complete. This amount of time is, itself, limited by the `DcmDspSecurityMaxAttemptCounterWriteTime` parameter.

4.4.3.3.8. Configuring infinite DelayTimers

In this section, you configure an infinite value for the `DelayTimer`.

Required modules

- ▶ Required diagnostic stack modules: `Dcm`



Configuring the `Dcm`

Step 1

To configure the value of the `DelayTimer` for a security level to be infinite, i.e. to make this security level inaccessible until the next reboot, configure the `DcmDspSecurityDelayTimeInfinite` parameter of the security level's `DcmDspSecurityRow` entry to `TRUE`.

**NOTE****Security levels are permanently inaccessible only until reboot**

A running infinite `DelayTimer` applies only until the next reboot of the ECU. Because at boot-time, the `DelayTimer` (if needed) which is to be applied to a security level is reevaluated based upon whether:

- ▶ forced `OnBootDelayTimers` are configured (see [Section 4.4.3.3.3.9, “Configuring OnBootDelayTimers”](#) and [Section 4.4.3.3.3.10, “Configuring forced OnBootDelayTimers”](#))
- ▶ `AttemptCounter` values are retrieved from non-volatile memory (The application may decide to provide an `AttemptCounter` value to the `Dcm` which is smaller than the surpassed value that caused the `DelayTimer` to be started prior to reboot.)

A permanently locked security level may be unlocked either after a delay on reboot or instantaneously upon reboot.

4.4.3.3.9. Configuring OnBootDelayTimers

In this section, you configure the `OnBootDelayTimer` value for a security level.

Required modules

- ▶ Required diagnostic stack modules: `Dcm`



Configuring the `Dcm`

Step 1

To configure the value of the `OnBootDelayTimer` for a security level, configure this as the value of the `DcmDspSecurityDelayTimeOnBoot` parameter of the security level's `DcmDspSecurityRow` entry.

**NOTE****OnBootDelayTimer applicability**

An OnBootDelayTimer only applies if one of the following conditions is true:

- ▶ The OnBootDelayTimer has a higher value than the regular DelayTimer and forced OnBootDelayTimers are not configured for this security level (see [Section 4.4.3.3.10, "Configuring forced OnBootDelayTimers"](#))
- ▶ Forced OnBootDelayTimers are configured for this security level (see [Section 4.4.3.3.10, "Configuring forced OnBootDelayTimers"](#))

4.4.3.3.10. Configuring forced OnBootDelayTimers

In this section, you configure the OnBootDelayTimer value for a security level to always be started upon boot-up, regardless of the retrieved value of its AttemptCounter.

Required modules

- ▶ Required diagnostic stack modules: Dcm



Configuring the Dcm

Step 1

To configure the OnBootDelayTimer for a security level to be always started on boot-up, configure the DcmDspSecurityDelayTimeOnBootOverride parameter of the security level's DcmDspSecurityRow entry to TRUE.

NOTE**Forced OnBootDelayTimers prioritization**

A forced OnBootDelayTimer overrides a DelayTimer which would apply if the value of the AttemptCounter retrieved from non-volatile memory surpasses the configured limit. This applies even if such an AttemptCounter is configured to be infinite (see [Section 4.4.3.3.8, "Configuring infinite DelayTimers"](#)).

Permanent inaccessibility of a security level as a result of a failure to retrieve the value of the AttemptCounter at boot-up overrides any other OnBootDelayTimers (see [Section 4.4.3.3.3, "Configuring infinite DelayTimers upon AttemptCounter retrieval failure"](#)).

4.4.3.3.11. Configuring a security level's AttemptCounter limit

In this section, you configure the limit of the AttemptCounter for a security level. When the AttemptCounter reaches this limit, the DelayTimer starts.

Required modules



- ▶ Required diagnostic stack modules: Dcm



Configuring the Dcm

Step 1

To configure the AttemptCounter limit for a security level, configure it as the value of the `DcmDspSecurityNumAttDelay` parameter of the security level's `DcmDspSecurityRow` entry.

NOTE



Incrementing the AttemptCounter past its configured limit

You can configure the AttemptCounter to continue incrementing its value past its configured limit, as explained in: [Section 4.4.3.3.3.5, “Configuring AttemptCounter incrementation limiting”](#). However, the maximum value which an AttemptCounter for a security level can reach is limited to 255.

4.4.3.3.12. Configuring a security level's AttemptCounter to be stored to non-volatile memory

In this section, you configure the AttemptCounter for a security level to be stored to non-volatile memory. Upon boot-up, the values of the AttemptCounters of security levels are retrieved from non-volatile memory. The actual storing and restoring operations are performed by the application software.

Required modules

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: Rte if you use ClientServerInterfaces



Configuring the Dcm

Step 1

To configure the AttemptCounter of a security level to be stored to non-volatile memory, configure the `DcmDspSecurityAttemptCounterEnabled` parameter of the security level's `DcmDspSecurityRow` entry to TRUE.

Step 2

If the chosen interface (value of the `DcmDspSecurityUsePort` parameter) is `USE_SECURITY_SYNCH_FNC` or `USE_SECURITY_ASYNCH_FNC`:

Step 2.1

Enable the `DcmDspSecuritySetAttemptCounterFnc` parameter.

Step 2.2

Configure the symbol name of the callout function used for the AttemptCounter store operation as the value of the `DcmDspSecuritySetAttemptCounterFnc` parameter.



Step 2.3

Enable the `DcmDspSecurityGetAttemptCounterFnc` parameter.

Step 2.4

Configure the symbol name of the callout function used for the AttemptCounter load operation as the value of the `DcmDspSecurityGetAttemptCounterFnc` parameter.

If the chosen interface (value of the `DcmDspSecurityUsePort` parameter) is `USE_SECURITY_SYNCH_CLIENT_SERVER` or `USE_SECURITY_ASYNC_CLIENT_SERVER`, no other steps are needed in order to configure the `Dcm`.

4.4.3.4. Configuring diagnostic services

This chapter explains how to configure internally-managed or externally-managed diagnostic services and subfunctions. The configuration consists of the following steps:

- ▶ [Section 4.4.3.4.1, “Configuring internally-managed diagnostic services and subfunctions”](#)
- ▶ [Section 4.4.3.4.2, “Configuring externally-managed diagnostic services and subfunctions”](#)
- ▶ [Section 4.4.3.4.3, “Configuring negative response suppression”](#)
- ▶ [Section 4.4.3.4.4, “Configuring access conditions”](#)
- ▶ [Section 4.4.3.4.5, “Configuring asynchronous service processing”](#)
- ▶ [Section 4.4.3.4.7, “Configuring the `SecurityAccess \(0x27\)` UDS service”](#)
- ▶ [Section 4.4.3.4.8, “Configuring the `CommunicationControl \(0x28\)` UDS service”](#)

4.4.3.4.1. Configuring internally-managed diagnostic services and subfunctions

This chapter describes how to configure internally-managed diagnostic services and subfunctions. For further information on the supported internally-managed diagnostic services and subfunctions, see [Section 2.3.1.2, “UDS services”](#) or [Section 2.3.1.3, “OBD services”](#).

4.4.3.4.1.1. Configuring internally-managed diagnostic services

In this section, you configure an internally-managed diagnostic service.

For more information on diagnostic services, see [Section 4.4.2.5, “Diagnostic services”](#).

Required feature: Verification of service identifier

Required modules:



- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Step 1

To configure a new diagnostic service, add an entry in the **Diagnostic Services** table.

Step 2

In the `DcmDsdSidTabServiceId` parameter, enter the service ID of the diagnostic service.

Step 3

Clear the `DcmDsdSidTabFnc` parameter.

4.4.3.4.1.2. Configuring internally-managed subfunctions

In this section, you configure an internally-managed subfunction of a diagnostic service.

Required feature: Check format and subfunction support

Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Step 1

In the **SubService Configuration** table, add an entry.

Step 2

In the `DcmDsdSubServiceId` parameter, enter the subfunction ID of the subfunction as decimal value.

TIP



Entering hexadecimal subfunction IDs

The parameter is displayed by default in decimal format. You can also display and enter the subfunction ID as a hexadecimal value, e.g. 0x10. To do so, right-click the table cell that contains the subfunction ID and select **Autodisplay as HEX**.

Step 3

Disable the `DcmDsdSubServiceFnc` parameter.

**NOTE****Formatting checks**

Checks for the formatting of the request message are handled internally for internally-managed diagnostic services.

4.4.3.4.2. Configuring externally-managed diagnostic services and subfunctions

This chapter describes how to program and configure externally-managed diagnostic services and subfunctions.

For more information on diagnostic services, see [Section 4.4.2.5.5, “Externally managed diagnostic service handlers and subfunction handlers”](#).

4.4.3.4.2.1. Programming externally-managed diagnostic services

If you want to use an externally-managed diagnostic service, you must program your own diagnostic service handler.

Program the service handler function according to the following pattern:

```
FUNC(Std_ReturnType, DCM_CODE) <ServiceHandlerName>(
    Dcm_OpStatusType OpStatus,
    Dcm_MsgContextType* pMsgContext
)
```

Element	Description
Dcm_OpStatusType OpStatus	Type of the current call to the diagnostic service handler: <ul style="list-style-type: none"> ▶ DCM_INITIAL: initial call of this service handler. ▶ DCM_PENDING: subsequent call of this service handler after it has returned a DCM_E_PENDING error code in a previous call. This allows to split up the processing to more than one Dcm_MainFunction() cycle. ▶ DCM_CANCEL: call to cancel the service handler. The service handler should terminate all processing and reset all internal variables and states to a default value. All results of processing performed from the last call with a DCM_INITIAL OpStatus up to this call are considered null. The return code of this call is not internally considered by the Dcm. ▶ DCM_FORCE_RCRRP_OK: call that notifies the service handler that a RequestCorrectlyReceivedResponsePending (0x7f service ID



Element	Description
	0x78) NRC was successfully sent. This NRC was requested by the service handler to be sent by returning a <code>DCM_E_FORCE_RCRRP</code> error code.
<code>Dcm_MsgContextType*</code> <code>pMsgContext</code>	<p>Pointer to the context of the service handling. This is a pointer to a structure that contains the following parameters:</p> <ul style="list-style-type: none"> ▶ <code>Dcm_MsgType reqData</code>: a pointer to the service request message minus the service ID. ▶ <code>Dcm_MsgLenType reqDataLen</code>: informs the application about how many data bytes are valid, which begins from <code>reqData</code>, i.e. the length of the request minus the service ID. ▶ <code>Dcm_MsgType resData</code>: a pointer to the response buffer that the externally-managed service handler must use. The externally-managed service handler must write the data of the positive response to this pointer, which begins with the byte after the positive response code. The positive response code is handled internally by the <code>Dcm</code>. ▶ <code>Dcm_MsgLenType resDataLen</code>: informs the <code>Dcm</code> about how many data bytes are valid, beginning from <code>resData</code>, i.e. the length of the positive response minus the positive response code. ▶ <code>Dcm_MsgLenType resMaxDataLen</code>: informs the application about the maximum amount of data that can be written to <code>resData</code>. ▶ <code>Dcm_MsgAddInfoType msgAddInfo</code>: additional information. A bit field structure that contains the following information: <ul style="list-style-type: none"> ▶ <code>unsigned int reqType</code>: type of request, i.e. physical or functional. ▶ <code>unsigned int suppressPosResponse</code>: notifies the application that the positive response must be suppressed. If this bit is set to <code>TRUE</code>, the application can force the positive response to be suppressed. ▶ <code>Dcm_IdContextType idContext</code>: contains the service ID. ▶ <code>PduIdType dcmRxPduId</code>: the RxPduId on which the request was received.

The service handler can return the following valid error codes:

- ▶ `E_OK`: request processing was finished successfully. However, the `Dcm_ExternalProcessingDone()` API still needs to be called beforehand to complete processing.
- ▶ `E_NOT_OK`: request processing was finished unsuccessfully. However, the `Dcm_ExternalSetNegResponse()` API still needs to be called beforehand to set the NRC to be sent, followed by a call to the `Dcm_ExternalProcessingDone()` API to end processing.



- ▶ **DCM_E_PENDING:** request processing is pending and needs a further invocation of the service handler function. Return this error code to split the processing of a request into fragments that are not too large so they do not interfere with scheduling.
- ▶ **DCM_E_FORCE_RCRP:** the application requests that the Dcm transmits a RequestCorrectlyReceivedResponsePending (0x7f service ID 0x78) NRC immediately.

To complete processing and transmit a positive or negative response, the externally managed service handler shall call the following API:

```
void Dcm_ExternalProcessingDone(
    Dcm_MsgContextType* pMsgContext
)
```

You must call this API with the same `pMsgContext` pointer that it received from the `Dcm` before returning `E_OK`.

To transmit a NRC, the externally-managed service handler must call the following API:

```
void Dcm_ExternalSetNegResponse (
    Dcm_MsgContextType* pMsgContext,
    Dcm_NegativeResponseCodeType ErrorCode
)
```

This API must be called with the same `pMsgContext` pointer that it has received from the `Dcm`. The NRC must be written to `ErrorCode` before it calls the `Dcm_ExternalProcessingDone()` API and returns `E_NOT_OK`.

4.4.3.4.2.2. Configuring externally-managed diagnostic services

In this section, you configure externally-managed diagnostic services.

Required feature: External/user-defined service handler

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Prerequisite:

- The diagnostic service handler function is programmed. For more information, see [Section 4.4.3.4.2.1, “Programming externally-managed diagnostic services”](#).

Step 1

In the **Diagnostic Service Tables** table, add an entry.

Step 2

In the **Diagnostic Services** table, add an entry.

Step 3

In the `DcmDsdSidTabServiceId` parameter, enter the service ID of your externally-managed UDS or OBD service.

Step 4

Enable the `DcmDsdSidTabFnc` parameter.

Step 5

In the `DcmDsdSidTabFnc` parameter, enter the name of the diagnostic service handler function.

4.4.3.4.2.3. Programming externally-managed subfunctions

If you want to use an externally-managed subfunction, you must program your own subfunction handler.

Program the subfunction handler according to the following pattern:

```
FUNC(Std_ReturnType, DCM_CODE) <SubfunctionHandlerName>(
    Dcm_OpStatusType OpStatus,
    Dcm_MsgContextType* pMsgContext
)
```

Element	Description
<code>Dcm_OpStatusType</code> <code>OpStatus</code>	Type of the current call to the subfunction handler: <ul style="list-style-type: none"> ▶ <code>DCM_INITIAL</code>: initial call of this subfunction handler. ▶ <code>DCM_PENDING</code>: subsequent call of this subfunction handler after it has returned a <code>DCM_E_PENDING</code> error code in a previous call. This allows to split up the processing to more than one <code>Dcm_MainFunction()</code> cycle. ▶ <code>DCM_CANCEL</code>: call to cancel the subfunction handler. The subfunction handler should terminate all processing and reset all internal variables and



Element	Description
	<p>states to a default value. All results of processing performed from the last call with a <code>DCM_INITIAL</code> <code>OpStatus</code> up to this call are considered null. The return code of this call is not internally considered by the <code>Dcm</code>.</p> <ul style="list-style-type: none"> ▶ <code>DCM_FORCE_RCRRP_OK</code>: call that notifies the subfunction handler that a <code>RequestCorrectlyReceivedResponsePending</code> (0x7f service ID 0x78) NRC was successfully sent. This NRC was requested by the subfunction handler to be sent by returning a <code>DCM_E_FORCE_RCRRP</code> error code.
<code>Dcm_MsgContextType*</code> <code>pMsgContext</code>	<p>Pointer to the context of the subfunction handling. This is a pointer to a structure that contains the following parameters:</p> <ul style="list-style-type: none"> ▶ <code>Dcm_MsgType reqData</code>: a pointer to the subfunction request message minus the service ID and the subfunction ID. The service ID is removed before the parent service is called. ▶ <code>Dcm_MsgLenType reqDataLen</code>: informs the application about how many data bytes are valid, beginning from <code>reqData</code>, i.e. the length of the request minus the service ID and the subfunction ID. ▶ <code>Dcm_MsgType resData</code>: a pointer to the response buffer that the externally-managed subfunction handler must use. The externally-managed subfunction handler must write the data of the positive response to this pointer, beginning with the data byte after the positive response code. The positive response code is handled internally by the <code>Dcm</code>. The ID of the subfunction must be written by the subfunction handler. ▶ <code>Dcm_MsgLenType resDataLen</code>: informs the <code>Dcm</code> about how many data bytes are valid, beginning from <code>resData</code>, i.e. the length of the positive response minus the positive response code. ▶ <code>Dcm_MsgLenType resMaxDataLen</code>: informs the application about the maximum amount of data that can be written to <code>resData</code>. ▶ <code>Dcm_MsgAddInfoType msgAddInfo</code>: additional information. A bit field structure that contains the following information: <ul style="list-style-type: none"> ▶ <code>unsigned int reqType</code>: type of request, i.e. physical or functional. ▶ <code>unsigned int suppressPosResponse</code>: notifies the application that the positive response must be suppressed. If this bit is set to <code>TRUE</code>, the application can force the positive response to be suppressed. ▶ <code>Dcm_IdContextType idContext</code>: contains the service ID. ▶ <code>PduIdType dcmRxPduId</code>: the RxPduId on which the request was received.



The subfunction handler can return the following valid error codes:

- ▶ `E_OK`: request processing was finished successfully. However, the `Dcm_ExternalProcessingDone()` API does not need to be called since this functionality is handled by the parent service handler.
- ▶ `E_NOT_OK`: request processing was finished unsuccessfully. However, the `Dcm_ExternalSetNegResponse()` API still needs to be called beforehand to set the NRC to be sent.
- ▶ `DCM_E_PENDING`: Request processing is pending and needs a further invocation of the subfunction handler function. Return this error code to split the processing of a request into fragments that are not too large so they do not interfere with scheduling.
- ▶ `DCM_E_FORCE_RCRRP`: the application requests that the `Dcm` transmits a `RequestCorrectlyReceivedResponsePending` (0x7F service ID 0x78) NRC immediately.

To transmit a NRC, the externally-managed subfunction handler must call the following API:

```
void Dcm_ExternalSetNegResponse(
    Dcm_MsgContextType* pMsgContext,
    Dcm_NegativeResponseCodeType ErrorCode
)
```

This API must be called with the same `pMsgContext` pointer that it received from the `Dcm` and the NRC must be written to `ErrorCode` before it returns `E_NOT_OK`.

WARNING



Complete processing of subfunction handler

Do not call the `Dcm_ExternalProcessingDone()` API to complete the processing of a subfunction handler. This API is only needed for externally-managed diagnostic service handlers.

4.4.3.4.2.4. Configuring externally-managed subfunctions

In this section, you configure externally-managed subfunctions.

Required feature: External/user-defined subservice handler except for SID: 0x10, 0x27, 0x3E, and 0x31

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Prerequisite:

- The subfunction handler function is programmed. For more information, see [Section 4.4.3.4.2.3, “Programming externally-managed subfunctions”](#).
- A diagnostic service is configured.

Step 1

Enable the **DcmDsdSidTabSubfuncAvail** parameter for the diagnostic service.

Step 2

In the **SubService Configuration** table, add an entry.

Step 3

In the **DcmDsdSubServiceId** parameter, enter the subfunction ID of your externally-managed subfunction.

Step 4

Enable the **DcmDsdSubServiceFnc** parameter.

Step 5

In the **DcmDsdSubServiceFnc** parameter, enter the name of the subfunction handler function.

4.4.3.4.3. Configuring negative response suppression

In this section, you configure whether suppression of NRCs for functional requests should be handled in accordance to ISO14229-1:2013 or ISO14229-1:2006.

For more information on negative response suppression, see [Section 4.4.2.5.4, “Negative and positive response suppression”](#)

Required feature: Suppression of responses according to ISO 14229 2006 and 2013

Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Step 1

In the **DcmNRCPolicyISOVersion** parameter, select the ISO standard on which your configuration is based.



4.4.3.4.4. Configuring access conditions

This chapter explains how to configure access conditions for diagnostic services and subfunctions. These access conditions are checked by the Dcm before a diagnostic service or subfunction is allowed to run. The diagnostic service or subfunction is only processed if all access conditions are met.

For more information on access conditions, see [Section 4.4.2.5.1, “Diagnostic service access”](#)

You can configure the following access conditions:

- ▶ Check if a diagnostic service or subfunction is allowed to run in a specific diagnostic session.
- ▶ Check if mode conditions are met.
- ▶ Check if the required security level is met.
- ▶ Further manufacturer-specific or supplier-specific checks.

4.4.3.4.4.1. Configuring the availability of diagnostic services in diagnostic sessions

In this section, you configure in which diagnostic sessions a diagnostic service is allowed to run.

Required feature: Verification of diagnostic session

Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Step 1

In the **Session References** table of the diagnostic service, add an entry.

Step 2

In the new entry, reference the diagnostic session in which the diagnostic service is allowed to run.

If the diagnostic service should be available in all diagnostic sessions, leave the **Session References** table empty.

**NOTE****DiagnosticSessionControl service is available in all diagnostic sessions**

You do not need to configure references to diagnostic sessions for the `DiagnosticSessionControl` (0x10) service. This service is available by default in all diagnostic sessions since this service is used to switch to other diagnostic sessions.

4.4.3.4.4.2. Configuring the availability of subfunctions in diagnostic sessions

In this section, you configure in which diagnostic sessions a subfunction of a diagnostic service is allowed to run.

The configuration of this feature is identical for both internally-managed and externally-managed subfunctions.

Required feature: Verification of diagnostic session

Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Step 1

In the **Session Levels** table, add a new entry.

Step 2

In the new entry, reference the diagnostic sessions in which the service can be used.

If the diagnostic service should be available in the same diagnostic sessions as the parent diagnostic service, leave the **Session Levels** table empty.

NOTE**Availability of subfunctions**

The diagnostic sessions in which a subfunction is available must be a subset of the diagnostic sessions in which the parent diagnostic service is available.

4.4.3.4.4.3. Configuring security access verification for diagnostic services

In this section, you configure at which security levels a diagnostic service is allowed to run.

Required feature: Verification of service security access levels

Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Prerequisite:

- A security level is configured.

Step 1

In the **Security Level References** table, add an entry.

Step 2

In the new entry, reference the security level at which the diagnostic service is allowed to run.

If the diagnostic service could be available at all security levels, leave the **Security Level References** table empty.

NOTE**SecurityAccess (0x27) service is available at all security levels**

You do not have to configure references to security levels for the SecurityAccess (0x27) service handler. This service is available by default at all security levels since this service is used to switch to other security levels.

4.4.3.4.4. Configuring security access verification for subfunctions

In this section, you configure at which security levels a subfunction of a diagnostic service is allowed to run.

The configuration of this feature is identical for both internally-managed and externally-managed subfunctions.

Required feature: Verification of subfunction security access levels

Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Prerequisite:

- A security level is configured.

Step 1

In the **Security Levels** table, add an entry.

Step 2

In the new entry, reference the security level at which the diagnostic service is allowed to run.



If the subfunction should be available at the same security levels as the parent diagnostic service, leave the **Security Levels** table empty.

NOTE**Availability of subfunctions**

The security levels at which a subfunction is available must be a subset of the security levels in which the parent diagnostic service is available.

4.4.3.4.4.5. Configuring mode rule verification for diagnostic services

In this section, you configure a mode condition. This mode condition is checked before a diagnostic service is allowed to run.

The configuration of this feature is identical for both internally-managed and externally-managed diagnostic services.

Required feature: Verification of service ModeRules

Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Prerequisite:

- A mode rule is configured in the Dcm.

Step 1

Enable the `DcmDsdSubServiceModeRuleRef` parameter.

Step 2

In the `DcmDsdSubServiceModeRuleRef` parameter, select your mode rule.

4.4.3.4.4.6. Configuring mode rule verification for subfunctions

In this section, you configure a mode condition. This mode condition is checked before a subfunction is allowed to run.

The configuration of this feature is identical for both internally-managed and externally-managed subfunctions.

Required feature: Verification of ModeRules



Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Prerequisite:

- A mode rule is configured in the Dcm.

Step 1

In the subfunction, select the `DcmDsdSubServiceModeRuleRef` parameter.

Step 2

In the `DcmDsdSubServiceModeRuleRef` parameter, select your mode rule.

4.4.3.4.4.7. Configuring a manufacturer/supplier-specific verification

In this section, you configure manufacturer-specific or supplier-specific access verification, i.e. the manufacturer indication/confirmation and the supplier indication/confirmation.

For more information on manufacturer-specific and supplier-specific access verification, see [Section 4.4.2.5.2, “Manufacturer-specific or supplier-specific access verification”](#)

Required features:

- ▶ Verification of the manufacturer application environment/permission
- ▶ Verification of the supplier application environment/permission

Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: Rte



Configuring the Dcm

Step 1

If you want to enable the manufacturer indication/confirmation:

Step 1.1

Enable the `Enable Diagnostic Request Notification for Manufacturer` parameter.

Step 1.2

In the **Manufacturer Notification Functions** table, add an entry.



This entry allows the `Dcm` to generate *RequiredPorts* that are named according to the following pattern: `ServiceRequestManufacturerNotification_<Name>`. `<Name>` stands for the name of the entry within the **Manufacturer Notification Functions** table that implements the `ServiceRequestNotification ClientServerInterface`. These ports contain both the indication operations and confirmation operations. You can connect these ports to compatible *ProvidedPorts* that are supplied by the application.

Step 2

If you want to enable the supplier indication/confirmation:

Step 2.1

Enable the `DcmRequestSupplierNotificationEnabled` parameter.

Step 2.2

Add a supplier notification function entry to the **Supplier Notification Functions** table.

This entry allows the `Dcm` to generate *RequiredPorts* that are named according to the following pattern: `ServiceRequestSupplierNotification_<Name>`. `<Name>` stands for the name of the entry within the **Supplier Notification Functions** table that implements the `ServiceRequestNotification ClientServerInterface`. These ports contain both the indication operations and confirmation operations. You can connect these ports to compatible *ProvidedPorts* that are supplied by the application.

4.4.3.4.4.8. Disabling writing memory diagnostic services while driving

If diagnostic services need to be disabled under some conditions, for example while the car is driving, you can use one of the following features:

- ▶ [the manufacturer/supplier-specific verification](#)
- ▶ [the mode rule verification](#)



Disabling a diagnostic service using the manufacturer/supplier-specific verification

Step 1

Enable the manufacturer or the supplier indication/confirmation (e.g. enable `DcmRequestManufacturerNotificationEnabled` or `DcmRequestSupplierNotificationEnabled`).

Step 2

In the `Xxx_Indication()` operation of the `ServiceRequestNotification ClientServerInterface`, verify the condition under which the service needs to be disabled.

Step 3

If the condition under which the service needs to be disabled is fulfilled, configure the `Xxx_Indication()` operation to return `E_REQUEST_NOT_ACCEPTED` or `E_NOT_OK`.

For example, in the `Xxx_Indication()` operation, a diagnostic service that writes to memory can be disabled while driving.



The service that needs to be disabled can be checked via the `SID` parameter of the `Xxx_Indication()` interface. Also, the speed of the car needs to be checked. If both conditions are fulfilled, then:

- ▶ The `Xxx_Indication()` interface returns `E_REQUEST_NOT_ACCEPTED`. The `Dcm` module discards the request without any response.
- ▶ The `Xxx_Indication()` interface returns `E_NOT_OK`. The `Dcm` module sends a response with the negative response code set in the error code parameter from the `Xxx_Indication()` interface.

To configure the manufacturer/supplier-specific verification, see [Section 4.4.3.4.4.7, “Configuring a manufacturer/supplier-specific verification”](#).



Disabling a diagnostic service using the mode rule verification

Step 1

Enable the `DcmDsdSidTabModeRuleRef` configuration parameter of the service that needs to be disabled.

Step 2

In the `DcmDsdSidTabModeRuleRef`, reference a `DcmModeRule` that references a `DcmModeCondition` belonging to a `ModeDeclarationGroupPrototype` that shall be set by the application while driving.

Step 3

If such a service is requested, the response is the negative response code set in the `DcmModeRule` configuration. Configure the response code with the `DcmModeRuleNrcValue` configuration parameter.

To configure the mode rule verification, see [Section 4.4.3.4.4.5, “Configuring mode rule verification for diagnostic services”](#).

4.4.3.4.5. Configuring asynchronous service processing

In this section, you configure the asynchronous processing to increase the call times spent in the service handler. Use this feature only if large amounts of data are handled.

Asynchronous service handling is configured the same way for both internally and externally-managed service handlers.

For more information on asynchronous service processing, see [Section 4.4.2.5.6, “Asynchronous service processing via separate task”](#)

Required feature: Asynchronous service processing via separate task

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: `Os` and `Rte`



Configuring the Dcm

Step 1

Enable the `DcmAsyncServiceExecution` parameter for the diagnostic service.



Configuring the Os and Rte

Prerequisite:

- The `Rte` module is enabled and configured.
- At least two tasks are configured in the `Os` module.

Step 1

In the **Rte Editor**, map the `TimingEvent_MainFunction` `BswTimingEvent` event to a task.

Step 2

Ensure one of the following configurations for the `Os`:

- ▶ An extended task exists with a priority lower than the priority of the task to which the `TimingEvent_MainFunction` of the `Dcm` is mapped.
- ▶ A basic task exists with an `OsTaskActivation` of at least 2 and with a priority lower than the priority of the task to which the `TimingEvent_MainFunction` of the `Dcm` is mapped.

For more information on configuring `Os` tasks, see the EB tresos AutoCore OS documentation.

Step 3

In the **Rte Editor**, map the `BswInternalTriggerOccuredEvent_Async` `BswInternalTriggerOccuredEvent` event to the added basic or extended task.

4.4.3.4.6. Configuring the `ReadScalingDataByIdentifier` (0x24) UDS service

This chapter explains how to configure the `ReadScalingDataByIdentifier` (0x24) UDS service. For background information, see [Section 4.4.2.5.7, “`ReadScalingDataByIdentifier` \(0x24\) UDS service”](#).

4.4.3.4.6.1. Configuring the `ReadScalingDataByIdentifier` (0x24) UDS service handler

This section explains how to configure the `ReadScalingDataByIdentifier` (0x24) service handler.

Required feature: UDS service `ReadScalingDataByIdentifier` (0x24)

Required modules:



- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: PduR and EcuC



Configuring the Dcm

Prerequisite:

- At least one **Diagnostic Service Table** is configured.

Step 1

Add an entry in the **Diagnostic Services** table.

Step 2

In the `DcmDsdSidTabServiceId` parameter, enter `0x24`. This is the service ID of the `ReadScalingDataByIdentifier` (`0x24`) UDS service.

Step 3

Configure any access conditions necessary for this diagnostic service as described in the sections:

- ▶ [Section 4.4.3.4.4.1, “Configuring the availability of diagnostic services in diagnostic sessions”](#)
- ▶ [Section 4.4.3.4.4.3, “Configuring security access verification for diagnostic services”](#)
- ▶ [Section 4.4.3.4.4.5, “Configuring mode rule verification for diagnostic services”](#)

4.4.3.4.6.2. Configuring the Xxx_GetScalingInformation operation



Configuring the Xxx_GetScalingInformation operation

Step 1

Configure the `DcmDspDataUsePort` to indicate the type of interface to be used to implement the `Xxx_GetScalingInformation` operation to retrieve the scaling information.

- ▶ **USE_DATA_SYNCH_CLIENT_SERVER:** the Dcm accesses the data using an R-Port requiring a synchronous ClientServerInterface. The R-Port is named `DataServices_{Data}` where `{Data}` is the name of the container `DcmDspData`.
- ▶ **USE_DATA_ASYNC_CLIENT_SERVER:** the Dcm accesses the data using an R-Port requiring an asynchronous ClientServerInterface. The R-Port is named `DataServices_{Data}` where `{Data}` is the name of the container `DcmDspData`.
- ▶ **USE_DATA_SYNCH_FNC:** the Dcm accesses the data using the functions that are defined in parameters of type `EcuFunctionNameDef` (but without `DcmDspDataReadDataLengthFnc`) in the `DcmDspData` container. The return value `DCM_E_PENDING` is not allowed. `OpStatus` does not exist as IN parameter.



In this case, also configure `DcmDspDataGetScalingInfoFnc` with a valid C function name.

- ▶ USE _DATA _ASYNCH _FNC: the Dcm accesses the data using the functions that are defined in parameters of type `EcucFunctionNameDef` (but without `DcmDspDataReadDataLengthFnc`) in the `DcmDspData` container. The return value `DCM_E_PENDING` is allowed. `OpStatus` exists as IN parameter.

In this case, also configure `DcmDspDataGetScalingInfoFnc` with a valid C function name.

4.4.3.4.6.3. Configuring `DcmDspDataScalingInfoSize`

The configuration parameter `DcmDspDataScalingInfoSize` provides the size in bytes of the scaling information for the referencing `DcmDspData` element.



Configuring `DcmDspDataScalingInfoSize`

Step 1

- ▶ Specify a non-negative integer value for the `DcmDspDataScalingInfoSize` parameter.

NOTE



Configuring `DcmDspDataScalingInfoSize` is mandatory in the following scenario

If the configuration parameter `DcmDspDataGetScalingInfoFnc` for a `DcmDspData` element is configured, then the configuration parameter `DcmDspDataScalingInfoSize` must also be configured in the `DcmDspDataInfo` element referenced via `DcmDspDataInfoRef` in that `DcmDspData` element. Otherwise, a configuration error is reported.

4.4.3.4.6.4. Configuring `DcmDspDidReadScalingData`

The configuration container `DcmDspDidReadScalingData` consists of the following two configuration parameters:

- ▶ `DcmDspDidReadScalingDataSecurityLevelRef` to define the permissions for the access of the UDS service `ReadScalingDataByIdentifier` (0x24).
- ▶ `DcmDspDidReadScalingDataSessionRef` to define the diagnostic sessions in which the access of the UDS service `ReadScalingDataByIdentifier` (0x24) for the referencing `DcmDspDid` DID is allowed.

**NOTE****DcmDspDidReadScalingData is not defined by AUTOSAR**

The AUTOSAR specifications 4.0.3 and R20-11 do not define the configuration container `DcmDspDidReadScalingData`. Therefore, the configuration parameters `DcmDspDidReadScalingDataSecurityLevelRef` and `DcmDspDidReadScalingDataSessionRef` are not part of the specification.

If the configuration container `DcmDspDidReadScalingData` in a `DcmDspDidInfo` element is disabled (i.e. not configured), the `Dcm` cannot process any `ReadScalingDataByIdentifier` (0x24) service request for a `DcmDspDid` DID referencing that `DcmDspDidInfo` element via `DcmDspDidInfoRef`. Consequently, the `Dcm` transmits a negative response with NRC equal to 0x31 (requestOutOfRange) in such a scenario.

4.4.3.4.6.4.1. Configuring DcmDspDidReadScalingDataSecurityLevelRef

If configured, `DcmDspDidReadScalingDataSecurityLevelRef` provides references to the `DcmDspSecurityRows`, defining the security levels in which the access for the `ReadScalingDataByIdentifier` (0x24) UDS service should be allowed for the referencing `DcmDspDid` element.

If no reference to `DcmDspSecurityRows` is configured, no check of security level is performed. In this case, `ReadScalingDataByIdentifier` (0x24) access is allowed in any security level.



Configuring `DcmDspDidReadScalingDataSecurityLevelRef`

Step 1

- ▶ Specify the reference(s) to `DcmDspSecurityRows` to define the security levels in which the access for the `ReadScalingDataByIdentifier` (0x24) UDS service is to be allowed.

4.4.3.4.6.4.2. Configuring DcmDspDidReadScalingDataSessionRef

If configured, `DcmDspDidReadScalingDataSessionRef` provides references to the `DcmDspSessionRows`, defining the diagnostic sessions in which the access for the `ReadScalingDataByIdentifier` (0x24) UDS service is to be allowed for the referencing `DcmDspDid` element.

If no reference to `DcmDspSessionRows` is configured, no check of diagnostic session is performed. In this case, `ReadScalingDataByIdentifier` (0x24) access is allowed in any diagnostic session.



Configuring `DcmDspDidReadScalingDataSecurityLevelRef`

Step 1

- ▶ Specify the reference(s) to the `DcmDspSessionRows` to define the diagnostic sessions in which the access for the `ReadScalingDataByIdentifier` (0x24) UDS service is to be allowed.

4.4.3.4.7. Configuring the `SecurityAccess` (0x27) UDS service

This chapter explains how to configure the `SecurityAccess` (0x27) UDS service, the security levels which are available to the Dcm, and the `DelayTimer` and `AttemptCounter` behavior related to security levels.

4.4.3.4.7.1. Configuring the `SecurityAccess` (0x27) UDS service handler

In this section, you configure the `SecurityAccess` (0x27) UDS service.

Required feature: UDS service `SecurityAccess` (0x27)

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: `PduR` and `EcuC`



Configuring the `Dcm`

Prerequisite:

- At least one **Diagnostic Service Table** is configured.

Step 1

Add an entry in the **Diagnostic Services** table.

Step 2

In the `DcmDsdSidTabServiceId` parameter, enter 0x27. This is the service ID of the `SecurityAccess` (0x27) UDS service.

Step 3

Configure the `DcmDsdSidTabSubfuncAvail` parameter to TRUE.

Step 4

Clear the `DcmDsdSidTabFnc` parameter.

**Step 5**

Configure the access conditions that are necessary for this diagnostic service as described in [Section 4.4.3.4.4.1, “Configuring the availability of diagnostic services in diagnostic sessions”](#) and [Section 4.4.3.4.4.5, “Configuring mode rule verification for diagnostic services”](#).

NOTE**Availability of security level conditions**

Security level conditions cannot be configured for the SecurityAccess (0x27) UDS service.

4.4.3.4.7.2. Configuring subfunctions of the SecurityAccess (0x27) UDS service handler

In this section, you configure the SecurityAccess (0x27) UDS service.

Required feature: UDS service SecurityAccess (0x27)

Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Prerequisite:

- At least one **Diagnostic Service Table** entry for the SecurityAccess (0x27) UDS service is configured.

Step 1

For each **SecurityLevel** which is configured in the **DcmDspSecurity** table, add 2 entries in the **SubService Configuration** table of the SecurityAccess (0x27) UDS service's service table.

Step 2

In the **DcmDsdSubServiceId** parameter of the added **SubService Configuration** table entries, per **SecurityLevel**, enter the subfunction IDs of the two subfunctions. These IDs have the following numeric values:

- ▶ $2 * \text{SecurityLevel} - 1$ for the **RequestSeed** subfunction corresponding to the **SecurityLevel**
- ▶ $2 * \text{SecurityLevel}$ for the **SendKey** subfunction corresponding to the **SecurityLevel**

**NOTE****Necessity of subfunction configuration for the SecurityAccess (0x27) UDS service****UDS service**

Subfunction configuration for the SecurityAccess (0x27) UDS service is only necessary if special access conditions are necessary for transitions to certain SecurityLevels. Otherwise, this configuration may be left empty and the necessary subfunctions will be automatically handled in the background, based on the configuration in the DcmDspSecurity table.

Step 3

Clear the DcmDsdSubServiceFnc parameter.

Step 4

Configure the access conditions that are necessary for this subfunction as described in [Section 4.4.3.4.4.2, “Configuring the availability of subfunctions in diagnostic sessions”](#) and [Section 4.4.3.4.4.6, “Configuring mode rule verification for subfunctions”](#).

NOTE**Subfunction-level SecurityLevel access condition configuration for subfunctions**

The SecurityLevel conditions for subfunctions of the SecurityAccess (0x27) UDS service cannot be configured.

NOTE**External subfunctions for the SecurityAccess (0x27) UDS service**

User-implemented externally-managed subfunctions for the SecurityAccess (0x27) UDS service cannot be configured.

4.4.3.4.8. Configuring the CommunicationControl (0x28) UDS service

This chapter explains how to configure the CommunicationControl (0x28) UDS service, which ComM channels are considered for CommunicationControl (0x28) requests, and the ModeRule to reset the CommunicationControl.

This chapter contains instructions on how to configure a BswM interaction according to AUTOSAR 4.0.2 or the ModeDeclarationGroupPrototype interaction via Rte according to AUTOSAR 4.0.3 or higher.

4.4.3.4.8.1. Configuring the CommunicationControl (0x28) UDS service handler

In this section, you configure the CommunicationControl (0x28) UDS service.

Required feature: UDS service CommunicationControl (0x28)

Required modules:

- ▶ Required diagnostic stack modules: Dcm



- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Prerequisite:

- At least one **Diagnostic Service Table** is configured.

Step 1

Add an entry in the **Diagnostic Services** table.

Step 2

In the `DcmDsdSidTabServiceId` parameter, enter 0x28. This is the service ID of the `CommunicationControl` (0x28) UDS service.

Step 3

Clear the `DcmDsdSidTabFnc` parameter.

Step 4

Configure the `DcmDsdSidTabSubfuncAvail` parameter to TRUE.

Step 5

Configure any access conditions necessary for this diagnostic service as described in [Section 4.4.3.4.4.1, “Configuring the availability of diagnostic services in diagnostic sessions”](#), [Section 4.4.3.4.4.3, “Configuring security access verification for diagnostic services”](#) and [Section 4.4.3.4.4.5, “Configuring mode rule verification for diagnostic services”](#).

4.4.3.4.8.2. Configuring subfunctions of the `CommunicationControl` (0x28) UDS service

In this section, you configure the `CommunicationControl` (0x28) UDS service.

Required feature: UDS service `CommunicationControl` (0x28)

Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Prerequisite:

- At least one **Diagnostic Service Table** entry for the `CommunicationControl` (0x28) UDS service is configured.

**Step 1**

Within the `CommunicationControl (0x28)` Service Table entry, add a new entry in the **SubService Configuration** table.

Step 2

In the `DcmDsdSubServiceId` parameter of the **SubService Configuration** table entry, enter the subfunction ID of the subfunction. For more information on the supported UDS subfunctions and their subfunction ID, see [Section 2.3.1.2, “UDS services”](#).

NOTE**Limitations of `CommunicationControl (0x28)` UDS service subfunction support**

The internally-managed handling for the following subfunctions of the `CommunicationControl (0x28)` UDS service is only available if the `CommunicationControl (0x28)` service uses the ModeDeclaration-based interaction according to AUTOSAR 4.-0.3 or higher:

- ▶ `enableRxAndDisableTxWithEnhancedAddressInformation (0x04)`
- ▶ `enableRxAndTxWithEnhancedAddressInformation (0x05)`

In order to configure the `CommunicationControl (0x28)` UDS service handler to use the ModeDeclarationGroup-based interaction, see [Section 4.4.3.4.8.3, “Configuring ModeDeclaration support for the `CommunicationControl \(0x28\)` UDS service”](#).

Step 3

Clear the `DcmDsdSubServiceFnc` parameter.

Step 4

Configure the access conditions that are necessary for this subfunction as described in [Section 4.4.3.4.4.2, “Configuring the availability of subfunctions in diagnostic sessions”](#), [Section 4.4.3.4.4.4, “Configuring security access verification for subfunctions”](#) and [Section 4.4.3.4.4.6, “Configuring mode rule verification for subfunctions”](#).

4.4.3.4.8.3. Configuring ModeDeclaration support for the `CommunicationControl (0x28)` UDS service

In this section, you configure the `CommunicationControl (0x28)` UDS service to use the ModeDeclaration-interaction according to AUTOSAR 4.0.3 or higher.

Required features:

- ▶ UDS service `CommunicationControl (0x28)`
- ▶ Diagnostic mode declaration groups

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`



- ▶ Required non-diagnostic stack modules: SchM/Rte



Configuring the Dcm

Prerequisite:

- At least one **Diagnostic Service Table** entry for the `CommunicationControl` (0x28) UDS service is configured.

Step 1

To configure the `Dcm` to use ModeDeclaration-based interaction to handle the `CommunicationControl` (0x28) UDS service, enable the `DcmModeDeclarationSupport` parameter. The mode handling of the `Dcm` is handled via the `Rte`.

NOTE**ModeDeclaration-interaction cannot be selected per service**

ModeDeclaration-based interaction is not selectable per-service. This means that configuring one of the following services to use ModeDeclaration-based interaction forces all of these services to use the same mechanism:

- ▶ UDS service `DiagnosticSessionControl` (0x10)
 - ▶ UDS service `ECUReset` (0x11)
 - ▶ UDS service `CommunicationControl` (0x28)
 - ▶ UDS service `ControlDTCSetting` (0x85)
-



Configuring the Rte

Step 1

To configure the `Rte`, connect all **Required Mode Groups** to the **Provided Mode Groups** in the `Rte` Editor.

4.4.3.4.8.4. Configuring BswM interaction for the CommunicationControl (0x28) UDS service

In this section, you configure the `CommunicationControl` (0x28) UDS service to use the `BswM` callout interaction according to AUTOSAR 4.0.2 or lower.

Required features:

- ▶ UDS service `CommunicationControl` (0x28)
- ▶ Mode switching via `BswM` callouts

Required modules:



- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: BswM



Configuring the Dcm

Prerequisite:

- At least one **Diagnostic Service Table** entry for the `CommunicationControl` (0x28) UDS service is configured.

Step 1

To configure the `Dcm` to use `BswM` callout interaction to handle the `CommunicationControl` (0x28) UDS service, clear the `DcmModeDeclarationSupport` parameter. The mode handling of the `Dcm` is handled via the `BswM`.

NOTE



BswM callout interaction is not selectable per service

`BswM` callout-based interaction is not selectable per service. This means that configuring one of the following services to use the `BswM` callout-based interaction forces all of these services to use the same mechanism:

- ▶ UDS service `DiagnosticSessionControl` (0x10)
- ▶ UDS service `ECUReset` (0x11)
- ▶ UDS service `CommunicationControl` (0x28)
- ▶ UDS service `ControlDTCSetting` (0x85)



Configuring the BswM

In this section, you create a `BswM` action that calls the `Dcm_CommunicationControl<Mode_>ModeEntry()` API when the `Dcm` requests a switch to the corresponding `CommunicationMode`.

The following instruction describes in more detail how to configure the `BswM`.

Step 1

In the **Configuration** table, add an entry.

Step 2

The `BswMAction` requests the `BswM` module to switch to a `CommunicationControl`. To configure the `BswMAction`:

Step 2.1

In the **BswMAction** table, add an entry.



Step 2.2

In the `BswMAvailableActions` parameter, select `BswMUserCallout`.

Step 2.3

In the `BswMUserCalloutFunction` parameter, enter `Dcm_CommunicationControl<Mode_>ModeEntry()`.

This function acknowledges to the `Dcm` module the transition to the new `CommunicationControl`.

Step 2.4

In the **`BswMActionList`** table, add an entry.

Step 2.5

In the `BswMActionListExecution` parameter, select `BSWM_CONDITION`.

The `BswMActions` that are contained in the `BswMActionList` are only executed when specific conditions are met. The conditions are configured in the later steps.

Step 2.6

In the **`BswMActionListItem`** table, add an entry.

Step 2.7

In the `BswMActionListItemRef` parameter, select your `BswMAction`.

Step 3

The `BswMMModeRequestPort` enables the `Dcm` to notify the `BSWM` about the switch request. To configure the `BswMMModeRequestPort`:

Step 3.1

In the **`BswMMModeRequestPort`** table, add an entry.

Step 3.2

In the `BswMRequestProcessing` parameter, select either `BSWM_IMMEDIATE` or `BSWM_DEFERRED`.

Step 3.3

In the `BswMMModeRequestSource` parameter, select `BswMDcmSessionModeRequest`.

Step 4

The `BswMMModeCondition` specifies the conditions under which the `BswMAction` is executed. To configure the `BswMMModeCondition`:

Step 4.1

In the **`BswMMModeCondition`** table, add an entry.

Step 4.2

In the `BswMConditionType` parameter, select `BSWM_EQUALS`.

Step 4.3

In the `BswMConditionMode` parameter, select your `BswMMModeRequestPort`.

Step 4.4

In the `BswMConditionValue` parameter, select `BswMBswMode`.

Step 4.5

In the `BswMBswRequestedMode` parameter, enter the name of your `CommunicationControl`.

**Step 5**

The BswMLogicalExpression allows to evaluate BswMModeConditions as part of a logical expression. To configure the BswMLogicalExpression:

Step 5.1

In the **BswMLogicalExpression** table, add an entry.

Step 5.2

Clear the `BswMLogicalOperator` parameter.

Step 5.3

In the **BswMArgumentRef** table, add an entry.

Step 5.4

In the `BswMArgument` parameter, select your BswMModeCondition.

Step 6

The BswMRule combines the BswMLogicalExpression and the BswMAction to define what action is executed under what conditions. To configure the BswMRule:

Step 6.1

In the **BswMRule** table, add an entry.

Step 6.2

In the `BswMRuleInitState` parameter, select `BSWM_FALSE`.

Step 6.3

In the `BswMRuleExpressionRef` parameter, select your BswMLogicalExpression.

Step 6.4

Select the `BswMRuleTrueActionList` parameter.

The configured behavior is executed when the `Dcm` requests a switch to the `BSWM` mode to the desired `CommunicationControl`, i.e. when the configured `BswMRule` becomes TRUE.

Step 6.5

In the `BswMRuleTrueActionList` parameter, select your `BswMActionList`.

4.4.3.4.8.5. Configuring a ModeRule to reset CommunicationControls

In this section, you configure a ModeRule which, upon becoming FALSE, resets all CommunicationControls that are in progress, i.e. it resets the `CommunicationControl` for `Dcm_CommunicationControl__<Network_>` ModeDeclarationGroupPrototypes that belong to all `ComM` channels which the `Dcm` is aware of and can control via the `CommunicationControl (0x28)` UDS service to the `DCM_ENABLE_RX_TX_NORM_NM` mode, if their current mode is different from `DCM_ENABLE_RX_TX_NORM_NM`. The logical state of this ModeRule is polled every `Dcm_MainFunction()` cycle.

Required feature: UDS service `CommunicationControl (0x28)`

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`



- Required non-diagnostic stack modules: ComM and Rte/SchM



Configuring the Dcm

Prerequisite:

- At least one `DcmModeRule` `ModeRule` is configured.
- The `ModeDeclaration` support is enabled for the `CommunicationControl` (0x28) UDS service. For more information, see [Section 4.4.3.4.8.3, “Configuring ModeDeclaration support for the CommunicationControl \(0x28\) UDS service”](#).

Step 1

In the `DcmDspComControlCommunicationReEnableModeRuleRef` parameter, reference one `DcmModeRule`. The `Dcm` monitors the logical value of this `ModeRule` every `Dcm_MainFunction()` cycle and resets `CommunicationControl` modes when this `ModeRule` becomes FALSE.



Configuring the ComM

Step 1

Configure the ComM module according to the ComM user guide.



Configuring the Rte/SchM

Step 1

Configure the Rte/SchM module for `ModeDeclaration` interaction for the `CommunicationControl` (0x28) UDS service handler according to [Section 4.4.3.4.8.3, “Configuring ModeDeclaration support for the CommunicationControl \(0x28\) UDS service”](#).

4.4.3.4.8.6. Configuring ComM channels for the CommunicationControl (0x28) service

In this section, you configure which ComM channels are controlled by requests to the internally-managed sub-functions of the `CommunicationControl` (0x28) UDS service.

**NOTE****Limitations of ComM channel configuration**

The configuration of considered ComM channels for the CommunicationControl (0x28) UDS service handler besides those which are referenced from the Dcm's reception channels is only possible if the CommunicationControl (0x28) service uses AUTOSAR 4.0-3 or higher ModeDeclaration based interaction. For more information on how to configure the CommunicationControl (0x28) UDS service handler to use AUTOSAR 4.0.3 or higher ModeDeclaration based interaction, see section [Section 4.4.3.4.8.3, “Configuring ModeDeclaration support for the CommunicationControl \(0x28\) UDS service”](#).

4.4.3.4.8.6.1. Configuring ComM channels for CommunicationControl (0x28) UDS service requests with a subnetNumber between 0x1 and 0xE

In this section, you configure which ComM channels are controlled by requests to the following internally-managed subfunctions of the CommunicationControl (0x28) UDS service that contains a subnetNumber between 0x1 and 0xE:

- ▶ enableRxAndTx (0x00)
- ▶ enableRxAndDisableTx (0x01)
- ▶ disableRxAndEnableTx (0x02)
- ▶ disableRxAndTx (0x03)

Required feature: UDS service CommunicationControl (0x28)

Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: ComM and Rte/SchM



Configuring the Dcm

Prerequisite:

- ModeDeclaration-interaction is configured for the CommunicationControl (0x28) UDS service. For more information, see [Section 4.4.3.4.8.3, “Configuring ModeDeclaration support for the CommunicationControl \(0x28\) UDS service”](#).

Step 1

Within the DcmDspComControlSpecificChannel table, configure one entry for each ComM channel for which the Dcm can set the CommunicationControl specifically, using a subnetNumber between 0x1 and 0xE in the CommunicationControl (0x28) UDS service request.



Step 2

In the `DcmDspSpecificComMChannelRef` parameter, reference the `ComM` channel for which the `Dcm` can set the `CommunicationControl`.

Step 3

In the `DcmDspSubnetNumber` parameter, configure the `subnetNumber` which will be used in the `CommunicationControl` (0x28) UDS service request to refer to this `ComM` channel.



Configuring the `ComM`

Step 1

Configure the `ComM` module according to the `ComM` user guide.



Configuring the `Rte/SchM`

Step 1

To configure the `Rte/SchM` for `ModeDeclaration`-interaction for the `CommunicationControl` (0x28) UDS service handler, see [Section 4.4.3.4.8.3, “Configuring ModeDeclaration support for the CommunicationControl \(0x28\) UDS service”](#).

4.4.3.4.8.6.2. Configuring `ComM` channels for `CommunicationControl` (0x28) UDS service requests with a `subnetNumber` of 0x00

In this section, you configure which `ComM` channels are controlled by requests to the following internally-managed subfunctions of the `CommunicationControl` (0x28) UDS service containing a `subnetNumber` of 0x00:

- ▶ `enableRxAndTx` (0x00)
- ▶ `enableRxAndDisableTx` (0x01)
- ▶ `disableRxAndEnableTx` (0x02)
- ▶ `disableRxAndTx` (0x03)

Required feature: UDS service `CommunicationControl` (0x28)

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: `ComM` and `Rte/SchM`



Configuring the Dcm

Prerequisite:

- ModeDeclaration-interaction is configured for the CommunicationControl (0x28) UDS service. For more information, see [Section 4.4.3.4.8.3, “Configuring ModeDeclaration support for the CommunicationControl \(0x28\) UDS service”](#).

Step 1

Within the DcmDspComControlAllChannel table, configure one entry for each ComM channel for which the Dcm can set the CommunicationControl globally, using a subnetNumber of 0x00 in the CommunicationControl (0x28) UDS service request.

Step 2

In the DcmDspComControlAllChannel parameter, reference the ComM channel for which the Dcm can set the CommunicationControl.



Configuring the ComM

Step 1

Configure the ComM module according to the ComM user guide.



Configuring the Rte/SchM

Step 1

Configure the Rte/SchM module for ModeDeclaration-interaction for the CommunicationControl (0x28) UDS service handler according to [Section 4.4.3.4.8.3, “Configuring ModeDeclaration support for the CommunicationControl \(0x28\) UDS service”](#).

4.4.3.4.8.6.3. Configuring ComM channels for CommunicationControl (0x28) UDS service requests with enhanced address information

In this section, you configure which ComM channels are controlled by requests to the internally-managed sub-functions:

- ▶ enableRxAndDisableTxWithEnhancedAddressInformation (0x04)
- ▶ enableRxAndTxWithEnhancedAddressInformation (0x05)

of the CommunicationControl (0x28) UDS service containing a subnetNumber of 0x00.

Required feature: UDS service CommunicationControl (0x28)



Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: ComM and Rte/SchM



Configuring the Dcm

Prerequisite:

- ModeDeclaration-interaction is configured for the CommunicationControl (0x28) UDS service. For more information, see [Section 4.4.3.4.8.3, “Configuring ModeDeclaration support for the CommunicationControl \(0x28\) UDS service”](#).

Step 1

Within the DcmDspComControlSubNode table, configure one entry for each ComM channel for which the Dcm can set the CommunicationControl using Enhanced Address Information via a CommunicationControl (0x28) UDS service request.

Step 2

In the DcmDspComControlSubNodeComMChannelRef parameter, reference the ComM channel for which the Dcm can set the CommunicationControl.

Step 3

In the DcmDspComControlSubNodeId parameter, configure the nodeIdentificationNumber. This number will be used in the CommunicationControl (0x28) UDS service request to refer to this ComM channel.



Configuring the ComM

Step 1

Configure the ComM module according to the ComM user guide.



Configuring the Rte/SchM

Step 1

To configure the Rte/SchM module for ModeDeclaration-interaction for the CommunicationControl (0x28) UDS service handler, see [Section 4.4.3.4.8.3, “Configuring ModeDeclaration support for the CommunicationControl \(0x28\) UDS service”](#).



4.4.3.4.9. Configuring the RoutineControl (0x31) UDS service

This chapter explains how to configure the RoutineControl (0x31) UDS service and the diagnostic routines to be used by this service.

4.4.3.4.9.1. Configuring the RoutineControl (0x31) UDS service handler

In this section, the RoutineControl (0x31) UDS service is configured.

Required feature: UDS service RoutineControl (0x31)

Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: PduR and EcuC



Configuring the Dcm

Prerequisite:

- At least one **Diagnostic Service Table** is configured.

Step 1

Add an entry in the **Diagnostic Services** table.

Step 2

In the DcmDsdSidTabServiceId parameter, enter 0x31. This is the service ID of the RoutineControl (0x31) UDS service.

Step 3

Configure the DcmDsdSidTabSubfuncAvail parameter to TRUE.

Step 4

Clear the DcmDsdSidTabFnc parameter.

Step 5

Configure any access conditions necessary for this diagnostic service as described in [Section 4.4.3.4.4.1, “Configuring the availability of diagnostic services in diagnostic sessions”](#), [Section 4.4.3.4.4.3, “Configuring security access verification for diagnostic services”](#) and [Section 4.4.3.4.4.5, “Configuring mode rule verification for diagnostic services”](#).

4.4.3.4.9.2. RoutineControl (0x31) UDS service subfunctions

Neither internally managed subfunctions nor user-implemented externally-managed subfunctions for the RoutineControl (0x31) UDS service can be explicitly configured under DcmDsdSubService.

**NOTE**

Subfunction availability is configured individually per diagnostic routine instead of service itself so implicitly all subfunctions are considered to be available for the `RoutineControl (0x31)` UDS service.

Subfunction `startRoutine` is implicitly enabled for all routines and can not be disabled.

Subfunctions `stopRoutine` and `requestRoutineResults` availability can be explicitly configured for each `DcmDspRoutine` by the user, see `DcmDspRequestResultsRoutineSupported` and `DcmDspStopRoutineSupported`.

4.4.3.4.9.3. Configuring diagnostic routines controlled by the `RoutineControl (0x31)` UDS service

4.4.3.4.9.3.1. Configuring a `DcmDspRoutine`



Configuring a `DcmDspRoutine`

Step 1

Add a `DcmDspRoutine` entry.

Step 2

Configure the `DcmDspRoutineIdentifier`.

Step 3

Configure the `DcmDspRoutineUsed` parameter which activates or deactivates the usage of the routine.

If `DcmDspRoutineUsed` is set to FALSE the respective routine is considered as not supported, i.e. the Dcm will respond with a negative response with the NRC `requestOutOfRange (0x31)` when a request for the respective routine is received.

Step 4

Configure `DcmDspRoutineUsePortASRVersion` to specify the AUTOSAR version compatibility for the routine interfaces.

- ▶ Default: the default version specified in `DcmDefaultASRServiceAPI` shall be used
- ▶ AUTOSAR_40: the routine interfaces shall be compliant to AUTOSAR 4.0.3
- ▶ AUTOSAR_42: the routine interfaces shall be compliant to AUTOSAR 4.2.1 and AUTOSAR 4.3

**NOTE**

If `DcmDspRoutineUsePortASRVersion` is disabled, then the default version specified in `DcmDefaultASRServiceAPI` shall be used.

**Step 5**

Configure `DcmDspRoutineFixedLength` to indicate if the routine has variable length signals.

NOTE

This parameter is used only in case the routine interfaces are compatible with AUTOSAR 4.0.3 (see **Step 3**). Otherwise the information regarding variable length signals is deduced from the presence of a `VARIABLE_LENGTH DcmDspRoutineSignalType` in the referenced `DcmDspRoutineInfo`.

**Step 6**

Configure `DcmDspRoutineUsePort` to indicate the type of the interface that shall be used to implement the `StartRoutine`, `StopRoutine` and `RequestResults` operations.

- ▶ TRUE: the DCM uses a port requiring a `PortInterface RoutineServices_{RoutineName}`. The R-Port is named `RoutineServices_{RoutineName}` where `{RoutineName}` is the name of the container `DcmDspRoutine`.
- ▶ FALSE: the DCM uses the functions configured in `DcmDspStartRoutineFnc`, `DcmDspStopRoutineFnc` or `DcmDspRequestResultsRoutineFnc`.

Step 7

If `DcmDspRoutineUsePort` is disabled, configure `DcmDspStartRoutineFnc` to define the name of the callout used for the `StartRoutine` sub-function.

Step 8

Configure `DcmDspRequestResultsRoutineSupported` to indicate whether the `RequestResults` sub-function of the `RoutineControl` (0x31) service is available for the diagnostic routine.

Step 9

If `DcmDspRoutineUsePort` is disabled and `DcmDspRequestResultsRoutineSupported` is enabled, configure `DcmDspRequestResultsRoutineFnc` to define the name of the callout used for the `RequestResults` sub-function.

Step 10

Configure `DcmDspStopRoutineSupported` to indicate whether the `StopRoutine` sub-function of the `RoutineControl` (0x31) service is available for the diagnostic routine.

Step 11

If `DcmDspRoutineUsePort` is disabled and `DcmDspStopRoutineSupported` is enabled, configure `DcmDspStopRoutineFnc` to define the name of the callout used for the `StopRoutine` sub-function.

Step 12

Configure `DcmDspRoutineInfoRef` to reference the `DcmDspRoutineInfo` container defining the signals and permissions of the diagnostic routine.



Step 13

If needed, configure `DcmDspRoutineTidRef` to reference the `DcmDspRequestControl` container defining the configuration of the Request control of on-board system, test or component service, OBD Service \$08.

4.4.3.4.9.3.2. Configuring a `DcmDspRoutineInfo`

`DcmDspRoutineInfo` contains the routine permissions and the routine signals configuration, which can be shared by many `DcmDspRoutines`.



Configuring a `DcmDspRoutineInfo`

Step 1

Configure the `DcmDspRoutineModeRuleRef` reference to the `DcmDspRoutineModeRule` which controls this routine. If no reference is configured, no check of the mode rule will be done.

Step 2

Configure the `DcmDspRoutineSecurityLevelRef` reference(s) to the `DcmDspSecurityRow` security levels allowed to control this routine. If no reference is configured, no check of security level will be done.

Step 3

Configure the `DcmDspRoutineSessionRef` reference(s) to the `DcmDspSessionRow` diagnostic sessions allowed to control this routine. If no reference is configured, no check of diagnostic session will be done.



Step 4

NOTE



- ▶ The input parameters for the `RequestResults` subservice shall be configured under `DcmDspRoutineRequestResIn`
- ▶ The output parameters for the `RequestResults` subservice shall be configured under `DcmDspRoutineRequestResOut`
- ▶ The input parameters for the `StopRoutine` subservice shall be configured under `DcmDspRoutineStopIn`
- ▶ The output parameters for the `StopRoutine` subservice shall be configured under `DcmDspRoutineStopOut`
- ▶ The input parameters for the `StartRoutine` subservice shall be configured under `DcmDspStartRoutineIn`
- ▶ The output parameters for the `StartRoutine` subservice shall be configured under `DcmDspStartRoutineOut`

All these containers can be configured in the same way by following the instructions in this step.

Configure for each routine signal:

- ▶ `DcmDspRoutineSignalLength` which represents:
 - ▶ the exact signal length in the `RoutineControl` request/response in bits if `DcmDspRoutineSignalType` is different than `VARIABLE_LENGTH`.
For array types signals, configure `DcmDspRoutineSignalLength` as an even multiple of the base type in bits.
 - ▶ the maximum signal length in the `RoutineControl` request/response in bits if `DcmDspRoutineSignalType` is different than `VARIABLE_LENGTH`. The exact signal length is returned by the application in the operation interface parameter `currentDataLength` (as specified in AUTOSAR 4.0.3) or `variableDataLength` (as specified in AUTOSAR 4.3).
- ▶ `DcmDspRoutineSignalPos` - signal position in the `routineControlOptionRecord/routineStatusRecord` in bits
- ▶ `DcmDspRoutineSignalType` - signal type that can be one of the following:
 - ▶ Primitive type:
 - ▶ `BOOLEAN`
 - ▶ `SINT16`
 - ▶ `SINT32`
 - ▶ `SINT8`



- ▶ ▶ `UINT16`
- ▶ ▶ `UINT32`
- ▶ ▶ `UINT8`
- ▶ **Array type:**
 - ▶ `SINT16_N`
 - ▶ `SINT32_N`
 - ▶ `SINT8_N`
 - ▶ `UINT16_N`
 - ▶ `UINT32_N`
 - ▶ `UINT8_N`
- ▶ **Variable length type:** `VARIABLE_LENGTH`

NOTE Only the last signal in the list can be configured with `VARIABLE_LENGTH`.



4.4.3.4.9.3.3. Configuring `DcmDspRoutineVariableLengthInBytes`

With `DcmDspRoutineVariableLengthInBytes`, you specify the measurement unit of `currentDataLength` (AUTOSAR 4.0.3) or `variableDataLength` (AUTOSAR 4.3). These parameters express the length of routine signals that have `DcmDspRoutineSignalType` set to `VARIABLE_LENGTH`. You have the following options:

- ▶ `TRUE`: The signal length is expressed in bytes.
- ▶ `FALSE`: The signal length is expressed in bits.

4.4.3.4.9.4. Configuring application-defined interfaces for unsupported `DcmDspRoutineIdentifiers`



Configuring application-defined interfaces for subfunctions `startRoutine`, `stopRoutine`, and `requestRoutineResults`

Step 1

Enable the configuration parameter `DcmDspRoutineEnableDefaultInterfaces`.

Step 2

In the application, define the following interface functions:



- ▶ `Std_ReturnType Dcm_DefaultRoutineStart(Dcm_OpStatusType OpStatus, Dcm_RoutineIDType RoutineID, uint8 *routineControlOptionRecord, uint32 routineControlOptionRecordLength, uint8 *routineInfoAndStatusRecord, uint32 *routineInfoAndStatusRecordLength, Dcm_NegativeResponseCodeType *Nrc)`

When a request is performed for the UDS service RoutineControl (0x31) with an unsupported DcmDspRoutineIdentifier with subfunction startRoutine, the Dcm calls Dcm_DefaultRoutineStart instead of transmitting a negative response.

- ▶ `Std_ReturnType Dcm_DefaultRoutineStop(Dcm_OpStatusType OpStatus, Dcm_RoutineIDType RoutineID, uint8 *routineControlOptionRecord, uint32 routineControlOptionRecordLength, uint8 *routineInfoAndStatusRecord, uint32 *routineInfoAndStatusRecordLength, Dcm_NegativeResponseCodeType *Nrc)`

When a request is performed for the UDS service RoutineControl (0x31) with an unsupported DcmDspRoutineIdentifier with subfunction stopRoutine, the Dcm calls Dcm_DefaultRoutineStop instead of transmitting a negative response.

- ▶ `Std_ReturnType Dcm_DefaultRoutineRequestRoutineResults(Dcm_OpStatusType OpStatus, Dcm_RoutineIDType RoutineID, uint8 *routineControlOptionRecord, uint32 routineControlOptionRecordLength, uint8 *routineInfoAndStatusRecord, uint32 *routineInfoAndStatusRecordLength, Dcm_NegativeResponseCodeType *Nrc)`

When a request is performed for the UDS service RoutineControl (0x31) with an unsupported DcmDspRoutineIdentifier with subfunction requestRoutineResults, the Dcm calls Dcm_DefaultRoutineRequestRoutineResults instead of transmitting a negative response.

Parameter definitions:

- ▶ `OpStatus`
 - ▶ `DCM_INITIAL`
 - ▶ `DCM_PENDING`
 - ▶ `DCM_CANCEL`
- ▶ `RoutineID: 2-byte DcmDspRoutineIdentifier`
- ▶ `routineControlOptionRecord: pointer to the location of the first byte of the routineControlOptionRecord from the request message`
- ▶ `InputLength: length of the routineControlOptionRecord from the request message`
- ▶ `routineInfoAndStatusRecord: pointer to the location in the output buffer of the first byte of the routineInfo and routineStatusRecord data`
- ▶ `Length: pointer to which the application shall write the amount of data, in bytes, that was written to the routineInfoAndStatusRecord pointer, comprising the routineInfo and routineStatusRecord`



- ▶ `Nrc`: pointer to which the application shall write the NRC to be returned. Valid only if the return value is `E_NOT_OK`.

NOTE

`DcmDspRoutineIdentifiers` that correspond to configured OBD elements are considered supported.



4.4.3.4.9.5. Configuring the application-defined interface for inserting the RoutineInfo byte in the diagnostic response as defined in ISO 14229-1/2013



Configuring the interface for inserting the RoutineInfo byte in the diagnostic response

Step 1

Enable the configuration parameter `DcmDspRoutineEnableRoutineInfoByte`.

Step 2

In the application, define the following interface function:

```
boolean Dcm_AddRoutineInfoByte( Dcm_RoutineIDType RoutineID, uint8 * RoutineInfoByte );
```

The application writes the RoutineInfo at the address specified by the `RoutineInfoByte` parameter and the Dcm inserts it in the positive response of the UDS service `RoutineControl` (0x31) after the routineIdentifier.

NOTE

`DcmDspRoutineIdentifiers` that correspond to configured OBD elements are considered supported.



4.4.3.4.10. Configuring the Request Vehicle Information (0x09) OBD service

This chapter explains how to configure the Request Vehicle Information (0x09) OBD service and the diagnostic vehicle information (Infotype) to be used by this service. For background information, see [Section 4.4.2.5.13, “OBD service Request Vehicle Information \(0x09\)”](#).

4.4.3.4.10.1. Configuring the Request Vehicle Information (0x09) OBD service handler

In this section, the Request Vehicle Information (0x09) is configured.



Required feature: OBD service Request Vehicle Information (0x09)

Required modules:

- ▶ Required diagnostic stack modules: Dcm
- ▶ Required non-diagnostic stack modules: PduR and EcuC



Configuring the Dcm

Prerequisite:

- At least one **Diagnostic Service Table** is configured.

Step 1

Add an entry in the **Diagnostic Services** table.

Step 2

In the DcmDsdSidTabServiceId parameter, enter 0x09. This is the service ID of the Request Vehicle Information (0x09) OBD service.

Step 3

Configure any access conditions necessary for this diagnostic service as described in the sections:

- ▶ [Section 4.4.3.4.4.1, “Configuring the availability of diagnostic services in diagnostic sessions”](#)
- ▶ [Section 4.4.3.4.4.3, “Configuring security access verification for diagnostic services”](#)
- ▶ [Section 4.4.3.4.4.5, “Configuring mode rule verification for diagnostic services”](#)

4.4.3.4.10.2. Configuring DcmDspVehInfoASRVersion of Request Vehicle Information (0x09) OBD service



Configuring DcmDspVehInfoASRVersion

Step 1

Configure the DcmDspVehInfoASRVersion, which defines the AUTOSAR version compatibility for the Request Vehicle Information (0x09) OBD service.

- ▶ AUTOSAR_403: The implementation and the service interface are AUTOSAR 4.0.3 compliant.
- ▶ AUTOSAR_440: The implementation and the service interface



4.4.3.4.10.3. Configuring diagnostic vehicle information (`Infotype`) controlled by the Request Vehicle Information (0x09) OBD service

4.4.3.4.10.3.1. Configuring a `DcmDspVehInfo`



Configuring a `DcmDspVehInfo`

Step 1

Add a `DcmDspVehInfo` entry.

Step 2

Configure the `DcmDspVehInfoInfoType` entry.

NOTE



The `DcmDspVehInfoInfoType` must be different from an *availability InfoType*.

Type

`DcmDspVehInfoInfoType` must not be equal to an *availability InfoType*. Thus, do not use 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, or 0xE0 for `DcmDspVehInfoInfoType`.

Step 3

Configure the `DcmDspVehInfoUsed` that activates or deactivates the usage of the `Infotype`.

If `DcmDspVehInfoUsed` is set to FALSE, the respective `Infotype` is considered as not supported. That means the `Dcm` does not send a response when a request for the respective `Infotype` is received.

Step 4

Configure the `DcmDspVehInfoNODIProvResp` that indicates the side responsible to fill the number of data items (NODI): `Dcm` or the provider of the `Infotype` data.

If `DcmDspVehInfoNODIProvResp` is set to FALSE, the `Dcm` computes the data byte `NofDataItems` in the diagnostic response, which defines the number of `DataItems` included in one `InfoType`.

If `DcmDspVehInfoNODIProvResp` is set to TRUE, the `Dcm` takes over the value returned by the provider and reports it as `NofDataItems` in the diagnostic response.

4.4.3.4.10.3.1.1. Configuring a `DcmDspVehInfoData`



Configuring a `DcmDspVehInfoData`

Step 1

Add a `DcmDspVehInfoData` entry.

**Step 2**

Configure the `DcmDspVehInfoDataOrder` entry.

NOTE**Unique `DcmDspVehInfoDataOrder` for each `DcmDspVehInfo`**

The configuration of the data order needs to be unique per `DcmDspVehInfo`.

**Step 3**

Configure the `DcmDspVehInfoDataUsePort` to indicate the type of the interface that shall be used to implement the `GetInfotypeValueData` operation, used to retrieve data from the `DcmDspVehInfoData`.

- ▶ TRUE: the Dcm uses a port requiring a `PortInterface_InfotypeServices_{VehInfoData}`. The R-Port is named `InfotypeServices_{VehInfoData}` where `{VehInfoData}` is the name of the container `DcmDspVehInfoData`.
- ▶ FALSE: the Dcm uses the function configured in `DcmDspVehInfoDataReadFnc`.

Step 4

If `DcmDspVehInfoDataUsePort` is disabled, configure `DcmDspVehInfoDataReadFnc` to define the name of the callout used for the `GetInfotypeValueData` operation.

Step 5

Configure the `DcmDspVehInfoDataSize` entry, which defines the size of the `Infotype` data item in bytes.

NOTE**`DcmDspVehInfoDataSize` must not be zero**

A `DcmDspVehInfoDataSize` with the value 0 is invalid.



4.4.3.5. Configuring the diagnostic service request handling

This chapter describes how to configure the Dcm module to allow it to receive a diagnostic service request from a diagnostic tool, process the request, and transmit a response.

The configuration of the service request handling consists of the following steps:

- ▶ [Section 4.4.3.5.1, “Configuring the diagnostic buffers”](#)
- ▶ [Section 4.4.3.5.2, “Configuring the diagnostic data flow”](#)
- ▶ [Section 4.4.3.5.3, “Configuring the communication mode handling”](#)
- ▶ [Section 4.4.3.5.4, “Enabling diagnostic services within a diagnostic protocol”](#)
- ▶ [Section 4.4.3.5.5, “Enabling diagnostic requests from the application”](#)



4.4.3.5.1. Configuring the diagnostic buffers

In this section, you configure a diagnostic buffer in which received service requests or transmitted responses are stored. You can also store received service requests and transmitted responses in the same buffer.

For more information on diagnostic buffers, see [Section 4.4.2.6.2, “Dcm buffers”](#)

Required feature: Buffer configuration with support of shared buffers

Required modules

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Step 1

To add a diagnostic buffer, add an entry to the **Diagnostic Buffers** table.

Step 2

In the `DcmDslBufferSize` parameter, enter at least 8 bytes as buffer size.

4.4.3.5.2. Configuring the diagnostic data flow

In this section, you configure PDUs, the PDU routing, and the communication channels of the `Dcm`.

For more information on the diagnostic data flow, see [Section 4.4.2.6.3, “Communication with other basic software modules”](#) or [Section 4.4.2.6.5, “EcuC configuration”](#).

Required feature: PduR interface according to AUTOSAR 4.x except generic connection handling

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: `PduR` and `EcuC`



Configuring the EcuC

Step 1

In the **Pdu** table, add 4 PDU entries.

Step 2

Rename the PDU entries as follows:

- ▶ `Pdu_Source_IfTp`



- ▶ Pdu_Target_Dcm
- ▶ Pdu_Source_Dcm
- ▶ Pdu_Target_IfTp

IfTp is used as a placeholder for the interface module that sends requests to the `Dcm` and receives responses from the `Dcm`.

Step 3

In the `PduLength` parameter of each PDU entry, enter the memory space in bytes.



Configuring the Dcm

Prerequisite:

- At least one main connection is configured for the diagnostic protocol.
- PDUs are configured in the `EcuC` module.

Step 1

To add a reception channel to a main connection, add an entry to the **Receive Channels** table.

In the `DcmDslProtocolRxTesterSourceAddr` parameter, enter the source address of the tester that is associated with the main connection.

Step 2

In the `DcmDslProtocolRxPduRef` parameter, reference the `Pdu_Target_Dcm` PDU. This PDU is used by the `Dcm` to receive requests.

Step 3

In the `DcmDslProtocolRxAddrType` parameter, select if the reception channel should be physical or functional.

Physical RxPdulds represent unicast RxPdulds. Functional RxPdulds represent multicast RxPdulds.

NOTE



Negative response suppression according to ISO 14229

Concurrent `TesterPresent` (0x3E 0x80) requests are also received on functional RxPdulds. Certain NRCs are suppressed if the request which results in them was received on a functional RxPduld. For detailed information on which negative response codes (NRCs) are suppressed, see [Section 4.4.2.5.4, “Negative and positive response suppression”](#).

Step 4

In the `DcmDslProtocolRxBufferID` parameter, reference the diagnostic buffer that is used by the `Dcm` to receive requests.

This Rx buffer must have enough storage to accept all requests that are expected on this RxPduld.



Step 5

In the `DcmDslTxConfirmationPduId` parameter, enter the transmit confirmation Pduld.

Step 6

In the `DcmDslProtocolTxPduRef` parameter, reference the `Pdu_Source_Dcm` PDU. This PDU is used by the `Dcm` to transmit responses.



Configuring the PduR

Step 1

In the **PduRRoutingTables**, add a new entry.

Step 2

In the **PduRRoutingTable** table, add 2 entries.

Every routing direction is configured as one routing table. Therefore, if you want to enable the `Dcm` to receive requests and transmit responses, you must add two routing tables.

Step 3

To enable the `Dcm` to receive service requests:

Step 3.1

In the **PduRRoutingPath** table of the first routing table, add an entry.

Step 3.2

In the `PduRSourcePduHandleId` parameter, enter the same value that is specified in the `DcmDslProtocolRxPduId` parameter of the `Dcm` reception channel that you want to use.

Step 3.3

In the `PduRSrcPduRef` parameter, reference the `Pdu_Source_IfTp` PDU.

Step 3.4

In the **PduRDestPdu** table, add an entry.

This `PduRDestPdu` entry defines the destination of the routed data. The `PduR` is capable of 1:n routing, i.e. routing one source PDU to more than one destination PDU. Because of this, the **PduRDestPdu** table can contain more than one destination to which the PDU can be routed.

Step 3.5

In the `PduRDestPduHandleId` parameter, enter the same value that is specified in the `DcmDslTxConfirmationPduId` parameter of the main connection that you want to use.

Step 3.6

In the **PduRDestPduRef** parameter, reference the destination `Pdu_Target_Dcm` PDU.

Step 4

To enable the `Dcm` to transmit responses:

Step 4.1

In the **PduRRoutingPath** table of the second routing table, add an entry.



Step 4.2

In the `PduRSourcePduHandleId` parameter, enter the same value that is specified in the `DcmDslProtocolRxPduId` parameter of the `Dcm` reception channel that you want to use.

Step 4.3

In the `PduRSrcPduRef` parameter, reference the `Pdu_Source_Dcm` PDU.

Step 4.4

In the **PduRDestPdu** table, add an entry.

Step 4.5

In the `PduRDestPduHandleId` parameter, enter the same value that is specified in the `DcmDslTxConfirmationPduId` parameter of the main connection that you want to use.

Step 4.6

In the `PduRDestPduRef` parameter, reference the destination `Pdu_Target_IftTp` PDU.

4.4.3.5.3. Configuring the communication mode handling

In this section, you configure the communication mode handling of the `Dcm`. The communication mode is provided by the `ComM` module.

For more information on the communication modes, see [Section 4.4.2.6.6, “Communication modes”](#).

Required features:

- ▶ Handling of communication requirements (full/silent/no communication)
- ▶ Indicating of active/inactive diagnostic

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: `ComM`



Configuring the ComM

Step 1

In the **ComMConfigSet** table, add an entry.

Step 2

In the **ComMChannel** table, add an entry.

Step 3

In the `ComMBusType` parameter, select your bus type.

Step 4

In the `ComMChannelId` parameter, enter a channel ID.



Configuring the Dcm

Step 1

In the `DcmDslProtocolRxComMChannelRef` parameter, reference the network channel of the `ComM` module.

Step 2

In the `DcmDslProtocolRxChannelId` parameter, enter the channel ID of the network channel that you referenced in the `DcmDslProtocolRxComMChannelRef` parameter.

The reference and the channel ID have to be configured both for backward compatibility reasons.

Step 3

If you want to make the `Dcm_SetActiveDiagnostic()` API and the `<module>_SetActiveDiagnostic()` operation of the `DCMServices ClientServerInterface` available to the application, enable the `DcmDslEnableSetActiveDiagnosis` parameter.

This API and operation allows the application to allow or deny the `Dcm` to request FullCommunication-Mode from the `ComM`.

4.4.3.5.4. Enabling diagnostic services within a diagnostic protocol

In this section, you configure a diagnostic service table to be available within a diagnostic protocol. The diagnostic service table contains diagnostic services and subfunctions. This defines which services can be requested on the RxPduds that are configured within the main connection of the protocol.

For more information on diagnostic protocols, see [Section 4.4.2.1, “Diagnostic protocol”](#).

Required feature: Service table per protocol

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: none



Configuring the Dcm

Prerequisite:

- At least one **Diagnostic Service Table** with UDS or OBD services is configured. For more information, see [Section 4.4.3.4, “Configuring diagnostic services”](#).

Step 1

In the `DcmDslProtocolSIDTable` parameter, reference a diagnostic service table.



4.4.3.5.5. Enabling diagnostic requests from the application

In this section, the configuration of the application diagnostic request feature in the `Dcm` is explained.

For background information on application diagnostic requests, see [Section 4.4.2.6.4, “Application diagnostic requests”](#).

Required feature: Application diagnostic requests

Required modules:

- ▶ Required diagnostic stack modules: `Dcm`
- ▶ Required non-diagnostic stack modules: none



Configuring the `Dcm`

Step 1

Enable the `DcmDslApplicationDiagnosticRequests` container.

Step 2

In `DcmDslApplicationDiagnosticRequestBufferSize`, specify the size limit for the planned application request.

Step 3

Configure the provided interface `Dcm.InjectDiagnosticRequest` to execute the application request.

- ▶ Interface name: `Dcm.InjectDiagnosticRequest`
- ▶ Entry parameters:

Parameter	Type	Description
<code>RxPduID</code>	<code>PduldType</code>	Reception PDU identifier on which the application request is to be executed
<code>requestData</code>	Pointer to constant <code>uint8</code>	Pointer containing the application request data
<code>requestLength</code>	<code>PduLengthType</code>	Length of the application request

- ▶ Return type: `BufReq.ReturnType`
- ▶ Possible return values:

Return response	Description
<code>BUFREQ_E_OK</code>	The request injection has been accepted and will be processed
<code>BUFREQ_E_BUSY</code>	Another application injected request is already ongoing
<code>BUFREQ_E_OVFL</code>	The request is too long and cannot be processed



Return response	Description
BUFREQ_E_NOT_OK	The provided RxPduID is invalid

4.4.3.6. Configuring BSW distribution

This chapter describes how to configure the `Dcm` module to allow `ComM` to be mapped to different partitions (cores). For more information on the `Dcm` BSW distribution, see [Section 4.4.2.7, “BSW distribution”](#).



Configuring the Dcm

Prerequisite:

- The `ComM` module has BSW distribution support enabled.
- In the `Rte`, the following is configured:
 - ▶ The module instances, i.e. the `BswImplementation`s, are properly mapped to the cores via `OsAppllications`.
 - ▶ Events are properly mapped to tasks.

For more information on event mapping, see [Section 4.4.4.4, “BSW distribution”](#) in the integration notes.

Step 1

To enable the `Dcm` BSW distribution support, enable the configuration parameter [`DcmComMMultiCoreSupport`](#).

Step 2

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

4.4.4. Dcm integration notes

This chapter provides `Dcm`-specific integration information.



4.4.4.1. Generic information

- ▶ `DcmTimStrP2ServerAdjust` has been introduced to guarantee that the response is on the bus before reaching P2. This guarantees that `Dcm` is trying to send the response in time only. It does not guarantee that the response is on the bus in time, since this depends on other modules as well as on the system load.
- ▶ Timers used in `Dcm` are based on the `Dcm_MainFunction()`. This limits the timer resolution to `Dcm-TaskTime`.
- ▶ Service execution is done completely in the `Dcm_MainFunction()` context. Therefore it is necessary to ensure that a service execution can be finished within a `Dcm_MainFunction()` cycle.
- ▶ If paged buffering is not activated for `ReadDTCInformation` (UDS 0x19) a buffer needs to be configured which is as large as the maximum size of all DTCs to read at one time.
- ▶ When ROE is used in conjunction with preconfigured and/or persistently started events it is inalienable to start the `Dcm` module after the application. This is necessary to allow `Dcm` to start events during the initialization phase.
- ▶ ROE defined by ISO knows the `persistence bit`. In `Dcm` there are two different kinds of persistence available:
 - ▶ A SETUP of a ROE request can be stored persistently and CLEARED persistently by setting the `persistence bit` in the request.
 - ▶ The state (STARTED / STOPPED) of an EVENT can be stored persistently by setting the `persistence bit` in the request.
- ▶ `Dcm` can handle internal and external service handlers as well as internal and external sub-service handlers. This can be configured individually for each service.
 - ▶ A service is handled internally. No external sub-service handlers are configured.
 - ▶ A service is handled internally. External sub-service handlers are configured. The external handlers are being called by invoking the API `<Module>_<DiagnosticService>_<SubService>()`.
 - ▶ A service is handled externally. External service handler is configured. The external handler is being called by invoking the API `<Module>_<DiagnosticService>()`. Sub-service handling has to be provided by the external service handler.
- ▶ Task configuration and mapping requirements for tasks used by the `Dcm` module

Mapping of the task responsible for asynchronous service handling

If asynchronous service handling is enabled for any of the configured service handlers, the `Dcm` generates in its Basic Software Module Description a `BswInternalTriggerOccuredEvent` named `BswInternalTriggerOccuredEvent_Async`. Asynchronous service handling is enabled if the configuration parameter `DcmAsyncServiceExecution` is configured to true for any of the entries configured in the **Diagnostic Services** table. This `BswInternalTriggerOccuredEvent` must be mapped to a task that has a lower priority than the task to which the `BswTimingEvent TimingEvent_MainFunction` of the `Dcm` is mapped. This is because the asynchronous service handler has to run in a task of a lower priority



than the task that runs the `Dcm_MainFunction()` which sets the `BswInternalTriggerOccuredEvent`. Thus the service handler is allowed to run in the low priority task in a decoupled fashion.

Besides the restriction in priority stated above, an additional restriction applies to the task to which the `BswInternalTriggerOccuredEvent` is mapped and therefore which runs the asynchronous diagnostic service handlers. This task must either be one of the following:

- ▶ An EXTENDED task.
- ▶ A BASIC task. However, the maximum number of activations configured for this task with configuration parameter `OsTaskActivation` must be at least 2.

The rationale for this restriction is that the asynchronous service processor task may continue to run for a short time after the service processing is completed and the response to the service request is sent to the tester. This behavior is the result of this task being preempted before it has the chance to terminate. If this behavior happens, the tester can further request another service from the `Dcm`. If this second service is also configured to be executed asynchronously, it leads to an activation of the asynchronous service processor task before it has terminated. This further leads to an error if the maximum number of activations is 1.

4.4.4.2. Protocol preemption

The `Dcm` module is able to handle multiple protocols. The maximum number of external requests which can be executed in parallel is currently 1. In parallel to a protocol that executes an external request, other protocols which execute periodic DID readouts or on-event request processing (ROE) can also run. However, this behavior is only possible if the transmissions resulting from the processing of these protocols are TYPE2 transmissions. This behavior has to be configured.

- ▶ The parameter `DcmDslProtocolPreemptTimeout` cannot be set 0 and is restricted to a minimum of one main function period.

Reason: Although the protocol preemption process can finish immediately there are numerous situations where the system needs to execute other tasks like the asynchronous server handler or `Rte` callbacks from service handlers or the protocol preemption itself. Since protocol preemption is initially processed within interrupt context the `Dcm` needs to have its main function executed at least once in such a situation.

- ▶ The container parameter `DcmDslCallbackDCMRequestService` configuring the respective `Rte` interface is now supported and cannot contain configuration items anymore if the `Rte` is disabled.

When a protocol preemption is in progress the `Dcm` will immediately return as configured via parameter `DcmDslDiagRespOnSecondDeclinedRequest` if another request for again another protocol is received regardless of the priority of the request. While this situation is considered as being very rare, it increases the responsiveness of the `Dcm` in such situations and reduces the need for additional resources without being able to predict if these preemption requests will ever be successful.

When a protocol preemption has been triggered and is processed but times out the preemption process continues to run. After the protocol has been stopped successfully no new protocol is started unless the request which timed out or a new request is received again.



4.4.4.3. Endianness interpretation

For backward compatibility, endianness interpretation supports any signal start bit position and any signal bit size. When configuring signals and DIDs, the signal size and signal start position must always be given in bits.

- ▶ The parameter [DcmDspDataSize](#) defines the signal size. The parameter [DcmDspDataByteSize](#) is not supported.

Reason: backward compatibility to bit type signals

- ▶ The parameter [DcmDspDidDataPos](#) defines the signal start position. The parameter [DcmDspDidByteOffset](#) is not supported.

Reason: backward compatibility to bit type signals

4.4.4.4. BSW distribution

The parameter [DcmComMMultiCoreSupport](#) enables the communication between the [Dcm](#) and the [ComM](#) when both modules are mapped to different partitions (cores).

NOTE
Mapping of Rte/BSW events to tasks


If the parameter [DcmComMMultiCoreSupport](#) is set to TRUE, the Rte/BSW event [Int-TriggerOccurredEvent_HighestPrioFunction](#) of the Rte configuration must be mapped to the task with the highest priority.

4.5. Dem module user guide

4.5.1. Overview

This chapter provides you with [Dem](#)-specific information:

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

4.5.2. Background information

The Diagnostic Event Manager ([Dem](#)) processes and stores the events (errors) that are detected by diagnostic monitors in both Software Components (SWCs) and Basic software (BSW) modules.



The `Dem` has the following main tasks:

- ▶ management of UDS status byte
- ▶ management of event memories
- ▶ management of non-volatile storage

The `Dem` enables the `Dcm` to access the event-related data via the `Dem-Dcm` API, e.g. to read or clear stored DTC information from the event memory.

As a service component, the `Dem` offers interfaces to the application layer via the `Rte` so that e.g. detected errors can be reported from the application.

4.5.2.1. Diagnostic events

Diagnostic events are reported from basic software modules or software components. The `Dem` processes these diagnostic events. The events are reported as failed, passed, pre-failed or pre-passed. Each diagnostic event has an event-ID that is associated with it. Furthermore, the `Dem` supports several attributes for each event such as:

- ▶ Diagnostic Trouble Code (DTC) in various formats used by different protocols
- ▶ several, partially optional, cycles such as operation cycle, failure cycle or aging cycle
- ▶ content, format, and destination of event-related data

4.5.2.2. Event processing

The `Dem` event processing is initiated by:

1. API [`Dem_SetEventStatus\(event, pre-passed/pre-failed/passed/failed\)`](#) from application
2. API [`Dem_ReportErrorStatus\(event, pre-passed/pre-failed/passed/failed\)`](#) from BSW
3. [`Dem_MainFunction\(\)`](#) if time- or frequency-debouncing qualifies an event

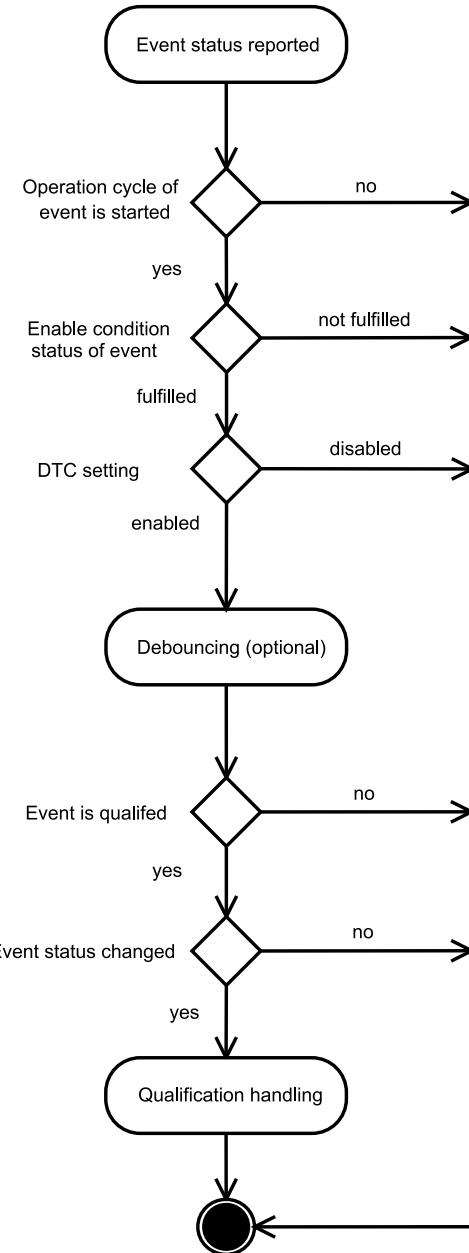


Figure 4.10. Event processing sequence

The event qualification comprises:

- ▶ update of the UDS status bits
- ▶ optional event memory entry storage

For BSW events, both operations are performed asynchronously in the [Dem_MainFunction\(\)](#).



For SWC events, the UDS status bit update of specific bits is always performed synchronously (refer to [Dem036]). All other operations can be configured as required. The switch between asynchronous and synchronous operations is configured with the [DemFreezeFrameCapture](#) / [DemExtendedDataCapture](#) parameters.

4.5.2.3. Event-related data (event memory)

The `Dem` is able to store event-related data if an event is reported as FAILED. The event-related data is configurable per event, thus different events are able to store different event-related data. Also the data elements are configurable. There are two types of event-related data that you can store with an event:

- ▶ Freeze Frame Data (freeze frame record or `DTCsnapshot` data)
- ▶ DTC Extended Data (extended data record)

You can configure every event to either store none, one, or both types of event-related data.

The point in time when the event-related data is initially captured and when it is updated depends on the event status. You can configure it via the configuration parameter [DemFreezeFrameCapture](#) for freeze frame records and [DemExtendedDataCapture](#) for extended data records.

4.5.2.3.1. Memory types

For every event in the `Dem`, you must configure an event destination via the parameter [DemEventDestination](#). The event destination defines in which memory the event and any configured event-related data are stored. This storage is also made non-volatile at least during shutdown if `NvM` is used for the non-volatile storage (see [Section 4.5.3.3, “NVRAM Manager \(NvM\)”](#) for more details).

With configuration parameter [DemImmediateNvStorage](#), you can configure the immediate storage to the non-volatile memory. If immediate non-volatile storage is enabled for a specific DTC, the `Dem` triggers the non-volatile storage of the respective event memory entry immediately after the entry creation, after each entry update, and after entry deletion (see [Section 4.5.2.3, “Event-related data \(event memory\)”](#)).

For ECUs without regular shutdown, the DTC status needs to be handled as volatile information. The `Dem` derives only the following status bits from the existence or non-existence of an event memory entry:

- ▶ TNCSLC (Test Not Completed Since Last Clear)
- ▶ CDTC (Confirmed DTC)
- ▶ TFSLC (Test Failed Since Last Clear)

Therefore, the monitors need to redetermine and report the current event status during each ECU start-up.

With configuration parameter [DemIntermediateNvStorageOfDTCStatus](#), you can configure the intermediate storage of the DTC status to the non-volatile memory. If enabled, the `Dem` triggers the non-volatile storage



of the complete DTC status whenever the event-related data is stored, in addition to the regular storage at shutdown. The `Dem` processes the non-volatile storage asynchronously and stores the latest DTC status at the time of storage.

NOTE**Potential inconsistencies**

If intermediate DTC storage is enabled, power failures and hard resets can lead to inconsistencies between the DTC status of different events and inconsistencies between the DTC status and the DTC occurrence time.

The `Dem` supports the following memory types:

- ▶ Primary memory
- ▶ Secondary memory
- ▶ Mirror memory
- ▶ Permanent memory (additional, not explicitly configurable)

NOTE**Permanent memory**

The permanent memory is not explicitly configurable and is used for emission-related events (OBD events) only. Emission-related events are automatically assigned to the permanent event memory if they reference the warning indicator MIL (malfunction indicator lamp). The event-related data of an emission-related event is stored in the primary event memory.

The permanent event memory is available only with EB tresos AutoCore Generic 8 Diagnostic Stack OBD.

NOTE**Accessing secondary memory**

The `Dem` APIs used by `Dcm` are implemented according to AUTOSAR 4.3.1, which does not support secondary memory. In order to access the data from secondary memory, use the parameter `DemUserDefMemoryId` to map the secondary memory to the desired user-defined memory ID. The data stored in the secondary memory can then be accessed using this specific user-defined memory ID.

4.5.2.3.2. Event-related data types

Extended data records:

Extended data records are linked to UDS service `ReadDTCInformation`, sub-function `reportDTCExtendedDataRecordByDTCNumber` (0x19-0x06).

Extended data consists of one or more extended data records. An extended data record consists of one or more [data elements \(DE\)](#), e.g. application-specific information, that is assigned to an event. If an error re-occurs,



you can update the extended data records with the latest data. You can configure the trigger for updating the record via [DemExtendedDataRecordUpdate](#). This is shown in [Figure 4.11, “Extended data record”](#). Thus, an extended data record always displays the latest or the initial state of the data associated with this event.

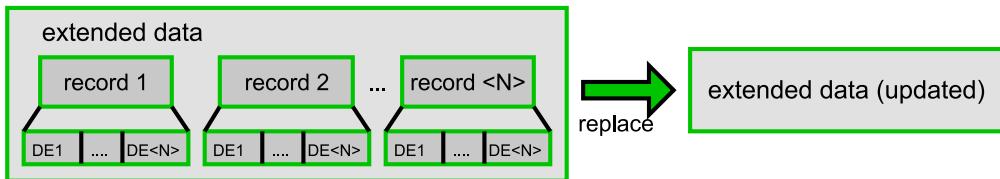


Figure 4.11. Extended data record

You can map various internal data elements to an extended data record with parameter [DemInternal-DataElement](#). You can configure some data elements to be always available in the extended data record, even if there is no event memory entry, with parameter [DemExtendedDataRecordAlwaysAvailable](#).

Freeze frame:

Freeze frame records are linked to UDS service `ReadDTCInformation`, sub-function `reportDTCSnapshotRecordByDTCNumber (0x19-0x04)`.

A freeze frame record consists of one or more data-IDs. A data-ID consists of one or more [data elements \(DE\)](#), e.g. environmental data, like time stamp or mileage. If an error re-occurs, a new record is appended to the event. The EB tresos AutoCore `Dem` stores the first N-1 records, while the last record is replaced if the maximum number of record slots for the event is reached. For details, see [DemMaxNumberFreezeFrameRecords](#). Thus freeze frame records store a history of the data-IDs associated with this event as shown in [Figure 4.12, “Freeze frame”](#).

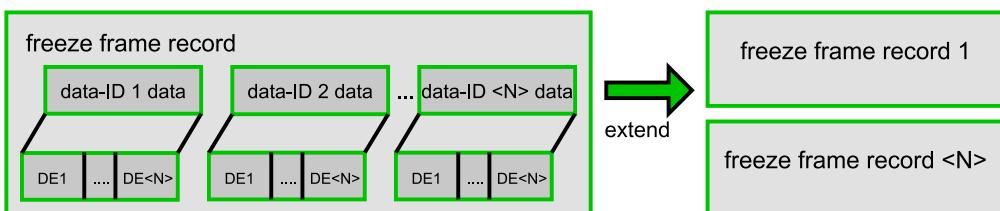


Figure 4.12. Freeze frame

Common freeze frame:

The `Dem` provides the ability to store common freeze frame data. You can configure the common freeze frame globally for the `Dem` via the configuration parameter [DemCommonFFDataClassRef](#). If this is done, the common freeze frame data is stored in addition to the regular freeze frame data of every event. It is also possible to only configure the common freeze frame without configuring any regular freeze frame record for any event. In that case, every event stores only the common freeze frame data. This reduces the configuration effort if a certain number of events shall store the same freeze frame data.



NOTE**Storage of freeze frames**

If the common freeze frame is enabled and configured, every event stores at least the common freeze frame. Thus, no event is stored without freeze frame data, even if no regular freeze frame is configured.

Pre-storage of freeze frame data:

The `Dem` provides the ability to pre-store freeze frame data. You can use this for events with highly volatile freeze frame data. If freeze frame pre-storage is enabled for a dedicated event, the `Dem` pre-stores the freeze frame data with the first FAILED or PREFAILED report of an event, even if the event is not yet debounced or qualified.

WARNING**Pre-storage of freeze frames**

A pre-stored freeze frame of an event might already be released before the main function is able to persistently store the data. This behavior occurs if a PASSED result is reported for an event while the `Dem` event queue still contains unprocessed FAILED reports.

OBD freeze frame:

The OBD freeze frame is a special kind of freeze frame that stores the freeze frame record for OBD-relevant events only. An OBD-relevant event is an event with an OBD DTC either configured or calibrated. In contrast to the regular freeze frame record, the OBD freeze frame does not consist of data-IDs (DIDs). The OBD freeze frame consists of parameter-IDs (PIDs), which again consist of configurable [data elements \(DE\)](#).

The `Dem` provides two variants for OBD freeze frame processing: single and multiple freeze frames. You select the variant with configuration parameter [`DemFreezeFrameCapture`](#). If you select `DEM_TRIGGER_ON_FDC_THRESHOLD`, a single OBD freeze frame is provided for all OBD-relevant events. For a selection other than `DEM_TRIGGER_ON_FDC_THRESHOLD`, one OBD freeze frame is available for each OBD-relevant event stored in the primary memory.

Single OBD freeze frame variant:

The `Dem` provides only one single OBD freeze frame record for all OBD-relevant events. This OBD freeze frame record stores the freeze frame data for the OBD-relevant event that is reported as FAILED and has the highest event priority.

Since there is only one global OBD freeze frame supported, the OBD freeze frame is always stored as freeze frame record `0x00`, as shown in [Figure 4.13, “OBD freeze frame record”](#).

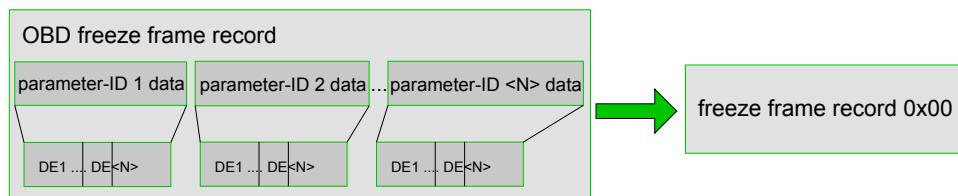


Figure 4.13. OBD freeze frame record



The OBD freeze frame is stored at the first FAILED report of an OBD-relevant event. Afterwards, the OBD freeze frame does not get updated by the same event that caused the storage of the OBD freeze frame. If another OBD-relevant event is reported as FAILED at a later point in time, the new event will update the OBD freeze frame of the former event only if it has a higher event priority, i.e. a lower value of configuration parameter [DemEventPriority](#).

The OBD freeze frame is cleared under the following circumstances:

- ▶ The associated OBD-relevant event changes its event status from FAILED to PASSED, i.e. the pending DTC bit PDTC changes from 1 to 0, and the event is not confirmed yet, ConfirmedDTC bit = 0.
- ▶ The associated OBD-relevant event gets aged.

Multiple OBD freeze frames variant:

The OBD freeze frame is stored on first error recognition, when UDS DTC status bit 2 (PDTC) transitions from 0 to 1 on the event memory entry creation.

The OBD freeze frame of an event is cleared when it is no longer pending or confirmed. An event is pending when its UDS DTC status bit 2 (PDTC) is set. An event is confirmed when its UDS DTC status bit 3 (CDTC) is set. The clearing occurs e.g. during aging, displacement, or clearing of emission data.

The OBD freeze frame of an OBD-relevant event is not updated with the new data on error re-occurrence.

Reporting of OBD freeze frame: Even though each emission-related event stored in the primary memory has a separate OBD freeze frame, only one is reported via service \$02 and \$19 with frame number as \$00. The freeze frame to be reported is selected as follows:

- ▶ The OBD freeze frame is reported only if the event is confirmed (CDTC = 1).
- ▶ If there are multiple events that are confirmed, then the OBD freeze frame of the event with the highest event priority is reported.
- ▶ If there are multiple confirmed events, all having the same highest priority, the freeze frame of the earliest entered entry (i.e. oldest) is reported.

**NOTE****Configuration of OBD freeze frames**

It is not possible to configure an OBD freeze frame explicitly via a dedicated [DemFreezeFrameClass](#).

To configure an OBD freeze frame, configure at least one parameter-ID (PID) in configuration parameter [DemPidClass](#). If at least one PID is configured, the OBD freeze frame is captured for OBD-relevant events as described above.

The OBD freeze frame is available only with EB tresos AutoCore Generic 8 Diagnostic Stack OBD.

4.5.2.3.2.1. Data elements and read functions

You can configure two types of data elements:

- ▶ external data elements ([DemExternalCSDataElementClass](#) and [DemExternalSRDataElementClass](#))
- ▶ internal data elements ([DemInternalDataElementClass](#))

In order for the `Dem` to collect the most recent data, synchronous server call points are used.

For external client-server data elements, the following two types of callback prototypes are supported that can be configured at the data element level:

- ▶ AUTOSAR-compatible callback prototype
- ▶ extended prototype

With the extended prototype, the extra parameter [DemEventId](#) is used. To configure a client-server data element callback with extended prototype, enable the parameter [DemDataElementInterfaceWithEventParameter](#).

For information about integration restrictions, see [Dem integration issues](#).

NOTE**Use of symbolic name for DemEventId**

It is recommended to use the symbolic name provided by the `Dem` for the [DemEventId](#) parameter. Do not use the numeric value because the [DemEventId](#) depends on the current `Dem` configuration. The callback shall avoid any interpretation of the [DemEventId](#) because only a pass-through mechanism is allowed.

4.5.2.4. Operation cycle

The processing of events is based on *operation cycles*. Examples for operation cycles are:

- ▶ Power cycle
- ▶ Engine warm-up cycle
- ▶ OBD driving cycle

The `Dem` performs the event handling based on the state of the associated operation cycles, but does not start or end these cycles. Starting and ending an operation cycle is triggered by the application via the client/server interface `SetOperationCycleState`.

As shown in [Figure 4.14, “`Dem` operation cycle”](#), an event is only processed by the `Dem` if the associated operation cycle has started and not yet ended. The `Dem` processes and stores events within the operation cycle. The gray arrow indicates that the `Dem` ignores the reported event beyond the operation cycle.

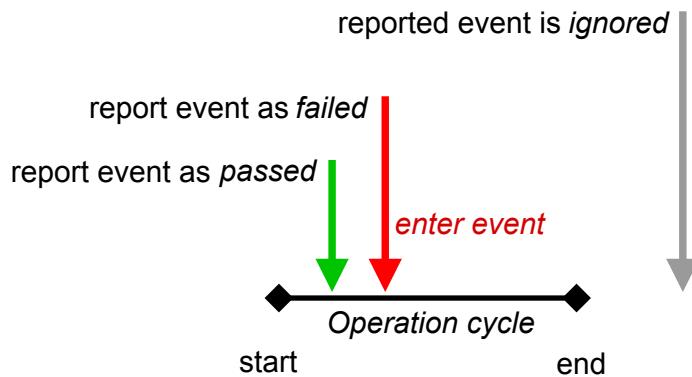


Figure 4.14. `Dem` operation cycle

The OBD driving cycle is a special case. This cycle can be additionally qualified. An event associated with an OBD driving cycle can be processed before qualification but the update of the confirmed status and MIL is provided only after the driving cycle is qualified. For details, see [Section 4.5.2.4.1, “OBD driving cycle \(DCY\)”](#).

4.5.2.4.1. OBD driving cycle (DCY)

The OBD driving cycle is a specific type of operation cycle used for emission-related ECUs. In emission-related ECUs, only one operation cycle shall be supported, which is identical to the driving cycle as defined by legislation. The OBD driving cycle is enabled by configuring an operation cycle via `DemOperationCycle` and assigning it the type `DEM_OPCYC_OBD_DCY` in `DemOperationCycleType`.

As shown in [Figure 4.15, “`Dem` OBD driving cycle”](#), diagnostic events that are linked to the OBD driving cycle are handled differently than events that are linked to other operation cycles. This is because the OBD driving cycle can be additionally qualified.

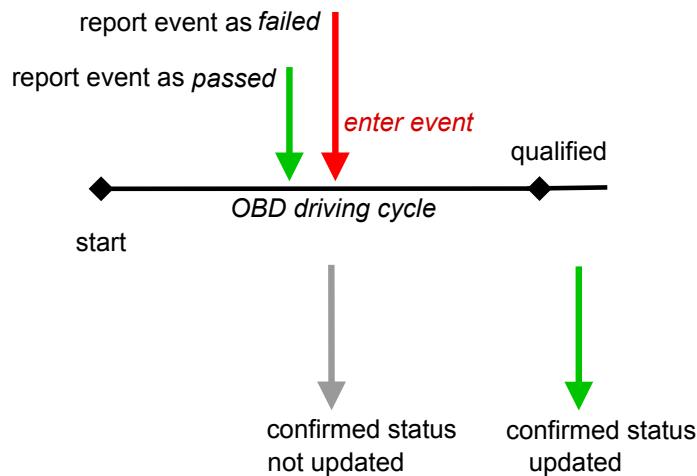


Figure 4.15. Dem OBD driving cycle

The gray arrow indicates that the reported events are processed but the confirmed status is not updated because the OBD driving cycle is not yet qualified. The green arrow indicates that once the OBD driving cycle is qualified, the `Dem` updates the confirmed status.

Thus, reported events that are linked to an OBD driving cycle are processed as usual but the confirmation status is updated only after qualification of the OBD driving cycle. The moment the OBD driving cycle is set to qualified, the `Dem` executes computations required to reach the confirmed states of events collected during the *not qualified* phase of the cycle. If these events reference the MIL, the activation of the MIL is handled accordingly.

The OBD driving cycle must not be ended or restarted before it was qualified. This must be guaranteed by the application. Consequently, it is also required that the OBD driving cycle stays active over power cycles. This means the configuration parameter [`DemOperationCycleStatusStorage`](#) must be TRUE when the OBD driving cycle is used.

4.5.2.4.2. OBD warm-up cycle (WUC)

The warm-up cycle (WUC) is a legally required cycle and is computed by the Master ECU. The Master ECU provides the information about start and end of the WUC via the bus system. The `Dem` requires this information to calculate aging of emission-related events and for internal PID calculations.

If a standard AUTOSAR aging mechanism is used, all emission-related events that are configured to support aging (refer to parameter [`DemAgingAllowed`](#)) shall use one common `DemAgingCycle` (refer to parameter [`DemAgingCycleRef`](#)) with the same trigger conditions as the WUC. The application must report these triggers via the client/server interface `AgingCycle`.

For typical OBD ECUs, DCY (used for healing) and WUC (used for aging) are triggered together. The DCY is started/restarted via [`Dem_SetOperationCycleState\(\)`](#), which leads to the processing of healing at DCY



transition. The WUC is triggered via [Dem_SetAgingCycleState\(\)](#), which leads to the processing of aging. If healing-dependent aging is configured (i.e. parameter [DemAgingBasedOnHealingCompletionSupport](#) is enabled) and the WUC is triggered after the restart of the DCY, the final healing cycle is also counted towards aging. To avoid this, the WUC must be triggered at the end of a DCY, i.e. immediately before restarting the DCY. Therefore, the following call sequence is recommended:

1. Dem_SetAgingCycleState (WUC)
2. Dem_SetOperationCycleState (DCY, STATE_START)

If [Dem_SetAgingCycleState\(\)](#) is called at the beginning of a DCY, an additional aging cycle must be configured (e.g. 41 instead 40) to finish aging at the expected point in time. An example with [DemAgingCycleCounterThreshold](#) = 3 is shown in [Figure 4.16, “Dem OBD aging”](#):

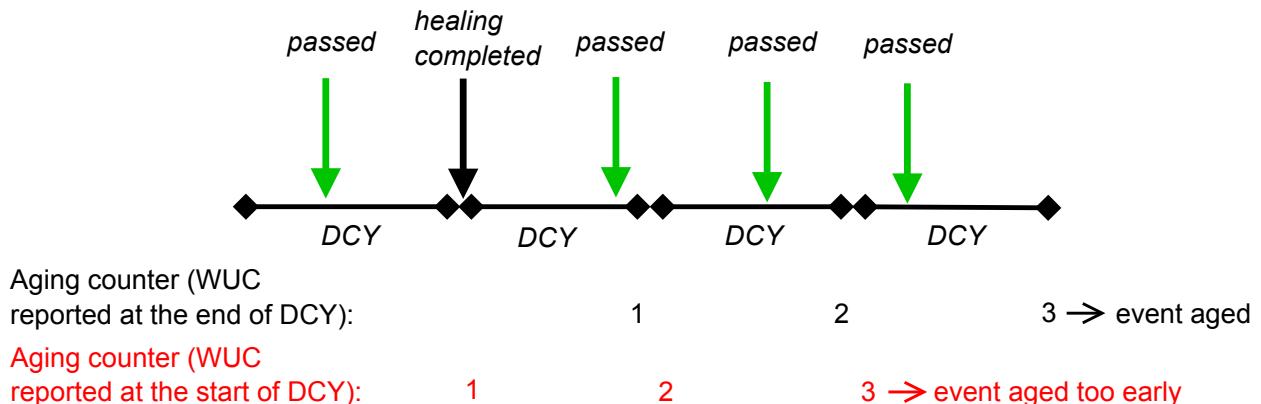


Figure 4.16. Dem OBD aging

For the OCC5-based aging mechanism and for internal PID calculations, the Dem requires a dedicated Dem-OperationCycle of type [DEM_OPCYC_WARMUP](#) (refer to parameter [DemOperationCycleType](#)). Based on the WUC trigger conditions, the application must report this cycle via the client/server interface [OperationCycle](#).

NOTE

No specific aging cycle required for OCC5

If OCC5 is used for aging, an event-specific aging cycle is not required.



4.5.2.5. Event aging

Aging (or *unlearning*) of an event means that an event, which no longer fails, is deleted from the event memory after a configurable number of operation/aging cycles. This is shown in [Figure 4.17, “Event aging”](#).

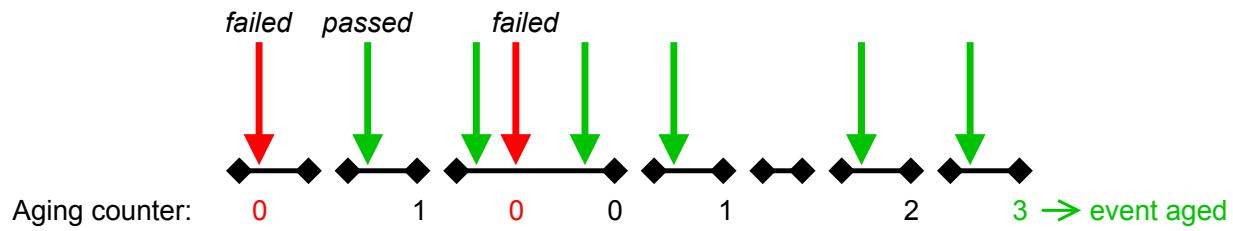


Figure 4.17. Event aging

Aging for a specific event can be enabled with the event-specific parameter `DemAgingAllowed`.

Depending on the configuration, the aging counter is initialized with 0. It is incremented after each operation cycle, in which the event is tested and not failed.

If the aging counter reaches the configured threshold value, the event is removed from the event memory.

The aging counter value can be mapped as upcounting or downcounting value to an extended data record with [DemInternalDataElement](#).

By default, aging is controlled via the regular operation cycle with [Dem_SetOperationCycleState\(\)](#) API. A separate aging cycle can be configured with [DemAgingCycleRef](#) and be reported with [Dem_SetAgingCycleState\(\)](#) API.

**NOTE****Call-point of Dem_SetAgingCycleState() affects aging behavior**

The aging processing is done synchronously at the call of [Dem_SetAgingCycleState\(\)](#).

This means that the point in time when `Dem_SetAgingCycleState()` is called during the operation cycle affects the aging behavior as follows:

If `Dem_SetAgingCycleState()` is called at the beginning of the operation cycle, the aging processing is based on the event status of the previous cycle. Also, aging does not work at all if `Dem_SetAgingCycleState()` is called at the beginning of the operation cycle and [DemAgingRequiresTestedCycleSetOperationCycleState](#) is set to true. In this case, the TNCTOC bit is set and hence, the aging criteria are never fulfilled.

If `Dem_SetAgingCycleState()` is called in the middle of an operation cycle, only a part of the operation cycle is taken into account for aging.

If `Dem_SetAgingCycleState()` is called several times during an operation cycle, also the aging counter is incremented several times. The incrementation is possibly based on the same event report if the event was not reported again in the meantime.

See also [Figure 4.16, “Dem OBD aging”](#) for an example of the aging cycle counting depending on if `Dem_SetAgingCycleState()` is called at the beginning or at the end of the operation cycle.

The aging mechanism can further be configured if only tested cycles or also untested cycles shall be considered, refer to parameters [DemAgingRequiresTestedCycleSetOperationCycleState](#) and [DemAgingRequiresTestedCycleSetAgingCycleState](#).

You can configure when aging should start with parameter [DemAgingBasedOnHealingCompletionSupport](#).

An aged event whose DTC record update is disabled shall be cleared from event memory only after a DTC record update is enabled or during a `Dem` shutdown.

4.5.2.6. Debouncing

The `Dem` can smooth unsteady event states by applying different debounce algorithms. The `Dem` works as a kind of *filter* if debouncing is configured. As the event status is the output of the debounce algorithm, the reporting components do typically not report the following states directly:

- ▶ DEM_EVENT_STATUS_PASSED
- ▶ DEM_EVENT_STATUS_FAILED

Instead, the components report the following states:

- ▶ DEM_EVENT_STATUS_PREPASSED



► DEM_EVENT_STATUS_PREFFAILED

If DEM_EVENT_STATUS_PASSED or DEM_EVENT_STATUS_FAILED is reported directly, then:

- The currently running debouncing is stopped
- The fault detection counter is set to minimum/maximum value
- The event status is qualified as passed/failed directly

The debounce algorithm is applied on the pre-status and the output is the event status. The debounce algorithm is applied by an external function, which the Dem calls or by the following Dem internal algorithms:

- Counter-based
- Time-based
- Frequency-based

[Figure 4.18, “Counter-based debounce”](#) shows the counter-based debouncing.

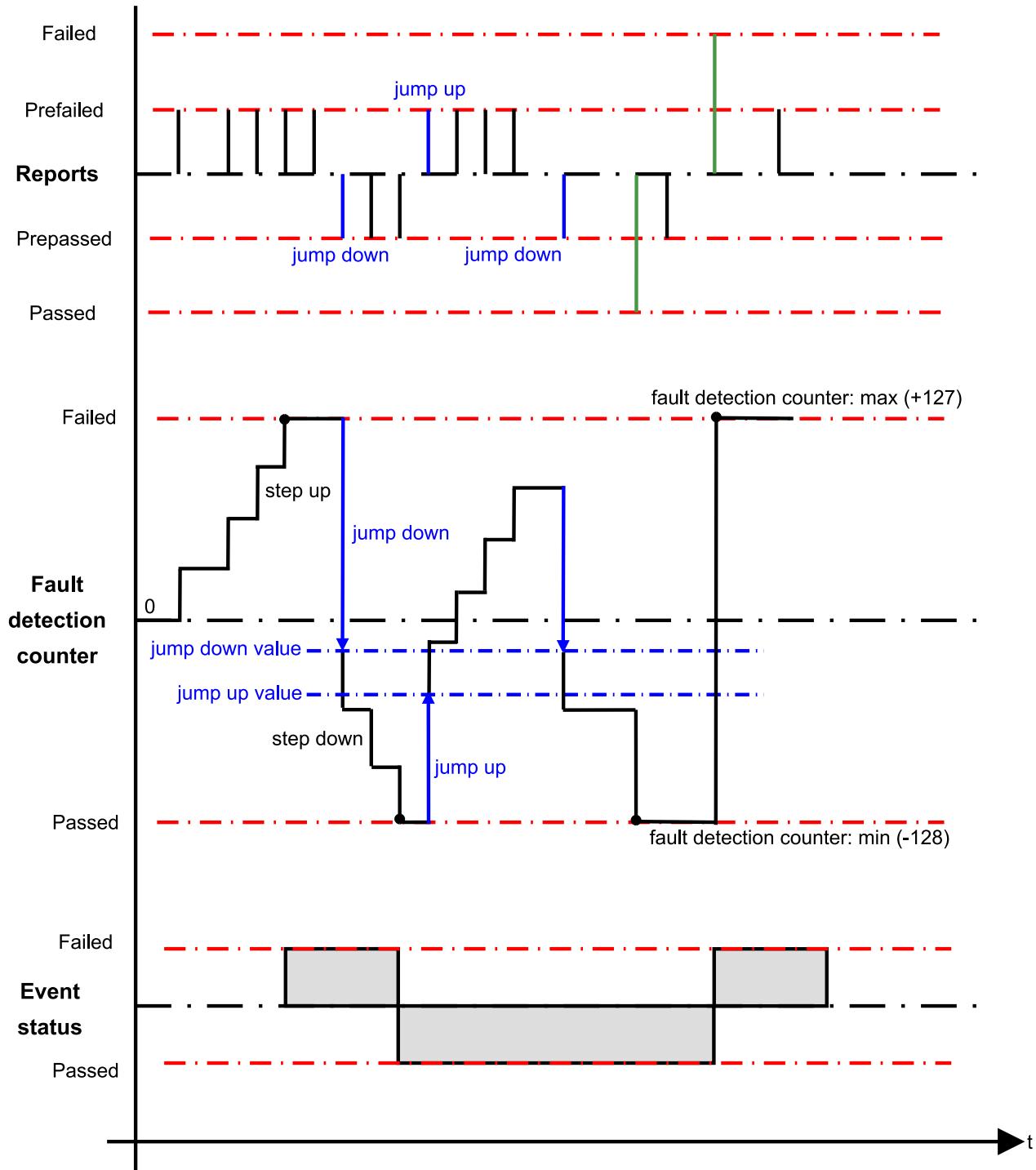


Figure 4.18. Counter-based debounce

- With each reported *prefailed* event, the *fault detection counter* is increased by a configurable *step up* size until the maximum value of 127 is reached. Then the *event status* is set to *failed*.



- ▶ With each reported *prepassed* event, the *fault detection counter* is decreased by a configurable *step down* size until the minimum value of -128 is reached. Then the *event status* is set to *passed*.
- ▶ Additionally the *jump down* feature can be configured. If the reporting of the pre-events switches from *prefailed* to *prepassed*, the fault detection counter is set to jump-down value first and stepped down after that.
- ▶ Also configurable is the *jump up* feature. If the reporting of the pre-events switches from *prepassed* to *prefailed*, the fault detection counter is set to the jump-up value first and stepped up after that.

[Figure 4.19, “Time-based debounce”](#) shows the time-based debouncing.

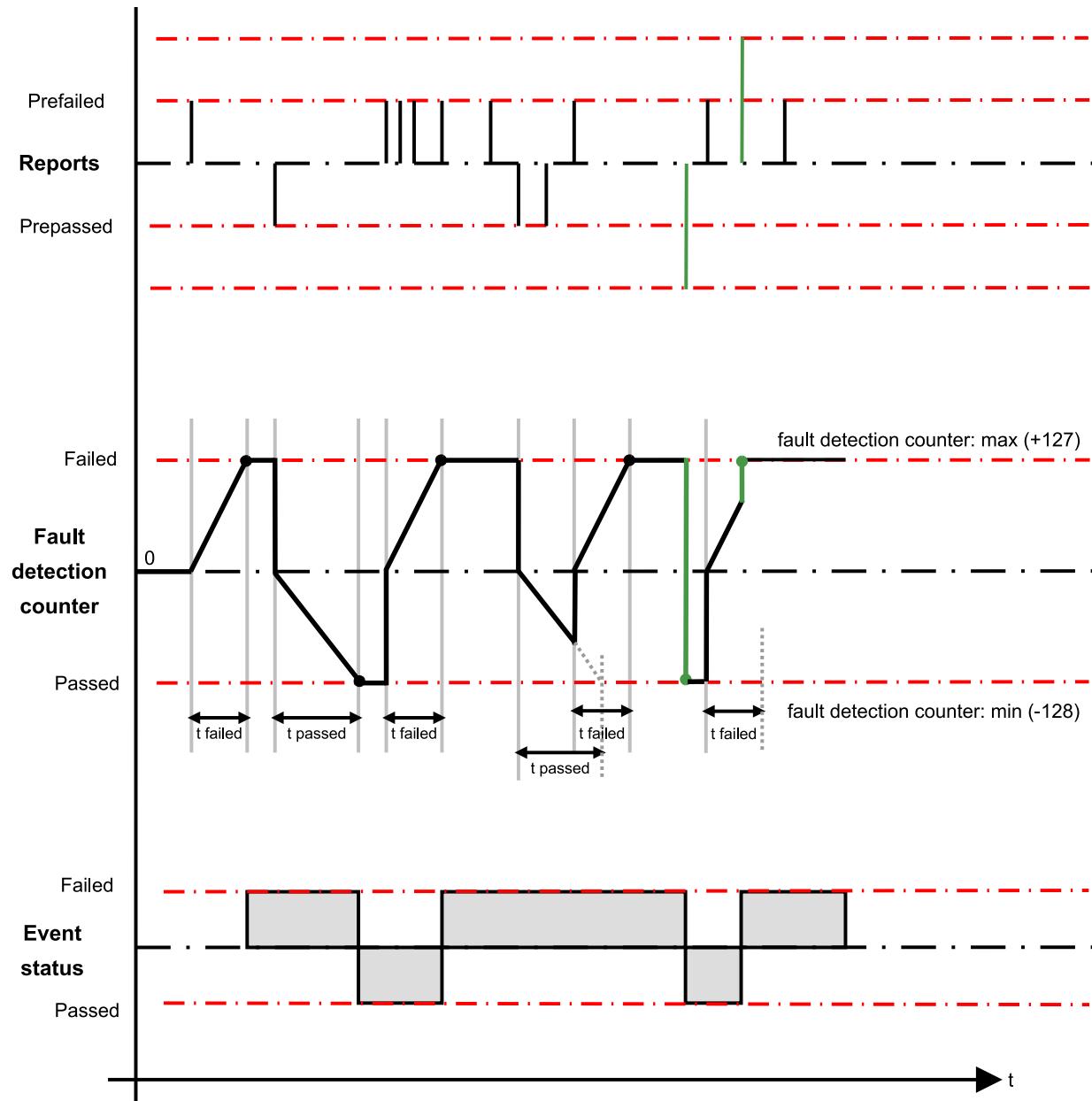


Figure 4.19. Time-based debounce

- If the reporting of the events switches from prepassed/passed to prefailed, the internal debounce timer is started. After the configurable time t_{failed} , the fault detection counter reaches its maximum value of 127 and the event status is set to *failed*.

If the internal debounce timer of a specific event is already triggered and the monitor reports consecutively prefailed again, the internal debounce timer is not restarted again.



- ▶ If the reporting of the events switches from prefailed/failed to prepassed, the internal debounce timer is started. After the configurable time τ_{passed} , the fault detection counter reaches its minimum value of -128 and the event status is set to *passed*.

If the internal debounce timer of a specific event is already triggered and the monitor reports consecutively prepassed again, the internal debounce timer is not restarted again.

- ▶ The fault detection counter is linearly derived from internal timer τ .

[Figure 4.20, “Frequency-based debounce”](#) shows the frequency-based debouncing.

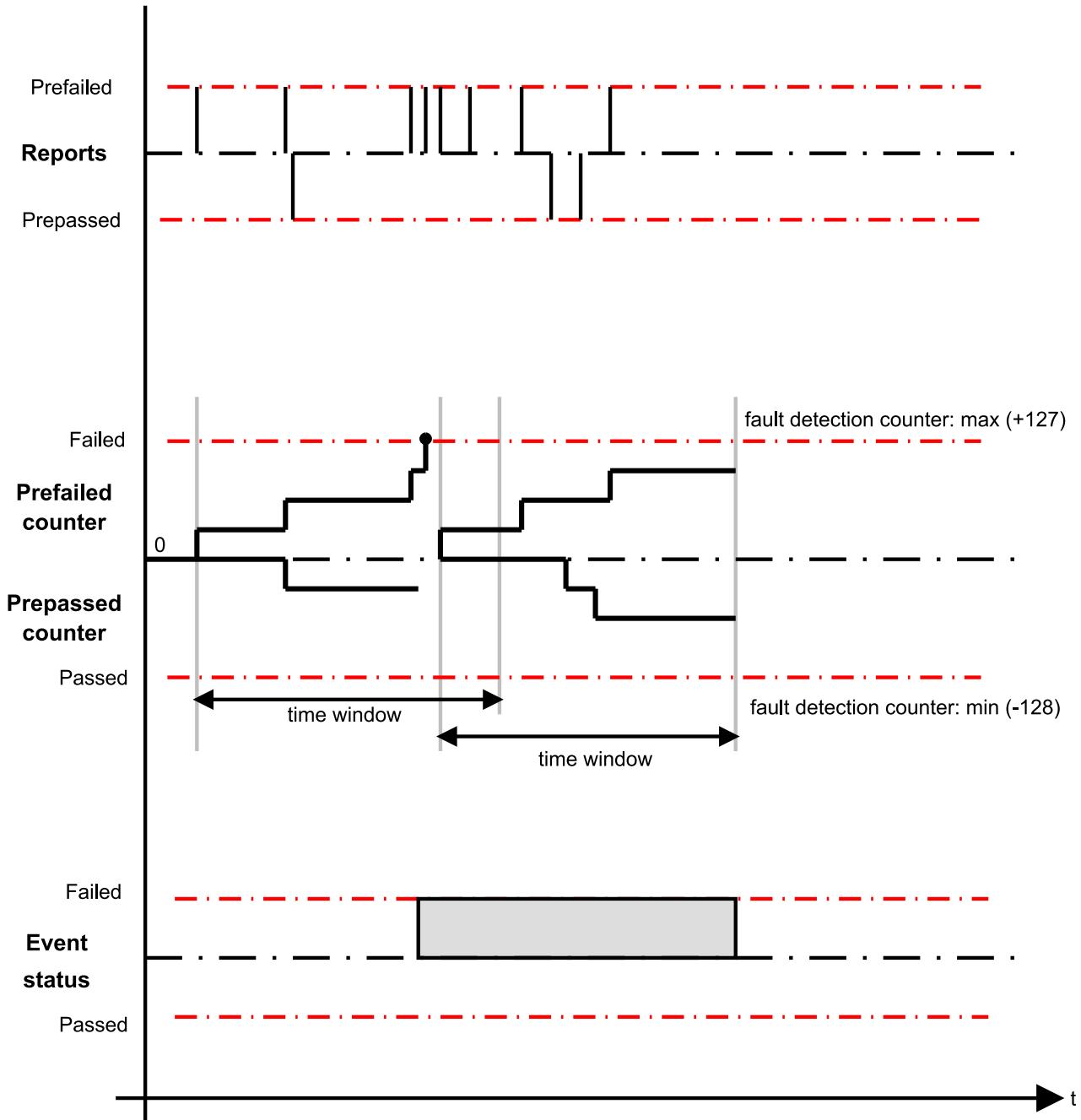


Figure 4.20. Frequency-based debounce

After the first pre-event status is reported, a configurable time window is started. There are two different counters: One counts the failed events within the time window, the other counts the passed events within the time window. If the maximum threshold is reached by the failed events counter within the time window, the event



status is set to *failed*. If the minimum threshold is reached by the passed event counter within the time window, the event status is set to *passed*.

The time window is restarted with the reporting of the next pre-event status.

4.5.2.6.1. Calibration

This feature allows you to change configuration parameters by means of a measurement, calibration and diagnostics (MCD) tool. Calibration support of specific configuration parameters can be enabled with the configuration parameter [DemCalibrationSupport](#).

In the current context, calibration enables the fine-tuning of the parameters concerning event debouncing. For example, the debouncing thresholds might not be exactly known during software development for a specific functionality and its associated events, and they can be determined only during ECU testing. By enabling the calibration feature, you can then change those threshold values to the ones that best suit the functionality during the calibration phase.

4.5.2.6.1.1. Class concept

It is critical to understand the class concept upon which calibration is based, and in particular the relationship between the data structures `Dem_DebounceCounterClass` and `Dem_DebounceCounterClassIdx` or `Dem_DebounceTimeClass` and `Dem_DebounceTimeClassIdx` for counter-based or time-based debounced events, respectively. These data structures are all generated inside the file `Dem_Cfg.c`.

On the one side, during code generation and for both counter-based and time-based debounced events, the code generator collects into distinct classes all distinct combinations of debouncing configurations (thresholds, step sizes, etc.) from [DemDebounceCounterBased](#) and [DemDebounceTimeBase](#), and adds them to the configuration classes `Dem_DebounceCounterClass` and `Dem_DebounceTimeClass`, respectively.

On the other side, the generator associates to each event an entry inside the index tables `Dem_DebounceCounterClassIdx` or `Dem_DebounceTimeClassIdx`, which represents a class index pointing to a debouncing class inside `Dem_DebounceCounterClass` or `Dem_DebounceTimeClass`, and which represents the link between the event and its associated debouncing class.

Which of the two tables must be considered depends on the algorithm type configured for the debouncing:

- ▶ If the debouncing for a given event is configured as **counter-based**, then the index identified by `Dem_DebounceCounterClassIdx` points to a configuration inside `Dem_DebounceCounterClass`.
- ▶ If the debouncing for a given event is configured as **time-based**, then the index identified by `Dem_DebounceTimeClassIdx` points to a configuration inside `Dem_DebounceTimeClass`.

Concerning the classes inside `Dem_DebounceCounterClass` and `Dem_DebounceTimeClass`, note that in addition to the classes determined by the code generator based on the configuration, further blank classes can



be made available by configuring the parameters [DemMaxNumberDebCounterClasses](#) and [DemMaxNumberDebTimeClasses](#).

4.5.2.6.1.2. Performing the calibration

WARNING**Static configuration**

Keep in mind that you cannot add further configuration elements during the calibration phase. You can only change the properties of already existing elements, because the number of resources in the ECU (DTCs, events, indicators, debounce classes, etc.) is statically defined during configuration.

As mentioned in [Section 4.5.2.6.1, “Calibration”](#), the calibration support of specific configuration parameters can be enabled with the configuration parameter [DemCalibrationSupport](#). See the documentation of this parameter for an exhaustive list of all the elements that can be calibrated.

If you want to change the passed and failed counter thresholds for a given counter-based debounced event, first identify the class inside `Dem_DebounceCounterClass` associated to such an event as described in [Section 4.5.2.6.1.1, “Class concept”](#). Then perform the calibration by directly changing the threshold values inside the identified class by using an MCD tool.

Consider that a class can be referenced by several events. A change of this class would affect also other events referencing such class. If this is not desired, then you can have the generator provide blank classes to be used during calibration by means of the parameter [DemMaxNumberDebCounterClasses](#).

During calibration, you then choose one of such blank classes and adapt the counter thresholds as required. Next, you need to associate this new class to the event that you consider by changing the class index inside the table `Dem_DebounceCounterClassIdx`, so that it points now to the new class that you have just calibrated.

This calibration procedure is shown in [Figure 4.21, “Debouncing class configuration before calibration”](#) and [Figure 4.22, “Debouncing class configuration after calibration”](#).

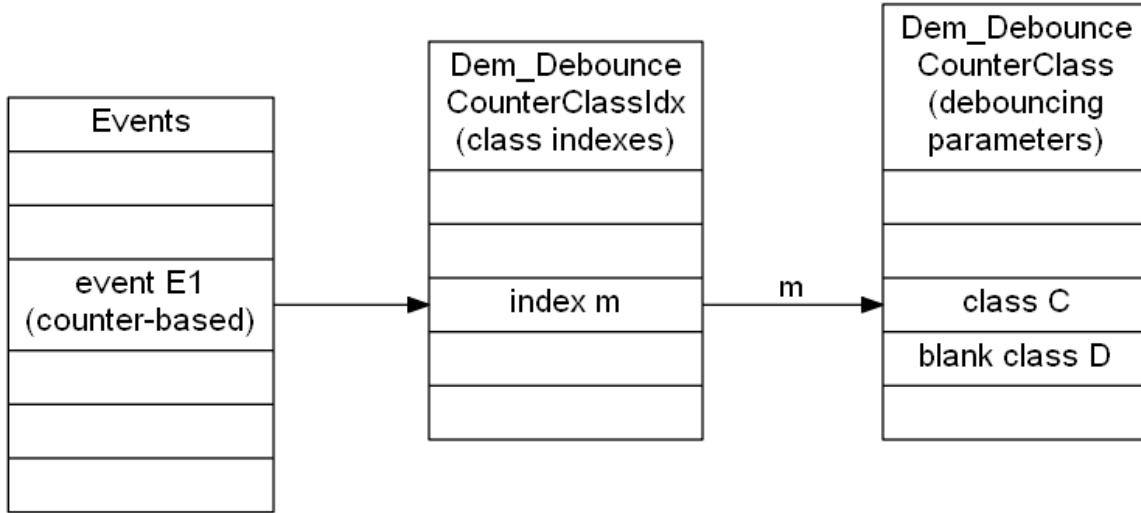


Figure 4.21. Debouncing class configuration before calibration

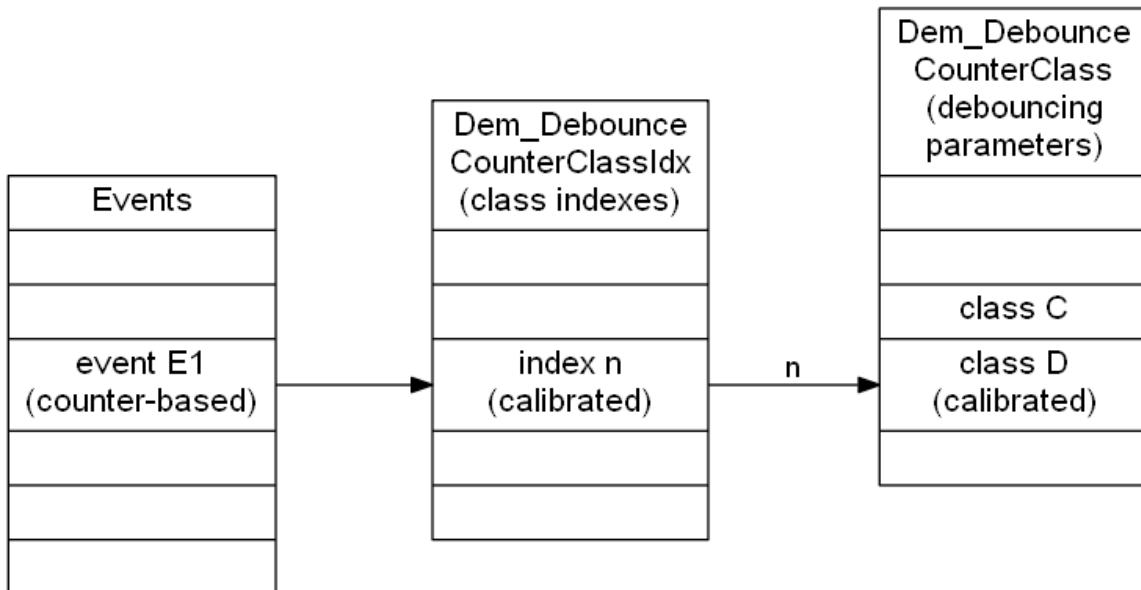


Figure 4.22. Debouncing class configuration after calibration

In [Figure 4.21, “Debouncing class configuration before calibration”](#), you can see that event E1 is a counter-based debounced event, and its associated counter-based debouncing class is given by the m-th element of Dem_DebounceCounterClass, i.e. class C. You can also see that a blank class is available for calibration, i.e. class D.

In [Figure 4.22, “Debouncing class configuration after calibration”](#), you can see what happens when event E1 is calibrated. By means of the MCD tool, the available blank class D was calibrated, corresponding to the n-th element of Dem_DebounceCounterClass. And it was assigned to the event E1 by calibrating the corresponding element of Dem_DebounceCounterClassIdx from the previous index m to the new index n.



The final result is that now the associated counter-based debouncing class of event E1 is given by the n-th element of `Dem_DebounceCounterClass`, i.e. class D.

The same concept also applies to time-based debounced events, where the data structures to be calibrated are `Dem_DebounceTimeClassIdx` and `Dem_DebounceTimeClass`, and the number of extra classes to be generated is given by the parameter [DemMaxNumberDebTimeClasses](#).

NOTE

Debounce behavior

Note that for both counter-based and time-based debounced events, the debounce behavior (parameter [DemDebounceBehavior](#)) cannot be calibrated!

The extra classes that are generated have the default behavior configured, i.e. `DEM_DEBOUNCE_FREEZE` for counter-based debouncing and `DEM_DEBOUNCE_RESET` for time-based debouncing.

If you require extra classes with a different behavior than the default behavior, then either you must ensure the generation of all needed classes for the existing events by configuration or you must configure *dummy events* with the desired debouncing behavior in order to have the generator provide such extra classes for later calibration use.

4.5.2.7. Enable conditions

Enable conditions allow you to enable or disable the processing of events by the `Dem`. To ease the configuration, enable conditions are grouped to enable condition groups (refer to configuration parameters [DemEnableConditionGroup](#) and [DemEnableCondition](#)). These groups can be assigned to certain events. Thus, one event can have different enable conditions assigned. An event can have only one enable condition group assigned (refer to configuration parameter [DemEnableConditionGroupRef](#)).

Status reports for events that have an enable condition group assigned are ignored by the `Dem` until all enable conditions grouped in the associated enable condition group are fulfilled. The status of the enable conditions (fulfilled / not fulfilled) is defined by the application and can be reported to the `Dem` by the client server interface `EnableCondition`.

4.5.2.8. Event combination

Event combination defines the ability of the `Dem` module to merge several events to one DTC. In this document, the merged events are called *sub-events*. Such a combined DTC provides one combined DTC status, which is calculated by a logical operation of all associated sub-event statuses.

AUTOSAR in general defines the following two types of event combination:

- ▶ **Event combination - Combined type 1 (Combination on storage)**



- ▶ The combined DTC is stored and updated in one single event memory entry
- ▶ The combined DTC status is the trigger for allocation of the event memory entry and the collection and update of its event-related data
- ▶ The aging of the combined DTC is calculated based on the combined DTC status
- ▶ **Event combination - Combined type 2 (Combination on retrieval)**
 - ▶ Each event is stored in a separate event memory location
 - ▶ The event status of each combined event is the trigger for allocation of the associated event memory entry and the collection and update of the dedicated event-related data
 - ▶ The aging of each event of a combined DTC is calculated individually for each sub-event

The `Dem` module currently supports event combination type 1 (combination on storage) only.

To enable event combination, take the following steps:

- ▶ Configure the configuration parameter [`DemEventCombinationSupport`](#) to `DEM_EVCOMB_TYPE1`
- ▶ [`DemDTCClassRef`](#) refers to a DTC class that is already referenced by another event.

An event is recognized as combined if its configuration parameter [`DemDTCClassRef`](#) refers to a DTC class that is already referenced by another event.

WARNING

Event limitations



The following limitations apply to the current implementation of event combination in `Dem`:

- ▶ Event combination type `DEM_EVCOMB_TYPE2` (`DEM_EVCOMB_ONRETRIEVAL`) is not supported
- ▶ Event combination is supported only if calibration support is disabled
- ▶ Event combination is supported only if OBD support is disabled
- ▶ Prestorage of freeze frames is not supported for combined DTCs

4.5.2.8.1. Event combination type 1 (Combination on storage)

Event combination type 1 uses only one event memory entry for the combined DTC. Therefore, several parameters of all sub-events combined by one combined DTC must be configured with the same values. To ease the configuration and avoid configuration mistakes, the EB `Dem` specifies the additional parameter [`DemEvtCmbCommonParamMaster`](#). This parameter can be configured to TRUE or FALSE for events that are combined by a combined DTC.

Every combined DTC must define exactly one event as common parameter master event by configuring the parameter [`DemEvtCmbCommonParamMaster`](#) to TRUE. This common parameter master event defines the



configuration values for all common configuration parameters of all sub-events combined by a certain DTC. All sub-events that are not defined as common parameter master event, but combined by the same DTC, will derive the configuration values of the common parameter master event. Therefore the common parameters for those sub-events are no longer editable in the configuration.

Common event parameters:

The following parameters are common for all events combined by a dedicated DTC, and are defined by the configuration of the common parameter master event.

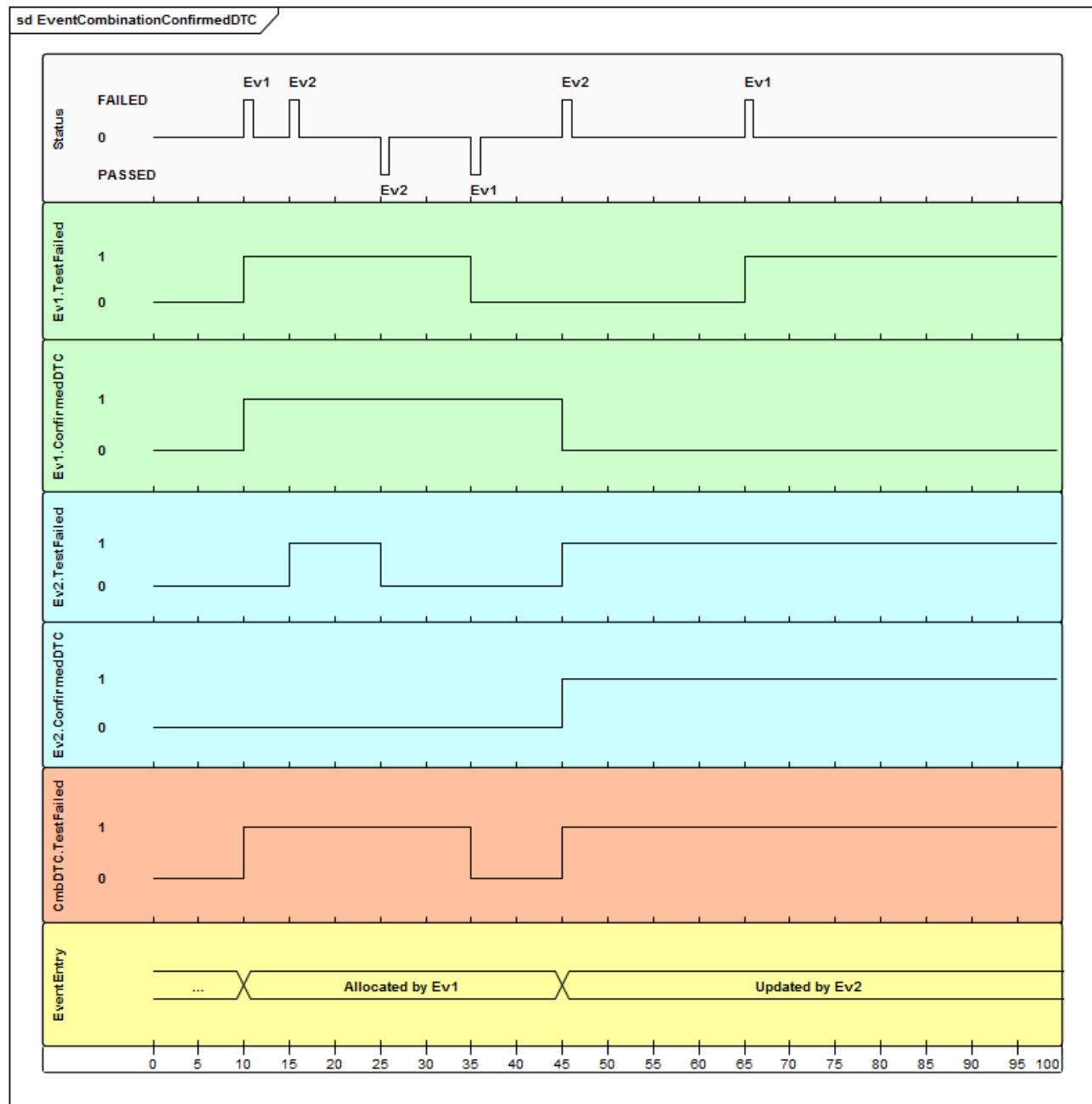
- ▶ [DemExtendedDataClassRef](#)
- ▶ [DemFreezeFrameClassRef](#)
- ▶ [DemFreezeFrameRecNumClassRef](#)
- ▶ [DemMaxNumberFreezeFrameRecords](#)
- ▶ [DemEventDestination](#)
- ▶ [DemEventPriority](#)
- ▶ [DemEventSignificance](#)
- ▶ [DemStorageConditionGroupRef](#)
- ▶ [DemAgingCycleRef](#)
- ▶ [DemAgingAllowed](#)
- ▶ [DemAgingCycleCounterThreshold](#)
- ▶ [DemEventFailureCycleRef](#)
- ▶ [DemEventFailureCycleCounterThreshold](#)

Processing of combined events:

The transition from 0 to 1 of the `TestFailed` bit of the combined DTC triggers the allocation of an event memory entry and the collection of the event-related data for the associated combined DTC. The `TestFailed` bit of the combined DTC status is defined by a logical OR operation over all sub-events combined by a dedicated DTC. The combined sub-event, that caused the allocation is located in the event memory until the next 0 → 1 transition of the combined `TestFailed` bit is performed.

If the sub-event that allocated the event memory entry gets PASSED (`TestFailed` bit = 0) and no other sub-event combined by the same DTC has its `TestFailed` bit set, the next combined sub-event that gets FAILED updates the event memory entry and the event-related data.

[Figure 4.23, “Handling of combined DTCs with confirmedDTC bit”](#) shows this behavior.

Figure 4.23. Handling of combined DTCs with `confirmedDTC` bit

**NOTE****Confirmed status of combined events**

Since only one event memory entry is used with event combination type 1, only one of the sub-events that are combined by a DTC can be confirmed, `ConfirmedDTC` bit = 1, at the same time.

Thus, if a sub-event that triggered the storage of the combined DTC and has the `ConfirmedDTC` bit set but is not actually failed, `TestFailed` bit = 0, and in addition the `TestFailed` bit of the combined DTC status is also 0, the next sub-event, combined by the same DTC, that gets failed and confirmed, updates the event entry. The `ConfirmedDTC` bit of the sub-event that triggered the combined DTC storage gets reset to 0.

Displacement of combined DTCs:

If a combined DTC is displaced by another DTC due to a full event memory, the `Dem` removes the event memory entry and resets the event status bits of all sub-events combined by the dedicated DTC.

The following status bits get reset depending on configuration parameter [`DemResetConfirmedBitOnOverflow`](#), [`DemResetPendingBitOnOverflow`](#), and [`DemStatusBitHandlingTestFailedSinceLastClear`](#):

- ▶ PendingDTC bit
- ▶ ConfirmedDTC bit
- ▶ TestFailedSinceLastClear bit

Aging of combined DTCs:

To age a combined DTC, all sub-events combined by the dedicated DTC must be aged. Therefore the aging may take longer than configured for the sub-events. The aging cycle as well as the aging cycle counter threshold are defined by the event combination parameter master event and therefore the same for all sub-events. You can configure the aging cycle with the configuration parameter [`DemAgingCycleRef`](#) and the aging cycle counter threshold with [`DemAgingCycleCounterThreshold`](#).

If a combined DTC is aged, the `Dem` removes the event memory entry and resets the `ConfirmedDTC` bit (event status bit 3) of the sub-event that currently occupies the event memory entry.

Intermediate storage of combined DTC status:

If the feature is enabled, the `Dem` stores the UDS status bytes of all sub-events, not just the combined DTC status, whenever the event-related data of the associated combined DTC is stored. For more information, see [Section 4.5.2.3.1, “Memory types”](#).



4.5.2.9. Multi-event triggering

The multi-event triggering enables you to multiply calls of certain APIs for defined events. When a defined master event is reported, it triggers the reporting of other slave events. Currently, only the [Dem_SetEventStatus\(\)](#) and [Dem_ReportErrorStatus\(\)](#) APIs are supported.

To enable multi-event triggering, configure at least one [DemMultiEventTriggering](#) configuration parameter. You can configure several multi-event triggering configurations. Each configuration consists of a defined master event and a group of slave events that are triggered when the master event is reported. The master event is referenced in configuration parameter [DemMultiEventTriggeringMasterEventRef](#). The slave events are referenced in configuration parameter [DemMultiEventTriggeringSlaveEventRef](#).

For each multi-event triggering configuration as configured with [DemMultiEventTriggering](#), you can select an alternative behavior. The default configuration is `DEM_MULTIEVENTTRIGGERING_ALT_NONE`. With the default configuration, the master event is processed along with the slave events. In this case, the slave events are processed first and the master event is processed at the end. As an alternative behavior, you can select the exclusive replacement of the master event. For details, see [Section 4.5.2.10, “Replacement DTC”](#).

When the master event is reported via [Dem_ReportErrorStatus\(\)](#), after [Dem_PreInit\(\)](#) and before [Dem_Init\(\)](#), the slave events assigned to the multi-event triggering configuration are queued into the error-queue regardless of their event kind. The error-queue is processed asynchronously in the next call of [Dem_MainFunction\(\)](#).

If multi-event triggering is enabled, the `Dem` module stores per memory entry a *root event ID*. For a regular reported event, the root event ID is the event ID of the event itself. If the event is triggered as a slave due to a multi-event triggering, the root event ID is the event ID of the initially reported master event. The root event ID value can be mapped to an extended data record with [DemInternalDataElement](#).

Example:

A multi-event triggering configuration with the alternative behavior set to `DEM_MULTIEVENTTRIGGERING_ALT_NONE` might have the following structure:

- ▶ [DemMultiEventTriggeringMasterEventRef](#): SWC event
- ▶ [DemMultiEventTriggeringSlaveEventRef](#):
 - ▶ SWC event
 - ▶ BSW event
 - ▶ SWC event
- ▶ [DemMultiEventTriggeringAlternativeBehavior](#): `DEM_MULTIEVENTTRIGGERING_ALT_NONE`



4.5.2.10. Replacement DTC

The replacement DTC is an alternative behavior of the multi-event triggering functionality (see [Section 4.5.2.9, “Multi-event triggering”](#)). It enables the exclusive replacement of the master event by one of the slave events. That means the slave event is processed instead of the master event.

To enable the exclusive replacement, set the [`DemMultiEventTriggeringAlternativeBehavior`](#) to `DEM_MULTIEVENTTRIGGERING_ALT_EXCLUSIVE_REPLACE` for a specific multi-event triggering configuration [`DemMultiEventTriggering`](#).

To select the slave event that is used for the replacement, configure the configuration parameter [`DemEnableCondition`](#). The master event is replaced by maximum one slave event. Therefore, you can only enable the enable condition group in [`DemEnableConditionStatus`](#) for maximum one of the slave events. Also during run-time, ensure that maximum one of the slave events is enabled. Otherwise it is impossible to predict which slave event replaces the master event.

If exclusive replacement is enabled, the master event is not processed. An exception is the case when no slave event is enabled. Then the master event is processed in the same way as if no [`DemMultiEventTriggering`](#) is configured for the master event.

During exclusive replacement, only the reports of `DEM_EVENT_STATUS_FAILED` are handled. Any other report of the master is processed by the master as it would be without multi-event triggering.

It is recommended to avoid using `Dem` internal debouncing algorithms for events referenced by multi-event triggering.

For the exclusive replacement of the master event, it is necessary that all slave events referenced in [`DemMultiEventTriggeringSlaveEventRef`](#) have a valid [`DemEnableConditionGroupRef`](#).

Furthermore, each referenced slave event must have exactly one enable condition defined in its [`DemEnableConditionGroupRef`](#).

Example:

A multi-event triggering configuration with the alternative behavior set to `DEM_MULTIEVENTTRIGGERING_ALT_EXCLUSIVE_REPLACE` might have the following structure:

- ▶ [`DemMultiEventTriggeringMasterEventRef`](#): BSW event
- ▶ [`DemMultiEventTriggeringSlaveEventRef`](#):
 - ▶ BSW event that references an enable condition group containing only one enable condition with status not fulfilled
 - ▶ BSW event that references an enable condition group containing only one enable condition with status fulfilled
 - ▶ SWC event that references an enable condition group containing only one enable condition with status not fulfilled



- ▶ [DemMultiEventTriggeringAlternativeBehavior](#): DEM_MULTIEVENTTRIGGERING_ALT_EXCLUSIVE_REPLACE

4.5.2.11. Monitor re-initialization

The `Dem` module provides the `InitMonitorForEvent` interface to trigger the initialization of a diagnostic monitor. The monitor re-initialization callback function is triggered in the following cases:

- ▶ At operation cycle start/restart (with `InitMonitorReason`: DEM_INIT_MONITOR_RESTART)
- ▶ At DTC clearing (with `InitMonitorReason`: DEM_INIT_MONITOR_CLEAR)

Additionally, optional support for an extended range of `Dem_InitMonitorReasonType` as described in the AUTOSAR 4.3.0 specification is provided. This means that the `InitMonitorForEvent` callback function is also triggered with `InitMonitorReason`: DEM_INIT_MONITOR_REENABLED when:

- ▶ Enable condition group status of the event is changed to fulfilled.
- ▶ ControlDTCSetting of the event is changed to enabled.
- ▶ Event availability is changed to available.

To enable this behavior, set the [DemCallbackInitMForEReenabledSupport](#) configuration parameter to true.

4.5.2.12. User-controlled WarningIndicatorRequested-bit

The `Dem` module allows the user/application to control the UDS status `WarningIndicatorRequested-bit` (WIR-bit) via the API [Dem_SetWIRStatus\(\)](#). The configuration parameter [DemUserControlledWIRBitSupport](#) acts as an on/off switch for this feature.

The WIR-bit for a corresponding event is controlled via a special *failsafe* SWC. The failsafe SWC reports the required WIR status to the `Dem`. As a result, the WIR-bit of the corresponding event is set/reset accordingly. The `Dem` supports two ways of updating the WIR-bit:

- ▶ The special failsafe SWC requests a status change via API [Dem_SetWIRStatus\(\)](#) to:
 - ▶ FALSE: The WIR-bit is only reset if no `Dem` indicators are enabled that are connected to the event.
 - ▶ TRUE: The WIR-bit is set.
- ▶ The `Dem` warning indicator handling sets the WIR-bit status to:
 - ▶ FALSE: The WIR-bit must only be reset if the WIR-bit is not set by the API [Dem_SetWIRStatus\(\)](#).
 - ▶ TRUE: The WIR-bit is set.



The user-controlled WarningIndicatorRequested-bit feature can only be used through the RTE.

The request via the API `Dem_SetWIRStatus()` is not stored non-volatile, but a possible change of the UDS WIR-bits caused by this API is stored non-volatile. After the `Dem_Init()` call following an ECU restart, the WIR status must be requested again by the failsafe SWC.

After an ECU restart, the following two cases may occur that require a special handling:

- ▶ After a reset, the WIR-bit is still on, but only if it is requested internally by the indicator.

It must not be assumed that the failsafe request is still enabled because the WIR-bit is still on. If it is necessary to ensure that the WIR-bit remains on, even after the indicator may go off, the failsafe request must be renewed via the `Dem_SetWIRStatus()` API.

- ▶ The WIR-bit is only requested by the user via the `Dem_SetWIRStatus()` API. No indicator is ON or no indicator is associated with the event.

After a restart, the WIR-bit is still on. If it is no longer intended to keep active the former failsafe status ON, the WIR-bit must be reset by using the `Dem_SetWIRStatus()` API.

4.5.2.13. Support for BSW event reporting from multiple cores

This feature allows you to distribute the functionality of the API `Dem_ReportErrorStatus()` over multiple cores. Thus, other distributed BSW modules can use this API via the Schedule Manager (`SchM`) sender-receiver functionality of the RTE.

The `SchM` sender-receiver functionality is used for the inter-core communication of `Dem` reports. The `SchM` is only initialized after the `Dem` is pre-initialized. If reports are triggered during `Dem` pre-initialization, event reports are stored in the local buffer on a `Dem` satellite instance.

With the `Dem` configuration parameter `DemSatelliteInstanceLocalBswEventBufferSize`, you define the size of the local buffer available for each `Dem` satellite instance. After the `Dem` is initialized, the events stored in the internal queue are sent automatically using a data-received-event, and are processed on the master instance.

When the ECU is in GoSleep phase, `Dem` events can still be reported. When the ECU is in Sleep mode, no `Dem` events can be reported. In order for `Dem` reports to be triggered, the `EcuM` needs to be in WakeUp phase. The `SchM` needs to be activated before any reports can be triggered by the satellite instance to the master instance.

Feature availability

The `Dem` BSW distribution depends on the multi-core configuration of the OS. As a mandatory precondition, the project must integrate and configure the OS stack with multi-core support. To enable multi-core support, the parameter `OsNumberOfCores` must be configured to a value higher than 1.



Only if `OsNumberOfCores` is set to a value higher than 1, is the `Dem` BSW distribution feature with all associated configuration parameters available.

Moreover, `OsNumberOfCores` defines the BSW instances, each having a `BswImplementation` associated to it. For example, if `OsNumberOfCores` is configured to 3, there are three `BswImplementation`s: one master instance and two satellite instances for a single `Dem` configuration.

In addition to the `Os` module, the BSW distribution functionality depends on the existence of the `Rte` module. The data transfer between the master and satellite instances is realized with the help of the `SchM`, which is integrated into the `Rte`.

For information on how to configure these three modules in order to enable BSW event reporting from multiple cores, see [Section 4.5.3.4, “Enabling BSW event reporting from multiple cores”](#).

For information on integration restrictions for the `Dem` BSW distribution functionality, see [Dem integration issues](#).

4.5.2.14. Variant handling

The `Dem` module provides the following features for variant handling:

- ▶ [Section 4.5.2.14.1, “Calibration”](#)
- ▶ [Section 4.5.2.14.2, “Dynamic event availability”](#)
- ▶ [Section 4.5.2.14.3, “Side allocation”](#)
- ▶ [Section 4.5.2.14.4, “DTC suppression”](#)

4.5.2.14.1. Calibration

In the following, it is shown how calibration can be used to support variant handling. For an introduction to calibration, see [Section 4.5.2.6.1, “Calibration”](#).

- ▶ Use-case: Hardware-specific ECU support

An ECU and its software support different kinds of hardware. To assure the correct behavior with a specific hardware variant, a post-build configuration with a hardware-specific calibration set is applied via the MCD tool.

Example:

An ECU software supports multiple hardware variants. For hardware parts that are not included in a hardware variant, the related events need to be disabled. This means the corresponding `DemEventAvailable` parameter needs to be calibrated to `FALSE`.



4.5.2.14.2. Dynamic event availability

This feature allows you to change the configured event available status dynamically at run-time via the [`Dem_SetEventAvailable\(\)`](#) API. The event available status is stored volatile. That means that the desired status needs to be set again within each power cycle.

The dynamic event availability feature can be enabled with configuration parameter [`DemAvailabilitySupport`](#).

- ▶ Use-case: Hardware-specific ECU support

An ECU software supports multiple hardware variants. When a specific hardware variant is changed during production, or even at run-time by the garage, the software adapts to the hardware changes. Based e.g. on coding pin levels or EEPROM flags, some coding bits are written to the ECU, or some network message is received to change the software behavior. A software component for variant handling calls [`Dem_SetEventAvailable\(\)`](#) to set the desired event available status for variant-specific events at run-time.

4.5.2.14.3. Side allocation

The `Dem` module supports the configuration of a callout function to be called every time the `Dem` needs to retrieve a DTC for an event.

This feature can be used by an application component to change the statically configured DTCs of events, e. g. by adding a side-specific constant offset to affected DTCs.

The side allocation feature serves as an alternative approach to the AUTOSAR *post-build selectable* variant, which is not supported by the `Dem` module.

The side allocation feature can be enabled with configuration parameter [`DemCalloutDynamicDTCFnc`](#).

- ▶ Use-case: ECUs with side allocation

ECUs with side allocation (e.g. left/right variation) are flashed with the same software but the diagnostic ECUID must differ for the left and right variant at run-time. The configured [`DemCalloutDynamicDTCFnc\(\)`](#) callback function is called for obtaining the alternative DTC, based on the configured DTC value and/or the EventId. The function reads e.g. coding pin levels or EEPROM flags for selecting the left/right variant and performs the desired DTC translation accordingly.

4.5.2.14.4. DTC suppression

This feature allows you to suppress/unsuppress a DTC at run-time via [`Dem_SetDTCSuppression\(\)`](#) API. A suppressed DTC is not visible for the tester but does not stop the event processing.

The DTC suppression feature can be enabled with configuration parameter [`DemSuppressionSupport`](#).



- ▶ Use-case: Hide DTCs at start-up phase

An ECU software supports multiple hardware variants. For hardware parts that are not included in a hardware variant, the monitor shall be deactivated and the related DTCs shall not be visible to the tester.

Based e.g. on coding pin levels or EEPROM flags, a software component for variant handling deactivates the monitor and calls [`Dem_SetDTCSuppression\(\)`](#) to suppress the dependent DTCs. The dependent DTCs need to be suppressed again in each power cycle, immediately after [`Dem_Init\(\)`](#). A deactivated monitor shall report the event status PASSED to the `Dem` module.

- ▶ Use-case: Suppress/unsuppress DTCs dependent on the diagnostic session

On entering the development session, an application can use [`Dem_SetDTCSuppression\(\)`](#) to enable all suppressed DTCs that should only be visible in the development session.

On leaving the development session, the application suppresses all DTCs that should not be visible outside of the development session.

4.5.2.15. Dcm-Dem OBD interface for distance traveled PIDs (\$21/\$31)

The `Dem` provides internal calculation of the following PIDs related to distance traveled:

- ▶ PID \$21 *distance traveled while MIL is activated*
- ▶ PID \$31 *distance traveled since DTCs cleared*

The calculation is offered if the data layout is compliant to:

- ▶ SAE J1979-DA (revised 2014-06)
- ▶ AUTOSAR 4.2.2 (PID \$21)
- ▶ AUTOSAR 4.1.2 (PID \$31)

The distance traveled of the PID \$21/\$31 is updated inside the [`Dem_MainFunction\(\)`](#) once every second as stipulated by the scaling of PID \$1F. Measured in kilometers, the distance traveled data has the following range:

- ▶ The minimum value is 0. This value is also used if the internal PID \$21/\$31 calculation is not configured.
- ▶ The maximum value is 65535 (0xFFFF), with no possibility to overflow the data. Thus, the upper range of the distance traveled information represents 65535 km or more since the last reset.

The distance traveled data is stored non-volatile in the memory block `DEM_NVM_BLOCK_ID_DEFAULT` together with the additional data for the reset of PID \$21.

The processing of PID \$21:

- ▶ starts as soon as the MIL indicator gets activated.
- ▶ resets under all of the following circumstances:



- ▶ The warm-up cycle (refer to chapter [Section 4.5.2.4.2, “OBD warm-up cycle \(WUC\)”](#)) is used to reset the distance traveled if at least 40 warm-up cycles elapse after the MIL was deactivated.
- ▶ Clear/reset emission-related diagnostic information via service \$04 is requested.
- ▶ The MIL indicator changes from *deactivated* to *activated*.

The processing of PID \$31:

- ▶ starts after each [Dem_ClearDTC\(\)](#) and after an initial software update.
- ▶ resets if a request to clear diagnostic information via service \$04 or \$14 is received.

Feature availability

- ▶ Common preconditions:
 - ▶ The centralized handling of the PID must be disabled to allow the `Dem`-internal PID calculation:
 - ▶ Centralized PID \$21 handling: [DemOBDCentralizedPID21Handling](#)
 - ▶ Centralized PID \$31 handling: [DemOBDCentralizedPID31Handling](#)
 - ▶ The calculation of the distance traveled is based on
 - ▶ the input vehicle speed - PID \$0D ([DemOBDInputVehicleSpeed](#))
 - ▶ time since engine start - PID \$1F ([DemOBDTimesinceEngineStart](#))
 Both need to be configured.
- ▶ Additionally, for PID \$21 configure:
 - ▶ the MIL indicator ([DemMILIndicatorRef](#))
 - ▶ the operation cycle `DEM_OPCYC_WARMUP` as described in chapter [Section 4.5.2.4.2, “OBD warm-up cycle \(WUC\)”](#). Otherwise the reset trigger is not available.
- ▶ The value locally calculated by the `Dem` can be provided to the `Dcm` (parameter [DcmDspPidDataReadFnc](#) for service \$01) via the following APIs:
 - ▶ PID \$21: [Dem_DcmReadDataOfPID21\(\)](#).
 - ▶ PID \$31: [Dem_DcmReadDataOfPID31\(\)](#).

4.5.2.16. Parallel event memory access

To enable parallel access to the event memory, the `Dem` uses the concept of diagnostic clients. A diagnostic client in the EB `Dem` solution is defined as a BSW (e.g. `Dcm`) module that accesses the `Dem` module in order to delegate diagnostic services related to the event memory. You can configure several diagnostic clients using the container [DemClient](#) and associate each of them with a unique ClientId via the parameter [DemClientId](#).

To access the event memory, the `Dem` offers several APIs that support the ClientId as a parameter. When calling these APIs, you pass the ClientId assigned to the respective client as the argument. The APIs use the passed



ClientId to distinguish one diagnostic client from another and handle the respective operations independent of each other. This way, the `Dem` allows access to the event memory from different clients in parallel.

In the EB `Dem` solution, all configured diagnostic clients are treated equally. For example, it is not possible to restrict a certain client to access a certain event memory.

- ▶ Use case: Parallel processing of OBD and UDS services

Parallel processing of OBD and UDS services can be achieved by configuring two diagnostic clients dedicated to each of the protocols. The `Dcm` can then pass the appropriate ClientId when calling the `Dem` APIs to process a service request.

Currently, this is the main and the only supported use case. For example, processing of UDS in parallel to UDS is not yet supported. This is because not all the APIs used by `Dcm` to implement the UDS services support multiple clients in parallel.

WARNING
Parallel access by the same diagnostic client


Do not use the same ClientId to make multiple calls to the `Dem` APIs concurrently. The APIs that support ClientId as a parameter are reentrant only for different ClientIds but not for the same ClientId. Besides, the EB `Dem` solution assumes that different APIs are called only sequentially for a given client. If you use the **same** ClientId to make more than one call to the `Dem` APIs concurrently, the behavior is undefined.

4.5.2.17. J1939 support

The `Dem` provides the feature of storing and reading J1939 DTCs and J1939 freeze frames.

The J1939 support is implemented according to:

- ▶ SAE J1939 (revised 2016-73)
- ▶ AUTOSAR 4.2.2

The J1939 support allows you to configure [J1939 DTC](#) values for the DTCs. It also provides a filtering mechanism to read these DTCs. The following applies:

- ▶ If [OBD](#) is disabled
 - ▶ DM5: Diagnostic Readiness 1
 - ▶ DM6: Emission-Related Pending Diagnostic Trouble Codes
 - ▶ DM12: Emission-Related MIL-On Diagnostic Trouble Codes
 - ▶ DM23: Emission-Related Previously MIL-On Diagnostic Trouble Codes
 - ▶ DM28: Emission-Related Permanent Diagnostic Trouble Codes.



will not be available

- ▶ Any DTC that has no UDS or OBD value is only visible through the Read DTC mechanism provided by the J1939 support.
- ▶ Any emission-related J1939 DTC must reference a [MIL](#). For the DTC to be able to reference a MIL, the DTC must have an OBD value.

Clearing a J1939 DTC with J1939 parameters is only possible via the J1939 support. You cannot use the J1939 parameters in any generic API. However, Clear J1939 DTC clears the entire DTC data, incl. the status byte and the event memory entry. See [DemJ1939ClearDTCsupport](#).

The J1939 freeze frames ([DemJ1939FreezeFrameClassRef](#)) are retrieved according to the order of the events that match the filter. Within each event, the SPNs ([Suspect Parameter Numbers](#)) are retrieved in the order configured within the freeze frame.

Mandatory SPNs have to be configured first for each freeze frame with the same order as specified in the SAE.

4.5.2.18. Reset of Dem NvM data after an incompatible configuration update

A software update with a faulty `Dem` configuration might lead to a false NvM default memory. In the worst case, such NvM inconsistencies could cause an ECU crash due to an invalid pointer access in `Dem` during processing of the restored NvM data.

To avoid faulty NvM data, a mechanism identifies any update of the `Dem` configuration that might impact the structure and size of the stored NvM data. For this, configuration signatures are stored both within the NvM data and the current program code. Both signatures are compared in [Dem_Init\(\)](#) during the start of the module. If the two signatures are different, the `Dem` NvM data is reinitialized. All previous data is lost.

This feature is optional and can be enabled with the configuration parameter [DemNvDataConfigSignatureUsed](#).

4.5.2.19. Considerations for vehicle production

During particular vehicle production phases, malfunctions might be detected and the fault codes, including permanent fault codes, might unintentionally be stored in NVRAM.

Fault codes can be cleared via service requests \$04 or \$14, e.g. at end-of-line. However, the `Dem` module does not provide interfaces to clear permanent fault codes. According to the emission regulations, permanent fault codes must not be cleared until the OBD system itself confirms it.

The following sub-sections describe a `Dem`-external mechanism to:

- ▶ [Inhibit the storage of permanent faults during production](#)



► [Reset the permanent memory at end-of-line](#)

Both mechanisms are based on integration code that directly interacts with NvM APIs under certain conditions during production.

For more information on the NvM module, see the EB tresos AutoCore Generic Memory Stack documentation.

For NvM block references, see the [DemNvRamBlockId](#) parameter description.

WARNING
Risk of legal non-compliance


Although the Dem-external intervention on permanent memory could be used during any phase, the integrator must ensure the legal usage, e.g. clear only during vehicle production. The Dem has no possibility to supervise such conditions.

4.5.2.19.1. Inhibit the storage of permanent faults during production

For this use case, the integration code can write-protect the NV block `DEM_NVM_BLOCK_ID_PERMANENT` to prevent the storage of permanent fault codes during vehicle production. This can be achieved by calling `NvM_SetBlockProtection()` before [Dem_Init\(\)](#) on ECU start-up.

It is assumed that the ECU is powered off when the vehicle leaves the production line.

4.5.2.19.2. Reset the permanent memory at end-of-line

For this use case, the integration code can overwrite the NV block `DEM_NVM_BLOCK_ID_PERMANENT` with reset data at end-of-line. This can be achieved by calling `NvM_WriteBlock()` with a pointer to an explicit temporary RAM data block. The size of the RAM data block must be equal to the configured NV block size and the data needs to be initialized with zero.

Example: `unit8 PMResetData[DEM_MAX_NUMBER_EVENT_ENTRY_PER * 2U] = {0}`

`DEM_MAX_NUMBER_EVENT_ENTRY_PER` is available via `Dem.h`.

To explicitly synchronize the Dem-internal permanent event memory with the data of the overwritten/cleared NV block, the integration code needs to call `NvM_ReadBlock()` with `NULL_PTR` as RAM block address.

`NvM_WriteBlock()` and `NvM_ReadBlock()` must not be called when the NV block is used by the Dem, see [Dem integration issue](#). Therefore, the following sequence needs to be executed:

1. [Dem_Shutdown\(\)](#)
2. `NvM_WriteBlock()`
3. `NvM_ReadBlock()`



4. [Dem_Init\(\)](#)

The following applies:

- ▶ If the sequence is executed synchronously, the cyclic activation of Memory Stack tasks is needed while waiting for the job processing, e.g. by calling the [_DemCallbackMemStackMainFuncTrigger\(\)](#) callout function.
- ▶ A single block NvM operation must be started only if no job is pending for that NvM block. [NvM_GetErrorStatus\(\)](#) can be used to check that no operation for NV block [DEM_NVM_BLOCK_ID_PERMANENT](#) is pending before calling [NvM_WriteBlock\(\)](#) or [NvM_ReadBlock\(\)](#).
- ▶ You must ensure that no `Dem` API that requires a fully initialized `Dem` is called while the sequence is processed (refer to `Dem` error [DEM_E_UNINIT](#)).

4.5.3. Configuring the Dem module

4.5.3.1. BSW Mode Manager (BswM)

Configure the BSW Mode Manager to trigger the initialization handling of the `Dem`. For details, see the EB tresos AutoCore Generic Mode Management documentation.

4.5.3.2. Basic Software Scheduler (Rte)

- ▶ Configure the `Rte` to support one exclusive area of the Basic Software Module instance `Dem`. Make sure that the configured exclusive area is named `SCHM DEM EXCLUSIVE AREA 0`.
- ▶ Additionally, use the `Rte` to call the [_Dem_MainFunction\(\)](#) periodically.
- ▶ The `Rte` must use the same period as configured for the parameter `DemGeneral/DemTaskTime`.

4.5.3.3. NVRAM Manager (NvM)

You can configure the `Dem` to store event-related data in the following ways:

- ▶ volatile
- ▶ non-volatile without immediate storage
- ▶ non-volatile with immediate storage

For the non-volatile storage of event memory entries, `Dem` uses non-volatile memory (NvM) blocks.



In Dem, you need to configure an NvM block reference, the [DemNvRamBlockId](#), for every NvM block required. Based on the configuration of these references, all required NvM blocks are then automatically configured and referenced in the NVRAM Manager module.

For configuring the NvM block references, see the [DemNvRamBlockId](#) parameter description.

If the permanent storage of event memory entries is required, the NvM configuration must include the header file `Dem.h`. For this, add `Dem.h` to the configuration parameter `NvMUserHeader`.

For detailed information concerning NvM, see the EB tresos AutoCore Generic Memory Stack documentation.

4.5.3.4. Enabling BSW event reporting from multiple cores

To enable BSW event reporting from multiple cores, you have to configure the modules Dem, Os and Rte.

4.5.3.4.1. Configuring the Dem



Configuring the Dem

Prerequisite:

- In the Os module, the `OsNumberOfCores` parameter is set to a value higher than 1. Only then all Dem BSW distribution parameters are available.

Step 1

In the Dem editor, go to the **BSW Distribution** tab.

Step 2

To enable distributed BSW event reporting, enable [DemDistributedBSWEEventReporting](#).

Step 3

Configure the Dem master core ID in [DemMasterCoreId](#), e.g. `DemMasterCoreId: 1`.

`DemMasterCoreId` is the core where the main Dem instance runs. This master instance of the integrated Dem module has all the functionality available.

Step 4

Configure the sender-receiver queue length for BSW event reports received on the Dem master instance in [DemSRQueueLengthDistributedBSWEEventReporting](#).

Step 5

Configure the list of satellite instances in [DemSatelliteInstanceConfig](#). For example, for two satellite instances, configure as follows:

For `DemSatelliteInstanceConfig_0`



- ▶ Configure the core ID of the satellite in [DemSatelliteCoreId](#), e.g. DemSatelliteCoreId: 0. A satellite instance refers to all other cores beside the master instance.
- ▶ Configure the local buffer size for the `Dem` satellite instance in [DemSatelliteInstanceLocalBsw-EventBufferSize](#).

For `DemSatelliteInstanceConfig_1`

- ▶ Configure the core ID of the satellite in [DemSatelliteCoreId](#), e.g. DemSatelliteCoreId: 2. A satellite instance refers to all other cores beside the master instance.
- ▶ Configure the local buffer size for the `Dem` satellite instance in [DemSatelliteInstanceLocalBsw-EventBufferSize](#).

NOTE



Ensure core ID consistency

Ensure consistency between [DemMasterCoreId](#) and [DemSatelliteCoreId](#). If the master instance p (`DemMasterCoreId`) is mapped to ID value 1 and the satellite instances (`DemSatelliteCoreId`) are mapped to ID value 0 and ID value 2, you must use the same configuration when you map the `BswImplementations` in the `Rte`.

4.5.3.4.2. Configuring the Os



Configuring the Os

Step 1

On the **OsOS** tab, configure `OsNumberOfCores` to a value greater than 1.

Step 2

Depending on the value entered for `OsNumberOfCores`, configure the same number of `OsApplication` parameters, each corresponding to an instance. For 3 configured cores, the following `OsApplications` exist:

- ▶ `SystemPartition`
- ▶ `Partition1`
- ▶ `Partition2`

Step 3

In the **OsTask** tab, create the required `OsTasks`.

Step 4

Via each `OsApplication/OsAppTaskRef`, assign each task to a partition.

Step 5

For each `OsTask`, configure a `OsTaskAccessingApplication`.

**NOTE****Avoid Os time-out**

The task that is configured with the event `DataReceivedEvent_ReceiveBswEventReportsOnMaster` needs to have the parameter `OsTaskActivation` set to a value high enough so that the OS does not time out as soon as `Dem` is in INITIALIZED state. This can occur due to a high number of possible reports in the `Dem` pre-init phase. Consequently, there might be no valid reports or the OS might issue the error *ID 4 – activation limit reached*.

4.5.3.4.3. Configuring the Rte

The following configuration steps are performed in the **Rte Generic Editor**. You can use the **Rte Editor** for a more comprehensive view on the configured data. Moreover, the **Rte Editor** allows you an easier event-to-task mapping via the **Event Mapping** tab.



Configuring the Rte

Prerequisite:

- The BSWMDs were properly generated and imported.

Step 1

Open the **Rte Generic Editor**.

Step 2

In **BSW Module Instances** tab, create BSW module instances for each instance on which the module should run.

Step 3

Map each module instance to an `OsPartition`. For example, if there are 3 cores configured, you create and map 3 BSW module instances as follows:

- ▶ BSW instance for master core: `BSW_Dem` mapped at SystemPartition (ID 1)
- ▶ BSW instance for first satellite core: `BSW_Dem_1_Core_0` mapped at Partition1 (ID 0)
- ▶ BSW instance for second satellite core: `BSW_Dem_1_Core_2` mapped at Partition2 (ID 2)

Step 4

For each of the `RteBswModuleInstance`, configure the list of **BSW Event to Task Mapping**.

Step 4.1

For `BSW_Dem` (master core):

- ▶ Map an event to `DataReceivedEvent_ReceiveBswEventReportsOnMaster`.
- ▶ As usual, map an event to `TimingEvent_MainFunction`.

Step 4.2

For `BSW_Dem_1_Core_0` (first satellite instance):



- ▶ Map an event to `BswExternalTriggerOccurredEvent_Dem_Init`

Step 4.3

For `BSW_Dem_1_Core_2` (second satellite instance):

- ▶ Map an event to `BswExternalTriggerOccurredEvent_Dem_Init`

Step 5

For all events mentioned, select an appropriate Os task in the event configuration.

Step 6

In **RteBswRequiredSenderReceiverConnection**, configure the sender-receiver communication between satellites and master based on the following port connection information.

	Master instance
<code>RteBswRequiredVariableDataPrototypeRef</code>	<code>RteBswRequiredSenderReceiverConnection_0</code>
<code>RteBswRequiredVariableDataPrototypeRef</code>	<code>RteBswRequiredSenderReceiverConnection_1</code>

Table 4.1. Port connection for the master

Each connection on the master instance references a:

- ▶ `providedData`: `ReportErrorStatusProvide`
- ▶ `requiredData`: `ReportErrorStatusRequire`

Each master port needs to be connected to its corresponding satellite port.

	RteBswRequiredSender- ReceiverConnection_0	RteBswRequiredSender- ReceiverConnection_1
<code>RteBswProvidedVariableDataProto-</code> <code>typeRef</code>	<code>SetReportErrorStatu-</code> <code>sOnSatelliteInstance</code>	<code>SetReportErrorStatu-</code> <code>sOnSatelliteInstance</code>
<code>RteBswRequiredVariableDataProto-</code> <code>typeRef</code>	<code>GetReportErrorStatus-</code> <code>FromSatelliteInstance</code>	<code>GetReportErrorStatus-</code> <code>FromSatelliteInstance</code>

Table 4.2. Port connection for each satellite

4.6. FIM module user guide

4.6.1. Overview

This chapter provides you with FIM-specific information:



- ▶ [Section 4.6.2, “Background information”](#) explains the concepts of the `FiM` module.
- ▶ [Section 4.6.3, “Configuring the FiM module”](#) provides instructions on how to configure the `FiM`.
- ▶ [Section 4.6.4, “Calibration support for the FiM module”](#) provides information about calibration capabilities of the `FiM`.

4.6.2. Background information

The Function Inhibition Manager `FiM` is responsible for providing a control mechanism for software components and the functionality therein. The SWCs and BSWs call it to retrieve function (FID) states.

By means of the `FiM`, you can configure inhibiting (deactivation of application function) functionalities.

4.6.2.1. Function Inhibition

The `FiM` sets its FID permission to deactivate the application function. A function (FID) only has permission if all assigned event-status checks permit this function. An event-status check will only inhibit (not permit) a function, if all masked bits are equal to their expected/configured values.

The `FiM` is closely related to the `Dem` since diagnostic events and their status information are supported as inhibit conditions. If a failure is detected and the event is reported to the `Dem`, the `FiM` then inhibits the FID and therefore the corresponding functionality.

The `FiM` recalculates the function permission states either on request or when a configuration-based event change occurs:

- ▶ On request

The `FiM` implementation shall recalculate the function inhibition/permission every time when the status of a FID is requested by a software component initiated by the API [`FiM_GetFunctionPermission\(FID, permission\[true/false\]\)`](#).

- ▶ On event status change

The `Dem` triggers the API [`FiM_DemTriggerOnEventStatus\(\)`](#) on event status change and it calculates the function permission of associated FIDs by comparing it with the configured inhibition masks.

4.6.2.2. Initialization

To enter the initialized state, the ECU State Manager typically calls [`FiM_Init\(\)`](#) within the start-up task. From then on, the `FiM` is able to receive FID permission requests from the BSW modules by the API function [`FiM_`](#)



`GetFunctionPermission(FID, permission)`. `FiM_DemInit()` is provided to `Dem` to re-initialize the `FiM` module in case the `Dem` detects a status change of a certain number of events.

4.6.3. Configuring the FiM module

For details about configuration parameters, see the *ACG8 Diagnostic Stack module references* chapter.

4.6.3.1. ECU State Manager (EcuM)

Configure the ECU State Manager to trigger the initialization handling of the `FiM`. For details about the ECU State Manager, see the EB tresos AutoCore Generic Mode Management documentation.

4.6.3.2. Diagnostic Event Manager (Dem)

The `Dem` triggers the API `FiM_DemTriggerOnEventStatus()` on event status change to `FiM`.

4.6.3.3. Basic Software Scheduler (Rte)

Configure the `Rte` to support one exclusive area of the Basic Software Module instance `FiM`. Make sure that the configured exclusive area is named `SCHM_FIM_EXCLUSIVE_AREA_0`.

4.6.4. Calibration support for the FiM module

The `FiM` module supports calibration of selected configuration parameters.

To enable calibration support, use the configuration parameter `FiMExtendedSupport` and set `FiMExtendedLevel` to extension level 1 or 2.

An A2L file is provided with all relevant information about the calibration parameter. The A2L file is located in the `output/A2L` folder after the generating step.

4.6.4.1. Calibration parameter

Calibration parameters are organized in an array with an entry size of 16 bits. The lower 14 bits hold the event ID and bits 14 and 15 the inhibition mask.



► Event ID ([FiMIinhEventId](#)):

This parameter has a size of 14 bits and an offset of 0.

► Inhibition mask ([FiMIinhInhibitionMask](#)):

This parameter has a size of 2 bits and an offset of 14.



5. ACG8 Diagnostic Stack module references

5.1. Overview

This chapter provides module references for the ACG8 Diagnostic 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 Diagnostic 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. Dcm

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.
DcmConfigSet	1..1	Label: DCM Configuration Defines one configuration set of the <code>Dcm</code> . This container and its sub-containers exist once per configuration set.
PublishedInformation	1..1	Label: EB Published Information Additional published parameters not covered by Common-PublishedInformation container.

Parameters included	
Parameter name	Multiplicity
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT
Label	Configuration Variant
Multiplicity	1..1
Type	ENUMERATION



Default value	VariantPreCompile	
Range	VariantPreCompile	
Configuration class	VariantPreCompile:	VariantPreCompile

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	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	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	5
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	10
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	53
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	

5.2.1.2. DcmConfigSet

Containers included		
Container name	Multiplicity	Description
DcmDefensiveProgramming	1..1	<p>Label: Defensive Programming Options</p> <p>Parameters for defensive programming</p>
DcmDsd	1..1	<p>Label: Diagnostic Service Dispatcher</p> <p>Defines the configuration of the Diagnostic Service Dispatcher (DSD) submodule of the <code>Dcm</code>.</p> <p>The DSD submodule checks the validity of an incoming request and keeps track of the progress of the service request execution.</p> <p>Note1: Configure at least one <code>DcmDsdServiceTable</code>.</p> <p>Note2: One <code>DcmDsdServiceTable</code> may be referenced by more than one <code>DcmDslProtocol</code>.</p>
DcmDsl	1..1	<p>Label: Diagnostic Session Layer</p> <p>Defines the configuration for the Diagnostic Session Layer(DSL) submodule of the <code>Dcm</code>. <P>Parameters in this container define the configuration for:</p> <ul style="list-style-type: none"> ▶ Protocols and protocol configuration including: <ul style="list-style-type: none"> ▶ Protocol priority ▶ Protocol preemption ▶ Timing ▶ Reception Channels, request types (Physical and Functional), Network Identifiers and <code>RxPduId</code> configuration ▶ Connections, <code>TesterSourceAddresses</code> <code>TxConfirmationPduId</code> and <code>TxPduId</code> configuration ▶ Reception and transmission buffers



Containers included		
		<p>► ServiceRequestNotification interface port naming</p> <p>Note: There shall be exactly one DSL configuration for DCM.</p>
DcmDsp	1..1	<p>Label: Diagnostic Service Processing</p> <p>Defines the configuration of the Diagnostic Service Dispatcher (DSP) submodule of the Dcm.</p> <p>The DSP submodule analyzes the received Diagnostic Service Request messages, performs consistency checks, acquires data or executes the required function calls in order to execute the service request received and assembles the response.</p>
DcmGeneral	1..1	<p>Label: General Configuration Parameters</p> <p>Defines the general configuration of the Dcm.</p>
DcmPageBufferCfg	1..1	<p>Label: Paged Buffering Configuration</p> <p>Defines the configuration controlling the functionality of the Paged Buffer transmission mechanism.</p>
DcmProcessingConditions	0..1	<p>Label: Processing Conditions</p> <p>Defines the configuration for the mode arbitration functionality of the Dcm.</p>

5.2.1.3. DcmDefensiveProgramming

Parameters included	
Parameter name	Multiplicity
DcmDefProgEnabled	1..1
DcmPrecondAssertEnabled	1..1
DcmPostcondAssertEnabled	1..1
DcmStaticAssertEnabled	1..1
DcmUnreachAssertEnabled	1..1
DcmInvariantAssertEnabled	1..1
Parameter Name	DcmDefProgEnabled
Label	Enable Defensive Programming



Description	Enables or disables the defensive programming feature for the module Dcm. Note: This feature is dependent on the use of the development error detection module. To use the defensive programming feature, proceed as follows: <ol style="list-style-type: none">1. Enable development error detection2. Enable defensive programming3. Enable assertions as required
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmPrecondAssertEnabled
Label	Enable Precondition Assertions
Description	Enables handling of precondition assertion checks reported from the module Dcm. Dependency on parameter(s): <ul style="list-style-type: none">▶ Enable Development Error Detection (<code>DcmDevErrorDetect</code>): must be enabled▶ Enable Defensive Programming (<code>DcmDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmPostcondAssertEnabled
Label	Enable Postcondition Assertions
Description	Enables handling of postcondition assertion checks reported from the module Dcm. Dependency on parameter(s): <ul style="list-style-type: none">▶ Enable Development Error Detection (<code>DcmDevErrorDetect</code>): must be enabled



	<ul style="list-style-type: none"> ▶ Enable Defensive Programming (<code>DcmDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH
Parameter Name	DcmStaticAssertEnabled
Label	Enable Static Assertions
Description	<p>Enables handling of static assertion checks reported from the module Dcm.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Enable Development Error Detection (<code>DcmDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>DcmDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH
Parameter Name	DcmUnreachAssertEnabled
Label	Enable Unreachable Code Assertions
Description	<p>Enables handling of unreachable code assertion checks reported from the module Dcm.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Enable Development Error Detection (<code>DcmDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>DcmDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH



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

5.2.1.4. DcmDsd

Containers included		
Container name	Multiplicity	Description
DcmDsdServiceTable	1..256	<p>Defines the parameters corresponding to every diagnostic service table per row.</p> <p>Configuration of the services to which a protocol of the <code>Dcm</code> will respond here.</p> <p>A <code>DcmDsdServiceTable</code> contains a set of services for which the protocol referencing the <code>DcmDsdServiceTable</code> will provide support.</p> <p>Note: The <code>Dcm</code> allows services to either use an internal implementation (for selected service IDs) or an external, user-supplied service handler.</p>

5.2.1.5. DcmDsdServiceTable

Containers included		
Container name	Multiplicity	Description



Containers included

DcmDsdService	1..n	<p>Label: Diagnostic Service configuration Defines the parameters corresponding to a diagnostic service handler.</p> <p>Note : Depending on the license, the Dcm additionally supports the Vendor specific service "Read Generic information" (0xAF):</p> <p>This service extends the functionality of the Read DTC Information service by allowing the request of all DTCs stored in memory at once.</p> <p>In order to correctly make use of this Service, The Dcm shall map SI30 as Extended Data record 0x30 to all events. If this is not the case, the related DTCs will not be listed on the service response.</p>
-------------------------------	------	---

Parameters included

Parameter name	Multiplicity
DcmDsdSidTabId	1..1

Parameter Name	DcmDsdSidTabId	
Label	Table ID	
Description	Defines the unique identifier of this Diagnostic Service Table. Range: ▶ 0 .. 255.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.6. DcmDsdService

Containers included

Container name	Multiplicity	Description
----------------	--------------	-------------



Containers included

DcmDsdSubService	0..n	<p>Label: SubService Configuration</p> <p>Defines the sub-function configuration for this DcmDsdService.</p> <p>Use this container in order to configure sub-function support for the internal service handler for this service ID.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Sub-function availability (DcmDsdSidTabSubfuncAvail): This container is enabled only if the DcmDsdSidTabSubfuncAvail parameter for this service is set to TRUE and the parameter DcmDsdSidTabFnc is not configured or parameter DcmDsdDisableGenericServiceImplementation is set to FALSE, i.e. the service handler implementation is internal (Generic Service Handling is enabled).
----------------------------------	------	---

Parameters included

Parameter name	Multiplicity
DcmAsyncServiceExecution	1..1
DcmDsdServiceUsed	1..1
DcmDsdSidTabFnc	0..1
DcmDsdDisableGenericServiceImplementation	1..1
DcmDsdSidTabServiceId	1..1
DcmDsdSidTabSubfuncAvail	1..1
DcmDsdSidTabModeRuleRef	0..1
DcmDsdSidTabSecurityLevelRef	0..n
DcmDsdSidTabSessionLevelRef	0..n

Parameter Name	DcmAsyncServiceExecution
Label	Enable Asynchronous service handler
Description	<p>Note : This is a Vendor Specific parameter for Asynchronous Service Handler usage in the Dcm.</p> <p>Enables Asynchronous Service Handling for requests to this diagnostic service.</p> <p>If Asynchronous Service Handling is enabled for a service, the request processing is performed in a different task from the task executing the Dcm_MainFunc-</p>



	<p><code>tion()</code>. This removes the restriction for the service handlers that they must return before the next call to the <code>Dcm_MainFunction()</code>, thus allowing functions with higher execution times to still be used.</p> <p>If this parameter is enabled for any service, please map the <code>BswInternalTriggerOccurredEvent</code> <code>BswInternalTriggerOccurredEvent_Async</code> to a task having a lower priority than the task running the <code>Dcm_MainFunction()</code>.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: Enables Asynchronous Service Handling for this service. ▶ FALSE: Disables Asynchronous Service Handling for this service.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDsdServiceUsed
Label	Enable Service usage
Description	Defines whether this Service is available for use in the current configuration (for multi-purpose ECUs). Range: <ul style="list-style-type: none"> ▶ TRUE: This Service is available. ▶ FALSE: This Service is not available. The <code>Dcm</code> behaves as if this Service was not configured.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDsdSidTabFnc
Label	Service Handler
Description	Defines the Callout function for the ECU Supplier specific service handler for this particular <code>DcmDsdSidTabServiceId</code> . This parameter is related to the interface <code>&#60;Module&#62;_&#60;DiagnosticService&#62;</code> .



	If an ECU Supplier specific service handler is needed for this <code>DcmDsdSidTabServiceId</code> , enable this parameter and configure the name of the function to be called here. Otherwise, an internal service handler is used, if available.
Multiplicity	0..1
Type	FUNCTION-NAME
Default value	<code>DcmDsdSidTabFnc</code>
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DcmDsdDisableGenericServiceImplementation
Label	Disable Generic Service Handler
Description	<p>This parameter controls the availability of the Generic Service Handler, when the External Service Handler option has been enabled.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: The associated Generic Service Handler for this service is not available. ▶ FALSE: The associated Generic Service Handler for this service is available. It can be called inside the External Service Handler.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDsdSidTabServiceId
Label	Service Identifier
Description	<p>Defines the identifier of this Diagnostic Service according to ISO 14229-1 and ISO 15031-5.</p> <p>Please refer to AUTOSAR DCM SWS(Document Version 4.2.0) - Table 4 and Table 5 for identifiers supported by AUTOSAR.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0x00 .. 0xFF.
Multiplicity	1..1



Type	INTEGER	
Default value	0	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDsdSidTabSubfuncAvail	
Label	Sub-Function Availability	
Description	<p>Defines whether the service includes sub-functions or it does not.</p> <p>Please configure this parameter in order to specify, for a certain service, if it offers sub-function support. When sub-function support is available for a service, the most significant bit of the second byte of the request will be interpreted as the <code>suppressPosRspMsgIndicationBit</code>.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: Sub-functions are available for this service ▶ FALSE: Sub-functions are not available for this service 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDsdSidTabModeRuleRef	
Label	Reference to ModeRule	
Description	<p>Reference to a <code>DcmDspModeRule</code> which conditions the execution of this service handler.</p> <p>If this parameter is enabled and a <code>DcmDspModeRule</code> is referenced, this Mode Rule shall be evaluated before service execution is allowed to start. If the Mode Rule evaluates to TRUE, service processing will be allowed to proceed. If the Mode Rule evaluates to FALSE, service processing is not allowed to proceed and depending on the referenced mode rule a Negative Response Code is returned to the Tester. If no <code>DcmDspModeRule</code> is referenced, no Mode Rule permission check is performed.</p>	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile



Origin	AUTOSAR_ECUC
Parameter Name	DcmDsdSidTabSecurityLevelRef
Label	Security Level References
Description	<p>Reference to the <code>DcmDspSecurityRows</code> defining the Security Levels in which the service shall be allowed to execute.</p> <p>Note: If there is no reference configured, no check of Security Level is performed, i.e. this service is allowed to run in any Security Level.</p>
Multiplicity	0..n
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC V1.0.0
Parameter Name	DcmDsdSidTabSessionLevelRef
Label	Session References
Description	<p>Reference to the <code>DcmDspSessionRows</code> defining the Diagnostic Sessions in which the service shall be allowed to execute.</p> <p>Note: If there is no reference configured, no check of Diagnostic Session is performed, i.e. this service is allowed to run in any Diagnostic Session.</p>
Multiplicity	0..n
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC V1.0.0

5.2.1.7. DcmDsdSubService

Parameters included	
Parameter name	Multiplicity
DcmDsdSubServiceFnc	0..1
DcmROEonDtcStatusChangeShortResponse	1..1
DcmDsdSubServiceId	1..1
DcmDsdSubServiceModeRuleRef	0..1
DcmDsdSubServiceSecurityLevelRef	0..n
DcmDsdSubServiceSessionLevelRef	0..n



Parameter Name	DcmDsdSubServiceFnc	
Label	Sub Service Handler	
Description	<p>Defines a Callout function for the ECU Supplier specific sub-function handler for this particular <code>DcmDsdSubServiceId</code>. This parameter is related to the interface <code>&#60;Module&#62;_&#60;DiagnosticService&#62;_&#60;SubService&#62;</code>.</p> <p>If an ECU Supplier specific sub-function handler is needed for this <code>DcmDs-dSubServiceId</code>, enable this parameter and configure the name of the function to be used here. Otherwise, an internal sub-function handler is used, if available.</p>	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Default value	DcmDsdSubServiceFnc	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmROEonDtcStatusChangeShortResponse	
Label	Enable short response of ResponseOnEvent SubService onDTCStatusChange	
Description	<p>Defines whether the Subservice onDTCStatusChange of Service ResponseOn-Event responds with short format.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: The Subservice responds shortly (only 4 bytes). ▶ FALSE: The Subservice responds according to ISO-14229. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmDsdSubServiceId	
Label	SubService Identifier	
Description	Defines the identifier of this sub-function. This identifier is unique in the context of this sub-function's parent service.	
Multiplicity	1..1	
Type	INTEGER	



Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDsdSubServiceModeRuleRef
Label	Reference to ModeRule
Description	<p>Reference to DcmDspModeRule Mode rule which controls execution of this the DcmDsdSubService. If there is no reference, no check of the mode rule shall be done.</p> <p>Reference to a DcmDspModeRule which conditions the execution of this sub-function.</p> <p>If this parameter is enabled and a DcmDspModeRule is referenced, this Mode Rule shall be evaluated before service execution is allowed to start. If the Mode Rule evaluates to TRUE, service processing will be allowed to proceed. If the Mode Rule evaluates to FALSE, service processing is not allowed to proceed and depending on the referenced mode rule a Negative Response Code is returned to the Tester. If no DcmDspModeRule is referenced, no Mode Rule permission check is performed.</p>
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDsdSubServiceSecurityLevelRef
Label	Reference to Security Levels
Description	<p>Reference to the DcmDspSecurityRow's defining the Security Levels in which the sub-function shall be allowed to execute.</p> <p>Note: If there is no reference configured, no check of Security Level is performed, i.e. this sub-function is allowed to run in any Security Level.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Parent service's Security Level references (DcmDsdSidTabSecurityLevelRef): Security Level references configured here must be a subset of the Security Level Reference configured for the parent service in DcmDsdSidTabSecurityLevelRef.
Multiplicity	0..n



Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDsdSubServiceSessionLevelRef	
Label	Reference to Session Levels	
Description	<p>Reference to the <code>DcmDspSessionRow</code>s defining the Diagnostic Sessions in which the sub-function shall be allowed to execute.</p> <p>Note: If there is no reference configured, no check of Diagnostic Session is performed, i.e. this sub-function is allowed to run in any Diagnostic Session.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Parent service's Diagnostic Session references (<code>DcmDsdSidTabSessionLevelRef</code>): Diagnostic Session references configured here must be a subset of the Diagnostic Session references configured for the parent service in <code>DcmDsdSidTabSessionLevelRef</code>. 	
Multiplicity	0..n	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.8. DcmDsl

Containers included		
Container name	Multiplicity	Description
DcmDslBuffer	1..256	<p>Defines the parameters for configuring Transmission and Reception Buffers which can further be referenced from:</p> <ul style="list-style-type: none"> ▶ The <code>DcmDslProtocolTxBufferID</code> Transmission buffer reference of any <code>DcmDslProtocol</code>. ▶ The <code>DcmDslProtocolRxBufferID</code> Reception buffer reference of any <code>DcmDslProtocolRx</code> channel of a <code>DcmDslMainConnection</code> belonging to a <code>DcmDslProtocol</code>.
DcmDslCallbackDCMRequestService	0..255	Defines the names of the Require Ports the <code>Dcm</code> generates for the interface <code>CallBackDCMRequestServices</code> .



Containers included		
		<p>The <code>CallBackDCMRequestServices</code> interface allows the Dcm to ask about prioritization of Diagnostic Protocols to the Application. The Application can further disallow to start a Diagnostic Protocol.</p> <p>The names of the containers in this list are used to define the name of the Require Ports generated to access the <code>CallBackDCMRequestServices</code> interface. The ports are named <code>CallBackDCMRequestServices_&#60;SWC&#62;</code>, where <code>&#60;SWC&#62;</code> is the name of the container <code>DcmDslCallbackDCMRequestService</code>.</p>
<u>DcmDslApplicationDiagnosticRequests</u>	0..1	<p>Label: Application Request Injector configuration</p> <p>Diagnostic requests coming from the Application are supported. The Application may generate a diagnostic service request on one of the Dcm's configured ReceptionPDUIDs by calling the interface:</p> <ul style="list-style-type: none"> ▶ <code>BufReq_ReturnType Dcm_-InjectDiagnosticRequest(PduldType RxPduld, const uint8 * RequestData PduLengthType RequestLength)</code> ▶ <code>RxPduld</code> : RxPduld on which the injected request is to be executed ▶ <code>*RequestData</code>: Pointer containing the injected request data ▶ <code>RequestLength</code>: Length of the injected request
<u>DcmDslDiagResp</u>	1..1	<p>Label: ResponsePending configuration</p> <p>Configures the handling of:</p> <ul style="list-style-type: none"> ▶ The <code>RequestCorrectlyReceivedResponsePending Negative Response Code</code> ▶ Requests which cannot be processed and are rejected by the DSL
<u>DcmDslProtocol</u>	1..1	<p>Label: Protocol Configuration</p> <p>Defines the configuration of the Diagnostic Protocols.</p> <p>Configuration for a Diagnostic Protocol contains:</p>



Containers included

		<ul style="list-style-type: none"> ▶ Configuration for the service table used ▶ Timing configuration defining timeouts and actions to be taken at timeout ▶ Buffer usage ▶ Communication channels including <code>TxPduId</code> and <code>TxConfirmationPduId</code> configuration ▶ Reception Connections for communication channels, including <code>RxPduId</code>, buffer and communication type settings ▶ Protocol Preemption and Protocol Priority settings <p>Note: at least one protocol must be configured.</p>
<u>DcmDslServiceRequestManufacturerNotification</u>	0..n	<p>Label: Manufacturer Notification</p> <p>Defines the names of the Require Ports the Dcm generates for the interface <code>ServiceRequestNotification</code>. for Manufacturer.</p> <p>The <code>ServiceRequestNotification</code> interface allows the Dcm to:</p> <ul style="list-style-type: none"> ▶ Inquire about permissions of servicing the received diagnostic service request to the Application, via the <code>xxx_Indication()</code> operation, which can choose to reject or accept the request. ▶ Notify the Application about the results of processing a diagnostic service request via the <code>xxx_Confirmation()</code> interface. <p>The names of the containers in this list are used to define the name of the Require Ports generated to access the <code>ServiceRequestNotification</code> interface. The R-Port is named <code>ServiceRequestManufacturerNotification_&#60;SWC&#62;</code> where <code>&#60;SWC&#62;</code> is the name of the container <code>DcmDslServiceRequestManufacturerNotification</code>.</p> <p>Note: This container is available only if the configuration parameter <code>DcmRequestManufacturerNotificationEnabled</code> is set to TRUE.</p>



Containers included

DcmDslServiceRequestSupplierNotification	0..n	<p>Label: Supplier Notification</p> <p>Defines the names of the Require Ports the Dcm generates for the interface ServiceRequestNotification. for Supplier.</p> <p>The ServiceRequestNotification interface allows the Dcm to:</p> <ul style="list-style-type: none"> ▶ Inquire about permissions of servicing the received diagnostic service request to the Application, via the <code>xxx_Indication()</code> operation, which can choose to reject or accept the request. ▶ Notify the Application about the results of processing a diagnostic service request via the <code>xxx_Confirmation()</code> interface. <p>The names of the containers in this list are used to define the name of the Require Ports generated to access the ServiceRequestNotification interface. The R-Port is named <code>ServiceRequestSupplierNotification_&#60;SWC&#62;</code> where <code>&#60;SWC&#62;</code> is the name of the container <code>DcmDslServiceRequestSupplierNotification</code>.</p> <p>Note: This container is available only if the configuration parameter <code>DcmRequestSupplierNotificationEnabled</code> is set to TRUE.</p>
--	------	---

Parameters included

Parameter name	Multiplicity
DcmDslEnableS3TimeoutNotification	1..1
DcmDslEnableTxConfirmationNotification	1..1

Parameter Name	DcmDslEnableS3TimeoutNotification
Label	Enable notification callout on S3 Timeouts
Description	<p>Allows the user to enable a callout to be notified when S3 timeout occurs.</p> <p>Signature of the notification callout: <code>void Dcm_S3TimeoutNotification (void).</code></p> <p>Range:</p>



	<ul style="list-style-type: none"> ▶ TRUE: Application is notified about S3 Timeouts. ▶ FALSE: Application is not notified about S3 Timeouts.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDslEnableTxConfirmationNotification
Label	Enable notification of transmission confirmation.
Description	<p>This switch enables or disables the notification of transmission confirmation.</p> <p>The signature of the notification callout is defined as follows:</p> <pre>void Dcm_ApplicationTransmissionConfirmation(Dcm_ServiceIDType ServiceID,Dcm_MsgContextPtrType MsgContext,Dcm_ResponseConfirmationType ResponseConfirmationType,boolean ResponseSuppressed,boolean TransmissionSuccessful)</pre> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: Enables the notification of transmission confirmation. ▶ FALSE: Disables the notification of transmission confirmation.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.2.1.9. DcmDslBuffer

Parameters included	
Parameter name	Multiplicity
DcmDslBufferSize	1..1
Parameter Name	DcmDslBufferSize
Label	Buffer Size



Description	<p>Defines the size of the Buffer.</p> <p>Note 1: If the Buffer is referenced as a Reception Buffer, the size must be configured large enough to fit the largest expected diagnostic service request message</p> <p>Note 2: If the Buffer is referenced as a Transmission Buffer and Paged Buffer Transmission is not to be used by services of the protocol using this Buffer, the size must be configured large enough to fit the largest expected diagnostic service response</p> <p>Note 3: If the Buffer is referenced as a Transmission Buffer and Paged Buffer Transmission is to be used by services of the protocol using this Buffer, the size must be configured large enough to fit the largest expected Page of data (subdivision of the diagnostic service response provided by the Application or another Basic Software Module).</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 8 .. 4294967294 Bytes.
Multiplicity	1..1
Type	INTEGER
Default value	8
Range	<p><=4294967294</p> <hr/> <p>>=8</p>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.2.1.10. DcmDslCallbackDCMRequestService

5.2.1.11. DcmDslApplicationDiagnosticRequests

Parameters included	
Parameter name	Multiplicity
DcmDslApplicationDiagnosticRequestBufferSize	1..1
Parameter Name	DcmDslApplicationDiagnosticRequestBufferSize



Label	Size of buffer used for Application-generated requests	
Description	Size of buffer used for Application-generated requests.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=4294967294 >=0	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

5.2.1.12. DcmDs1DiagResp

Parameters included	
Parameter name	Multiplicity
DcmDs1DiagRespMaxNumRespPend	1..1
DcmDs1DiagRespMaxNumRespPendInfinityLimit	1..1
DcmDs1DiagRespOnSecondDeclinedRequest	1..1

Parameter Name	DcmDs1DiagRespMaxNumRespPend
Label	Maximum number of ResponsePending messages
Description	<p>Configures the maximum number of consecutive RequestCorrectlyReceived-ResponsePending Negative Response Codes (NRCs) allowed for one diagnostic request.</p> <p>The Dcm shall send a GeneralReject (0x10) NRC when the number of RequestCorrectlyReceived-ResponsePending NRCs sent in the context of the current request processing has reached the limit defined by this parameter and if this parameter (DcmDs1DiagRespMaxNumRespPend) is configured to less than the value interpreted as an infinite number of RequestCorrectlyReceived-ResponsePending (NRCs). The value interpreted as infinity is given by the configuration parameter DcmDs1DiagRespMaxNumRespPendInfinityLimit:</p> <ul style="list-style-type: none"> ▶ 255 is interpreted as infinity if the configuration parameter DcmDs1DiagRespMaxNumRespPendInfinityLimit is set to "255". ▶ 65535 is interpreted as infinity if the configuration parameter DcmDs1DiagRespMaxNumRespPendInfinityLimit is set to "65535".



	If this parameter (<code>DcmDs1DiagRespMaxNumRespPend</code>) is configured to infinity limit (255 or 65535), there is no limit imposed on the maximum number of consecutive RequestCorrectlyReceived-ResponsePending NRCs allowed.
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU V1.0.0

Parameter Name	DcmDs1DiagRespMaxNumRespPendInfinityLimit	
Label	Infinity limit for RequestCorrectlyReceived-ResponsePending NRC	
Description	<p>Defines the value interpreted as infinite for the configuration parameters <code>DcmDspDiagSesRespMaxNumRespPend</code> and <code>DcmDs1DiagRespMaxNumRespPend</code>:</p> <ul style="list-style-type: none"> ▶ 255 - The range of possible values for the configuration parameters <code>DcmDspDiagSesRespMaxNumRespPend</code> (if enabled) and <code>DcmDs1DiagRespMaxNumRespPend</code> is 0 – 255 (inclusive) where the value 255 shall be used for transmitting an infinite number of RequestCorrectlyReceived-ResponsePending NRCs (0x78). ▶ 65535 - The range of possible values for the configuration parameters <code>DcmDspDiagSesRespMaxNumRespPend</code> (if enabled) and <code>DcmDs1DiagRespMaxNumRespPend</code> is 0 – 65535 (inclusive) where the value 65535 shall be used for transmitting an infinite number of RequestCorrectlyReceived-ResponsePending NRCs (0x78). 	
Multiplicity	1..1	
Type	INTEGER	
Default value	255	
Range	255 65535	
Configuration class	VariantPreCompile: VariantPreCompile	
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmDs1DiagRespOnSecondDeclinedRequest	
Label	Reaction upon a second request	
Description	Defines the reaction upon a second request (e.g. from another client) that cannot be processed (e.g. due to priority assessment or lack of a free transmission buffer).	



	<p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: when the second request cannot be processed, the Dcm shall respond to this request with a <code>BusyRepeatRequest</code> (0x21) Negative Response Code. ▶ FALSE: when the second request cannot be processed, no response shall be given.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU V1.0.0

5.2.1.13. DcmDslProtocol

Containers included		
Container name	Multiplicity	Description
DcmDslProtocolRow	1..256	Defines the configuration parameters belonging to different configured protocols.

5.2.1.14. DcmDslProtocolRow

Containers included		
Container name	Multiplicity	Description
DcmDslConnection	1..n	<p>Defines the connection between Reception Channels and Transmission Channels for the current <code>DcmDslProtocol</code>.</p> <p>Three types of connections can be defined:</p> <ul style="list-style-type: none"> ▶ <code>DcmDslMainConnection</code>, used for Request Triggered Transmissions. Configure such a connection for OBD services and UDS services. ▶ <code>DcmDslPeriodicTransmission</code>, used for Periodic DID Transmissions. Configure such a connection if the UDS service <code>ReadDataByPeriodicIdentifier</code> (0x2A) is used and Type 2 communication is required.



Containers included

		► DcmDslResponseOnEvent , used for Event Triggered Transmissions. Configure such a connection if the UDS service ResponseOnEvent (0x86) is used and Type 2 communication is required.
--	--	---

Parameters included

Parameter name	Multiplicity
DcmDslProtocolEndiannessConvEnabled	1..1
DcmDslObdNRCResponseSupression	1..1
DcmDslProtocolID	1..1
DcmDslProtocolsParallelExecution	1..1
DcmDslProtocolPreemptTimeout	1..1
DcmDslProtocolStackNumber	1..1
DcmDslProtocolPriority	1..1
DcmDslProtocolTransType	0..1
DcmSendRespPendOnTransToBoot	1..1
DcmTimStrP2ServerAdjust	1..1
DcmTimStrP2StarServerAdjust	1..1
DcmDslProtocolRxBufferID	1..1
DcmDemClientRef	0..1
DcmDslProtocolSIDTable	1..1
DcmDslProtocolSessionRef	0..1
DcmDslProtocolTxBufferID	1..1

Parameter Name	DcmDslProtocolEndiannessConvEnabled
Label	Enable endianness conversion
Description	<p>Enables the endianness conversion, per Diagnostic Protocol.</p> <p>If enabled, for signals of fixed length which are received or transmitted when using the following services:</p> <ul style="list-style-type: none"> ► RoutineControl (0x31) ► ReadDataByIdentifier (0x22) ► WriteDataByIdentifier (0x2E) ► InputOutputControlByIdentifier (0x2F)



	<ul style="list-style-type: none"> ▶ OBD Service \$01 <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: Enable endianness conversion for this Diagnostic Protocol. ▶ FALSE: Disable endianness conversion for this Diagnostic Protocol.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU V1.0.0

Parameter Name	DcmDslObdNrcResponseSupression
Description	<p>Enable/Disable prompting the "Service Not Supported" NRC when OBD services are requested but not enabled.</p> <ul style="list-style-type: none"> ▶ TRUE = Enable "Service Not Supported" NRC Supression: No response will be sent when OBD services are requested but are not configured. ▶ FALSE = Disable "Service Not Supported" NRC Supression: The NRC "Service Not Supported" will be sent when OBD services are requested but are not configured.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDslProtocolID
Label	Protocol ID
Description	<p>Defines the type of the Diagnostic Protocol being configured.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DCM_OBD_ON_CAN: OBD on CAN ▶ DCM_OBD_ON_FLEXRAY: OBD on Flexray ▶ DCM_OBD_ON_IP: OBD on IP ▶ DCM_UDS_ON_CAN: UDS on CAN ▶ DCM_UDS_ON_FLEXRAY: UDS on FlexRay



	<ul style="list-style-type: none"> ▶ DCM_UDS_ON_IP: UDS on IP ▶ DCM_PERIODIC_ON_CAN: Periodic DID transmission on CAN ▶ DCM_PERIODIC_ON_FLEXRAY: Periodic DID transmission on Flexray ▶ DCM_PERIODICTRANS_ON_IP: Periodic DID transmission on IP ▶ DCM_ROE_ON_CAN: Event-triggered response on CAN ▶ DCM_ROE_ON_FLEXRAY: Event-triggered response on Flexray ▶ DCM_ROE_ON_IP: Event-triggered response on IP ▶ DCM_SUPPLIER_1: Reserved for SW supplier specific ▶ DCM_SUPPLIER_2: Reserved for SW supplier specific ▶ DCM_SUPPLIER_3: Reserved for SW supplier specific ▶ DCM_SUPPLIER_4: Reserved for SW supplier specific ▶ DCM_SUPPLIER_5: Reserved for SW supplier specific ▶ DCM_SUPPLIER_6: Reserved for SW supplier specific ▶ DCM_SUPPLIER_7: Reserved for SW supplier specific ▶ DCM_SUPPLIER_8: Reserved for SW supplier specific ▶ DCM_SUPPLIER_9: Reserved for SW supplier specific ▶ DCM_SUPPLIER_10: Reserved for SW supplier specific ▶ DCM_SUPPLIER_11: Reserved for SW supplier specific ▶ DCM_SUPPLIER_12: Reserved for SW supplier specific ▶ DCM_SUPPLIER_13: Reserved for SW supplier specific ▶ DCM_SUPPLIER_14: Reserved for SW supplier specific ▶ DCM_SUPPLIER_15: Reserved for SW supplier specific
Multiplicity	1..1
Type	ENUMERATION
Range	DCM_OBD_ON_CAN DCM_OBD_ON_FLEXRAY DCM_OBD_ON_IP DCM_UDS_ON_CAN DCM_UDS_ON_FLEXRAY DCM_UDS_ON_IP DCM_ROE_ON_CAN DCM_ROE_ON_FLEXRAY



	DCM_ROE_ON_IP
	DCM_PERIODIC_ON_CAN
	DCM_PERIODIC_ON_FLEXRAY
	DCM_PERIODICTRANS_ON_IP
	DCM_SUPPLIER_1
	DCM_SUPPLIER_2
	DCM_SUPPLIER_3
	DCM_SUPPLIER_4
	DCM_SUPPLIER_5
	DCM_SUPPLIER_6
	DCM_SUPPLIER_7
	DCM_SUPPLIER_8
	DCM_SUPPLIER_9
	DCM_SUPPLIER_10
	DCM_SUPPLIER_11
	DCM_SUPPLIER_12
	DCM_SUPPLIER_13
	DCM_SUPPLIER_14
	DCM_SUPPLIER_15
Configuration class	VariantPreCompile:
Origin	AUTOSAR_ECUC

Parameter Name	DcmDslProtocolsParallelExecutab
Label	Enable parallel execution of protocols
Description	<p>Enables the parallel processing of Event-triggered responses (configured via the ResponseOnEvent (0x86) UDS service) and Periodic DID Transmissions (configured via the ReadDataByPeriodicIdentifier (0x2A) UDS service).</p> <p>Note 1: Only these two protocols (i.e. one managing Event-triggered responses and one managing Periodic DID Transmissions) are allowed to run in parallel to another protocol servicing either a UDS or an OBD service request.</p> <p>Note 2: the functionality related to this parameter is not supported by the current implementation.</p>
Multiplicity	1..1



Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU V1.0.0	

Parameter Name	DcmDslProtocolPreemptTimeout			
Label	Protocol preemption timeout			
Description	<p>This parameter is the time-out value used in protocol preemption if this protocol preempts another diagnostic protocol.</p> <p>The protocol shall be started maximum DcmDslProtocolPreemptTimeout time after the first request in the new protocol.</p> <p>Note: Since some resources like buffers and data providers cannot be forced by the Dcm to be freed, the Dcm can only start a new protocol if other protocols cooperatively returned resources by confirming cancellation. See limitations for more information.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 1 second 			
Multiplicity				
Type				
Default value	1.0			
Range	<p><=1</p> <p>>=0.001</p>			
Configuration class	VariantPreCompile:	VariantPreCompile		
Origin	AUTOSAR_ECU			

Parameter Name	DcmDslProtocolStackNumber	
Label	Protocol Stack Number	
Description	<p>Defines the stack number. The stack numbers configured shall be zero-based and consecutive.</p> <p>Protocols on the same Stack will obey the preemption mechanism, i.e. will obey preemption upon priority laws.</p> <p>Two Protocols on different stacks may run in parallel, unless they are of the same kind (OBD/UDS).</p>	



	<p>In order for two Protocols to run in parallel:</p> <ul style="list-style-type: none"> ▶ Configure different Stack Numbers for them. ▶ Configure different types for the two Protocols via configuration parameter DcmDslProtocolID <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 255
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<p><=255</p> <hr/> <p>>=0</p>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDslProtocolPriority
Label	Protocol Priority
Description	<p>Defines the priority for this Diagnostic Protocol. This is considered for protocol preemption handling. A higher priority Diagnostic Protocol is allowed to preempt an already executing lower priority one.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 255 <p>Note:</p> <ul style="list-style-type: none"> ▶ 0 is the highest priority (one Diagnostic Protocol which has the priority equal to 0 cannot be preempted) ▶ 255 is the lowest priority
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<p><=255</p> <hr/> <p>>=0</p>
Configuration class	VariantPreCompile: VariantPreCompile



Origin	AUTOSAR_ECUC	
Parameter Name	DcmDslProtocolTransType	
Label	Protocol Transmission Type	
Description	Note: This configuration parameter is deprecated and should no longer be used. It does not have any effect on the generated code.	
Multiplicity	0..1	
Type	ENUMERATION	
Range	TYPE1 TYPE2	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC V1.0.0	
Parameter Name	DcmSendRespPendOnTransToBoot	
Label	Send RP on TransToBoot	
Description	<p>Defines whether the Dcm shall transmit a RequestCorrectlyReceivedResponsePending (0x78) Negative Response Code before transitioning to the bootloader as a result of processing a DiagnosticSessionControl (0x10) service request for a Diagnostic Session DcmDspSession which has the DcmDspSessionForBoot parameter configured to either DCM_OEM_BOOT or DCM_SYS_BOOT.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: The Dcm shall transmit a RequestCorrectlyReceivedResponsePending (0x78) Negative Response Code before transitioning to the bootloader ▶ FALSE: The Dcm shall not transmit a RequestCorrectlyReceivedResponsePending (0x78) Negative Response Code before transitioning to the bootloader 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	
Parameter Name	DcmTimStrP2ServerAdjust	
Label	P2ServerAdjust	



Description	Defines the P2 timeout adjustment. This parameter is used to guarantee that a response is available on bus before the P2 timeout expiration. If a positive or negative response is not ready, the Dcm sends the first RequestCorrectlyReceivedResponsePending (0x78) Negative Response Code after (DcmDspSessionP2ServerMax - DcmTimStrP2ServerAdjust) seconds. Range: ▶ 0 .. 1 second
Multiplicity	1..1
Type	FLOAT
Default value	0.0
Range	<=1 ≥0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmTimStrP2StarServerAdjust	
Label	P2*ServerAdjust	
Description	Defines the P2* timeout adjustment. This parameter is used to guarantee that a response is available on bus before the P2* timeout expiration. If a positive or negative response is not ready, the Dcm sends subsequent RequestCorrectlyReceivedResponsePending (0x78) Negative Response Code after every (DcmDspSessionP2StarServerMax - DcmTimStrP2Star ServerAdjust) seconds according to enhanced response timing requirements. Range: ▶ 0 .. 5 seconds.	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.0	
Range	<=5 ≥0	
Configuration class	VariantPreCompile:	VariantPreCompile



Origin	AUTOSAR_ECUC
Parameter Name	DcmDslProtocolRxBufferID
Label	Reference to Rx Buffer
Description	<p>Reference to the reception <code>DcmDslBuffer</code> configured for the protocol.</p> <p>This parameter is not used and therefore disabled.</p> <p>Please use the vendor specific parameter "<code>as:modconf('Dcm')/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/ DcmDslMainConnection/DcmDslProtocolRx/DcmDslProtocolRxBufferID</code>" instead. This enhancement allows the usage of a different reception buffer for every RxPduId, thus allowing greater flexibility in configuration.</p> <p>Note: the functionality related to this parameter is not supported by the current implementation.</p>
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	DcmDemClientRef
Label	Reference to DemClient
Description	Reference to the <code>DemClient</code> configured for this protocol.
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	DcmDslProtocolSIDTable
Label	Reference to Service Table
Description	Reference to the <code>DcmDsdServiceTable</code> Diagnostic Service Table configured for this protocol.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC



Parameter Name	DcmDslProtocolSessionRef	
Label	Reference to Session Table	
Description	<p>Reference to the <code>DcmDspSession</code> Diagnostic Session Table configured for this protocol.</p> <p>Only Diagnostic Sessions belonging to this table shall be available for transitions via the <code>DiagnosticSessionControl</code> (0x10) diagnostic service.</p>	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDslProtocolTxBufferID	
Label	Reference to Tx Buffer	
Description	<p>Reference to <code>DcmDslBuffer</code> transmit buffer configured for this protocol.</p> <p>This is the buffer used for transmission of responses.</p> <p>Note 1: If Paged Buffer Transmission is not used by services of this protocol, the size must be configured large enough to fit the largest expected diagnostic service response.</p> <p>Note 2: If Paged Buffer Transmission is used by services of this protocol, the size must be configured large enough to fit the largest expected Page of data (subdivision of the diagnostic service response provided by the Application or another Basic Software Module).</p>	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.15. DcmDslConnection

Containers included		
Container name	Multiplicity	Description
DcmDslMainConnection	1..1	<p>Label: Channel Configuration</p> <p>Defines the configuration for a Main Connection.</p>



Containers included

		<p>The <code>DcmDslMainConnection</code> contains:</p> <ul style="list-style-type: none"> ▶ Configuration for Reception Channels ▶ Transmission configuration parameters ▶ Reference to <code>DcmDslResponseOnEvent</code> and <code>DcmDslPeriodicTransmission</code> if applicable ▶ Tester Source Address configuration <p>Configure this Connection for Request Triggered Transmissions for OBD and UDS protocols.</p>
DcmDslPeriodicTransmission	1..1	<p>Label: Periodic Transmission Configuration</p> <p>Defines the configuration for Periodic Transmission connections.</p> <p>If service <code>ReadDataByPeriodicIdentifier</code> (0x2A) is used and Type 2 communication is intended for the <code>DcmDslProtocol</code> processing the <code>ReadDataByPeriodicIdentifier</code> service requests, configure the Transmission parameters here and reference this <code>DcmDslPeriodicTransmission</code> from the <code>DcmDslPeriodicTransmissionConRef</code> of the corresponding <code>DcmDslProtocol</code>'s <code>DcmDslMainConnection</code>.</p>
DcmDslResponseOnEvent	1..1	<p>Label: ROE Configuration</p> <p>This parameter is used to configure Event Triggered responses for <code>ResponseOnEvent</code> (0x86) service</p> <p>If service <code>ResponseOnEvent</code> (0x86) is used and TYPE2 communication is intended for the <code>DcmDslProtocol</code> processing the <code>ResponseOnEvent</code> service requests, configure the Transmission parameters here and reference this <code>DcmDslResponseOnEvent</code> from the <code>DcmDslROEConnectionRef</code> of the corresponding <code>DcmDslProtocol</code>'s <code>DcmDslMainConnection</code>.</p> <p>If both sub-containers are inactive that means that the ROE event is processed by the protocol that owns this connection, and the response is sent using the Tx Pduld of the MainConnection where the StartROE request was received from.</p>



Containers included

		If DcmDslRoeTxPduRef/DcmDslRoeTxConfirmationPduld parameters are enabled and contains a valid pair for transmission, then the TxPduld on which the ServiceToRespond-To response is sent is the TxPduld configured within this DcmDslResponseOnEvent connection.
--	--	---

5.2.1.16. DcmDslMainConnection

Containers included

Container name	Multiplicity	Description
DcmDslProtocolRx	1..n	<p>Defines the configuration of the Reception Channels for the protocol.</p> <p>A Reception Channel defines, for each RxPduId:</p> <ul style="list-style-type: none"> ▶ The type of the requests from this RxPduId (Physical or Functional) ▶ The unique RxPduId ▶ The reference to the Pdu in the EcucPduCollection ▶ The reference to the DcmDslBuffer used to store requests received on this RxPduId ▶ The DcmDslProtocolRxChannelId Network Identifier for this Reception Channel ▶ The reference to the ComMChannel relative to this Reception Channel <p>Note: More than one Reception Channel can be configured per protocol.</p>
DcmDslProtocolTx	0..1	<p>Label: Transmit Channel Configuration</p> <p>Defines the parameters controlling Transmission for this DcmDslMainConnection.</p>

Parameters included

Parameter name	Multiplicity
DcmDslProtocolRxTesterSourceAddr	1..1
DcmDslPeriodicTransmissionConRef	0..1

**Parameters included**

DcmDslROEConnectionRef	0..1
--	------

Parameter Name	DcmDslProtocolRxTesterSourceAddr
Label	Tester Source Address
Description	<p>Defines the Tester Source Address used for Receptions on this DcmDslMainConnection.</p> <p>A TesterSourceAddress uniquely describes a client and will be used in:</p> <ul style="list-style-type: none"> ▶ Calls to operations of the <code>ServiceRequestNotification</code> interface in order to allow the Application to identify which client has executed the request ▶ The Programming Conditions (see interfaces <code>Dcm_SetProgConditions()</code> and <code>Dcm_GetProgConditions()</code>) storing the conditions of a jump to/from the Bootloader, in order to identify the client requesting the jump and therefore route the response to the request correctly. <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 65535
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<p><=65535</p> <p>>=0</p>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name**DcmDslPeriodicTransmissionConRef**

Label	Reference to PeriodicTransmission protocol
--------------	--

Description	<p>Reference the <code>DcmDslPeriodicTransmission</code> connection which shall be used for the transmission of Periodic DID Responses configured via a <code>ReadDataByPeriodicIdentifier</code> (0x2A) request received (if supported and successful) on an <code>RxPduId</code> belonging to a <code>DcmDslProtocolRx</code> Reception Channel configured for this <code>DcmDslMainConnection</code>.</p> <p>This parameter configures the link to the connection of PeriodicTransmission protocol for the processing of the PeriodicTransmission events.</p>
Multiplicity	0..1



Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU V1.0.0	

Parameter Name	DcmDslROEConnectionRef	
Label	Reference to ROE protocol	
Description	Reference the <code>DcmDslResponseOnEvent</code> connection which shall be used for the transmission of Event Triggered Responses configured via a <code>ResponseOnEvent</code> (0x86) request received (if supported and successful) on an Rx-PduId belonging to a <code>DcmDslProtocolRx</code> Reception Channel configured for this <code>DcmDslMainConnection</code> .	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU V1.0.0	

5.2.1.17. DcmDslProtocolRx

Parameters included	
Parameter name	Multiplicity
DcmDslProtocolRxAddrType	1..1
DcmDslProtocolRxChannelId	1..1
DcmDslProtocolRxPdulId	1..1
DcmDslProtocolRxComMChannelRef	1..1
DcmDslProtocolRxPduRef	1..1
DcmDslProtocolRxBufferID	1..1

Parameter Name	DcmDslProtocolRxAddrType
Label	Communication Type
Description	<p>Defines the type of communication for this channel.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ <code>DCM_PHYSICAL_TYPE</code>: Physical Addressing is used on this Reception Channel. A node of a specific communication network receives a message from one sending node (1 to 1 communication).



	<ul style="list-style-type: none"> ▶ DCM_FUNCTIONAL_TYPE: Functional Addressing is used on this Reception Channel. All the nodes of a specific communication network receives a message from one sending node (1 to n communication).
Multiplicity	1..1
Type	ENUMERATION
Default value	DCM_FUNCTIONAL_TYPE
Range	DCM_FUNCTIONAL_TYPE DCM_PHYSICAL_TYPE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDslProtocolRxChannelId
Label	Network Identifier
Description	<p>Defines the Channel Identifier associated to the received Pdu.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 255 <p>Note: The value configured in DcmDslProtocolRxChannelId must match the value configured in the ComMchannelId parameter of the ComMchannel referenced via DcmDslProtocolRxComMchannelRef.</p>
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDslProtocolRxPdulId
Label	Reception Pdu Id
Description	<p>Defines the unique identifier of the Pdu used for receiving diagnostic requests for this channel.</p> <p>This PduId is used by the lower layer (The PduR or a Transport Protocol) in the callbacks issued to the Dcm in order to perform a data Reception (from Dcm's point of view):</p> <ul style="list-style-type: none"> ▶ Dcm_StartOfReception()



	<ul style="list-style-type: none"> ▶ Dcm_CopyRxData() ▶ Dcm_TpRxIndication() <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 65535 <p>Note: Values configured for the <code>DcmDslProtocolRxPduIds</code> shall be zero-based and consecutive.</p>
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDslProtocolRxComMChannelRef
Label	Reference to the ComMChannel
Description	Reference to the ComMChannel on which the Pdu is received.
Multiplicity	1..1
Type	CHOICE-REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDslProtocolRxPduRef
Label	Reference to Rx Pduld
Description	<p>Reference to Pdu from the EcucPduCollection configured for this Reception Channel.</p> <p>Through this reference, the PduRouter can resolve the PduId used for Reception by the Dcm in the following APIs:</p> <ul style="list-style-type: none"> ▶ Dcm_StartOfReception() ▶ Dcm_CopyRxData() ▶ Dcm_TpRxIndication() <p>and defined by the Dcm in DcmDslProtocolRxPduId.</p>
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile



Origin	AUTOSAR_ECUC
Parameter Name	DcmDslProtocolRxBufferID
Label	Reference to Rx Buffer
Description	<p>Reference to the <code>DcmDslBuffer</code> configuring Reception Buffer assigned for this Reception Channel.</p> <p>Configure this reference so that the Reception Buffer is large enough to fit the largest expected request message arriving on this <code>RxPduId</code>.</p> <ul style="list-style-type: none"> ▶ For <code>SecurityAccess(0x27)</code>: <code>DcmDslBufferSize</code> need to be higher than <code>(DcmDspSecurityADRSIZE + 2)</code> and <code>(DcmDspSecurityKeySize + 2)</code> for all configured Security Levels. ▶ For <code>RoutineControl(0x31)</code>: <code>DcmDslBufferSize</code> need to be higher than <code>(DcmDspRoutineSignalPos + DcmDspRoutineSignalLength + 7) / 8 + 4</code> for <code>DcmDspRoutineRequestResIn + DcmDspRoutineStopIn + DcmDspStartRoutineIn</code>. <p>Note: The <code>Dcm</code> allows a single <code>DcmDslBuffer</code> to be used by more than one <code>DcmDslProtocolRx</code> Reception Channel, and at the same time allows full flexibility in using the same <code>DcmDslBuffer</code> as a Reception and a Transmission Buffer at the same time.</p>
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.2.1.18. DcmDslProtocolTx

Parameters included	
Parameter name	Multiplicity
DcmDslTxConfirmationPduld	1..1
DcmDslProtocolTxPduRef	1..1

Parameter Name	DcmDslTxConfirmationPduld
Label	Transmit Confirmation Pduld
Description	This <code>PduId</code> is used by the lower layer (The <code>PduR</code> or a Transport Protocol) in the callbacks issued to the <code>Dcm</code> in order to perform a data Transmission (from <code>Dcm</code> 's point of view):



	<ul style="list-style-type: none"> ▶ Dcm_CopyTxData() ▶ Dcm_TpTxConfirmation() <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 65535 <p>Note: Values configured for the <code>DcmDslTxConfirmationPduIds</code> shall be zero-based and consecutive.</p>
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDslProtocolTxPduRef	
Label	Reference to Tx Pdulid	
Description	<p>Reference to the <code>Pdu</code> in the <code>EcucPduCollection</code> configured for this Transmission Channel.</p> <p>Through this reference, <code>Dcm</code> can resolve the <code>PduId</code> used for Transmission in the <code>PduR_DcmTransmit()</code> API and defined by the <code>PduR</code>.</p> <p>Through this reference, <code>PduRouter</code> can resolve the <code>PduId</code> used for Transmission by in the following APIs:</p> <ul style="list-style-type: none"> ▶ <code>Dcm_CopyTxData()</code> ▶ <code>Dcm_TpTxConfirmation()</code> <p>and defined by the <code>Dcm</code> in <code>DcmDslTxConfirmationPduId</code>.</p>	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPreCompile: VariantPreCompile	
Origin	AUTOSAR_ECUC	

5.2.1.19. DcmDslPeriodicTransmission

Containers included		
Container name	Multiplicity	Description



Containers included

DcmDslPeriodicConnection	0..n	<p>Label: Transmission Pdu Configuration</p> <p>Defines the configuration for one Periodic DID Transmission connection.</p> <p>This container configures:</p> <ul style="list-style-type: none"> ▶ The ConfirmationTxPduId for Periodic DID Transmission ▶ The reference to the corresponding Pdu in the EcucPduCollection <p>This container configures the TxPduId for PeriodicTransmission.</p>
--	------	--

5.2.1.20. DcmDslPeriodicConnection

Parameters included

Parameter name	Multiplicity
DcmDslPeriodicTxConfirmationPduld	1..1
DcmDslPeriodicTxPduRef	1..1

Parameter Name	DcmDslPeriodicTxConfirmationPduld	
Label	TxConfirmation Hanlde Id	
Description	<p>Defines the configuration for one Periodic DID Transmission connection.</p> <p>This container configures:</p> <ul style="list-style-type: none"> ▶ The ConfirmationTxPduId for Periodic DID Transmission ▶ The reference to the corresponding Pdu in the EcucPduCollection <p>The Handle Id used by the lower layer to confirm the transmission of periodic data.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	VariantPreCompile:	VariantPreCompile



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

Parameter Name	DcmDslPeriodicTxPduRef	
Label	Reference to Tx Pduld	
Description	<p>Reference to the <code>Pdu</code> in the <code>EcucPduCollection</code> configured for this Transmission Channel.</p> <p>Through this reference, the <code>Dcm</code> can resolve the <code>PduId</code> used for Transmission in the <code>PduR_DcmTransmit()</code> API and defined by the <code>PduR</code>.</p> <p>Through this reference, the <code>PduRouter</code> can resolve the <code>PduId</code> used for Transmission by in the following APIs:</p> <ul style="list-style-type: none"> ▶ <code>Dcm_CopyTxData()</code> ▶ <code>Dcm_TpTxConfirmation()</code> <p>and defined by the <code>Dcm</code> in <code>DcmDslPeriodicTxConfirmationPduId</code>.</p> <p>This configures the Pduld to be used for sending Periodic Transmission.</p> <p>This is needed only for TYPE2 Periodic Transmissions.</p>	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.21. DcmDslResponseOnEvent

Parameters included		
Parameter name		Multiplicity
DcmDslRoeTxConfirmationPduld		0..1
DcmDslRoeTxPduRef		0..1

Parameter Name	DcmDslRoeTxConfirmationPduld
Description	Defines the <code>PduId</code> to be used by the lower layer (the <code>PduR</code> or a Transport-Protocol) in callbacks issued to the <code>Dcm</code> in order to perform a data Transmission (from the <code>Dcm</code> 's point of view):

- ▶ `Dcm_CopyTxData()`



	<ul style="list-style-type: none"> ▶ Dcm_TpTxConfirmation() <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 65535 <p>Note: Values configured for the <code>DcmDslRoeTxConfirmationPduIds</code> shall be zero-based and consecutive.</p>
Multiplicity	0..1
Type	INTEGER
Default value	0
Range	<p><=65535</p> <hr/> <p>>=0</p>
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DcmDslRoeTxPduRef	
Label	Reference to Tx PdulId	
Description	<p>Reference to the <code>Pdu</code> in the <code>EcucPduCollection</code> configured for this Transmission Channel.</p> <p>Through this reference, the <code>Dcm</code> can resolve the <code>PduId</code> used for Transmission in the <code>PduR_DcmTransmit()</code> API and defined by the <code>PduR</code>.</p> <p>Through this reference, the <code>PduRouter</code> can resolve the <code>PduId</code> used for Transmission by in the following APIs:</p> <ul style="list-style-type: none"> ▶ <code>Dcm_CopyTxData()</code> ▶ <code>Dcm_TpTxConfirmation()</code> <p>and defined by the <code>Dcm</code> in <code>DcmDslRoeTxConfirmationPduId</code>.</p>	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile: VariantPreCompile	
Origin	AUTOSAR_ECU	

5.2.1.22. DcmDslServiceRequestManufacturerNotification



5.2.1.23. DcmDslServiceRequestSupplierNotification

5.2.1.24. DcmDsp

Containers included		
Container name	Multiplicity	Description
DcmDspComControl	1..1	<p>Label: Com Control</p> <p>Defines the configuration for the CommunicationControl (0x28) UDS service.</p>
DcmDspControlDTCSetting	1..1	<p>Defines configuration for the behaviour of the ControlDTCSetting (0x85) service handler.</p>
DcmDspData	0..n	<p>Label: DID Data Configuration</p> <p>Defines the configuration parameters for DcmDspData elements representing DID signals.</p>
DcmDspDataInfo	0..n	<p>Defines the following information for a DcmDspData signal referencing this container via the DcmDspDataInfoRef parameter:</p> <ul style="list-style-type: none"> ▶ Whether the length of this signal is fixed or not. ▶ If scaling information is available for this signal, the size of the scaling information.
DcmDspDid	0..n	<p>Label: DSP DID</p> <p>Defines the configuration for DIDs (Data Identifiers).</p>
DcmDspDidInfo	0..n	<p>Label: DID Info</p> <p>Defines parameters configuring:</p> <ul style="list-style-type: none"> ▶ Whether the referencing DcmDspDid DID is dynamically or statically defined ▶ Types of access available for the referencing DcmDspDid DID (Read, Write or IOControl) ▶ Permissions for each type of access
DcmDspDidRange	0..n	<p>Label: Did Range</p> <p>Defines the configuration of the DID Range.</p>
DcmDspMemory	0..1	<p>Label: Memory Access Configuration</p>



Containers included

		<p>Defines the configuration for memory access</p> <p>The configuration within this container defines <code>memoryId</code> entities representing virtual memory sections. The actual Read and Write accesses for these virtual memory sections are done via callouts provided by the User, therefore the significance of such a <code>memoryId</code> section is Application implementation-dependent. Additionally, the supported <code>AddressAndLengthFormatIdentifiers</code> are configured here.</p> <p>Configure this container if any of the following diagnostic services are required to be supported by the <code>Dcm</code>:</p> <ul style="list-style-type: none"> ▶ <code>ReadMemoryByAddress</code> (0x23) ▶ <code>WriteMemoryByAddress</code> (0x3D) ▶ <code>RequestDownload</code> (0x34) ▶ <code>RequestUpload</code> (0x35) ▶ <code>TransferData</code> (0x36) ▶ <code>RequestTransferExit</code> (0x37) ▶ <code>DynamicallyDefineDataIdentifier</code> (0x2C)
<u>DcmDspPeriodicDidTransmission</u>	0..1	Defines the configuration for the Periodic Did transmission. This container exists only if the UDS Service <code>ReadDataByPeriodicIdentifier</code> (0x2A) is configured.
<u>DcmDspPeriodicTransmission</u>	0..1	Defines the configuration parameters for Periodic Transmission Processing
<u>DcmDspPid</u>	0..255	<p>Label: PID</p> <p>Defines the configuration of PIDs to be used in OBD Services:</p> <ul style="list-style-type: none"> ▶ \$01 - Request Current Powertrain diagnostic Data ▶ \$02 - Request Power Train FreezeFrame Data
<u>DcmDspReadGenericInformation</u>	1..1	Defines configuration for the behaviour of the <code>ReadGenericInformation</code> (0xAF) service handler.
<u>DcmDspRequestControl</u>	0..n	Defines the configuration parameters for the OBD Service <code>\$08</code> - Request Control of On-Board System, Test or Component).



Containers included		
		This operation is realized by either configuring a callout function with synchronous behaviour or alternatively by using by using a synchronous AUTOSAR ClientServerInterface with synchronous behaviour (the operation shall be completed in one single call to the operation of the generated interface). The Dcm shall generate the RequestControlServices_<TID> ClientServerInterface, where <TID> is the name of this DcmDspRequestControl container. This ClientServerInterface contains the necessary RequestControl operation. A Require Port is generated for this interface and shall be connected to a compatible Provide Port of an Application Software Component. The name of this Require Port is the same as the name of the ClientServerInterface.
<u>DcmDspRoe</u>	0..1	<p>Label: ROE Service Configuration</p> <p>Provide the configuration of the ResponseOnEvent mechanism.</p>
<u>DcmDspRoutine</u>	0..65535	Defines the configuration for Diagnostic Routines, to be used by the internal implementation of the RoutineControl (0x31) service handler.
<u>DcmDspRoutineInfo</u>	0..n	<p>Defines, for the referencing DcmDspRoutine Diagnostic Routine, parameters configuring:</p> <ul style="list-style-type: none"> ▶ Operations available to this Diagnostic Routine ▶ Signal configurations for request and response messages to the RoutineControl (0x31) Diagnostic Service for this Diagnostic Routine ▶ Permissions for operations on this Diagnostic Routine.
<u>DcmDspSecurity</u>	1..1	<p>Label: Security Level Configuration</p> <p>Defines the configuration for Security Levels, which is further used in order to condition functionality on the current Security Level of the ECU.</p>
<u>DcmDspSession</u>	1..n	<p>Label: Session Configuration</p> <p>Defines the configuration for Diagnostic Sessions, which is further used in order to:</p>



Containers included

		<ul style="list-style-type: none"> ▶ Condition functionality on the current Diagnostic Session of the ECU. ▶ Trigger events on switching from one Diagnostic Session to another ▶ Perform jumps to and from Bootloader software.
DcmDspTestResultByObdmid	0..1	<p>This parameter is enabled only if service 0x06 is supported.</p> <p>Defines the configuration for OBD service \$06 (Request On-Board Monitoring Test-results for Specific Monitored Systems).</p> <p>Contains configuration for:</p> <ul style="list-style-type: none"> ▶ OBDMIDs (On-Board Diagnostic Monitor IDs) ▶ Tids (Test IDs) belonging to OBDMIDs
DcmDspVehInfo	0..n	Defines the configuration parameters for the OBD Service \$09 - Request Vehicle Information.

Parameters included

Parameter name	Multiplicity
DcmDspDDIDCheckPerSourceDID	1..1
DcmDspGenerateOnlyNeededIOControlOperations	1..1
DcmDspMaxDidToRead	1..1
DcmDspMaxPeriodicDidToRead	1..1
DcmDspPowerDownTime	0..1
DcmDspReturnControlToEcuTimeout	1..1
DcmDspRoutineEnableDefaultInterfaces	1..1
DcmDspRoutineEnableRoutineInfoByte	1..1
DcmDspRoutineVariableLengthInBytes	1..1
DcmDspDidEnableDefaultInterfaces	1..1
DcmDspSessionControlTiming	1..1

Parameter Name	DcmDspDDIDCheckPerSourceDID
Label	Check sources of DDIDs each time before reading



Description	Defines whether the DCM module shall check the session, security and mode dependencies per source DIDs with a <code>ReadDataByIdentifier</code> (0x22) referencing a DID in the range from 0xF200 to 0xF3FF
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DcmDspGenerateOnlyNeededIOControlOperations
Label	Generate Only Needed IOControl Operations
Description	<p>Defines whether the DCM module shall generate only Client-Server Interface Operations of used IOControl operations.</p> <ul style="list-style-type: none"> ▶ TRUE: The <code>DataServices_<Data></code> <code>ClientServerInterface</code> generated for <code>DcmDspData</code> elements referenced by DIDs which have Control-Access enabled (<code>DcmDspData</code> elements referenced from <code>DcmDspDidSignal</code> elements belonging to <code>DcmDspDid</code> entries which reference a <code>DcmDspDidInfo</code> which has any operations within the <code>DcmDspDidControl</code> container enabled) shall contain only operations corresponding to used (configured) IO-Control operations. ▶ FALSE: The <code>DataServices_<Data></code> <code>ClientServerInterface</code> generated for <code>DcmDspData</code> elements referenced by DIDs which have Control-Access enabled (<code>DcmDspData</code> elements referenced from <code>DcmDspDidSignal</code> elements belonging to <code>DcmDspDid</code> entries which reference a <code>DcmDspDidInfo</code> which has the <code>DcmDspDidControl</code> container enabled) shall contain operations corresponding to all IOControl operations, regardless if they are configured to be used or not.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspMaxDidToDelete
Label	Maximum number of DIDs supported
Description	Defines the maximum allowed number of DIDs in a single <code>ReadDataByIdentifier</code> (0x22) service request.



	<p>Configure this number depending on the maximum number of DIDs expected to be read. This influences the size of the Reception Buffers, as these must accommodate the complete <code>ReadDataByIdentifier</code> (0x22) service request.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 1 .. 65535 DIDs supported
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<p><=65535</p> <hr/> <p>>=1</p>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DcmDspMaxPeriodicDidToRead
Label	DcmDspMaxPeriodicDidToRead
Description	Defines the maximum allowed number of periodic DIDs which can be requested in a single <code>ReadDataByPeriodicIdentifier</code> request.
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<p><=65535</p> <hr/> <p>>=1</p>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspPowerDownTime
Label	Power Down Time
Description	<p>Defines the minimum time of the stand-by sequence (time duration for which the server remains in the power down sequence during Rapid Power Shutdown), in seconds.</p> <p>If this parameter is enabled, its value shall be included in the positive response to the <code>EcuReset</code> (0x11) service request with sub-function <code>enableRapidPowerShutDown</code> (0x04).</p>



	<p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 254 seconds <p>Note: 255 indicates a failure or time is not available in server.</p>
Multiplicity	0..1
Type	INTEGER
Default value	255
Range	<p><=255</p> <p>>=0</p>
Configuration class	<p>PreCompile: VariantPreCompile</p>
Origin	AUTOSAR_ECU

Parameter Name	DcmDspReturnControlToEcuTimeout
Label	ReturnControlToEcu processing time
Description	<p>Note: This is a Vendor Specific parameter of the Dcm.</p> <p>Defines the maximum amount of time available to execute all <code>ReturnControlToEcu</code> operations for all signals belonging to DIDs which have <code>IOControl</code> operations available, i.e. for the DID in question, the referenced <code>DcmDspDidInfo</code> has the container <code>DcmDspDidControl</code> enabled and one of the parameters:</p> <ul style="list-style-type: none"> ▶ <code>DcmDspDidFreezeCurrentState</code> ▶ <code>DcmDspDidResetToDefault</code> ▶ <code>DcmDspDidShortTermAdjustment</code> <p>enabled. As the <code>ReturnControlToEcu</code> operation for some signals might be asynchronous, this means that the operation may span over more than one <code>Dcm_MainFunction()</code> execution, time in which a new service request might be received by the <code>Dcm</code>. If this new service request operates on DIDs, the newly requested operations might interfere with the <code>ReturnControlToEcu</code> operations still being run in the system. Because of this, new operations on DIDs must be rejected until the <code>ReturnControlToEcu</code> operations have all finished. As this may take an unlimited amount of time if one of the controlled signals fails to return a valid status to the <code>ReturnControlToEcu</code> operation, there must be a method to limit the time available for these operations to finish. Until the return of control to ECU is complete for all signals, the signals are not considered to be in a stable state. Thus, this parameter has been introduced in order to ensure that the following services:</p> <ul style="list-style-type: none"> ▶ <code>ReadDataByIdentifier (0x22)</code>



	<ul style="list-style-type: none"> ▶ <code>WriteDataByIdentifier</code> (0x2E) ▶ <code>IOControlByIdentifier</code> (0x2F) <p>which operate on DIDs don't remain blocked for an unlimited amount of time.</p> <p>Note: Configure to 0 to allow an unlimited amount of time for this operation to succeed.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 30 seconds
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<p><=30</p> <p>>=0</p>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspRoutineEnableDefaultInterfaces	
Label	RoutineControl Enable Default Interfaces	
Description	<p>Note: This is a Vendor Specific parameter of the Dcm.</p> <p>Defines whether the DCM Module shall call Application default interfaces for each unsupported (unconfigured or unused) RID (RIDs corresponding to configured OBD elements are considered supported).</p> <ul style="list-style-type: none"> ▶ TRUE: The following callout functions shall be present in Application software: <code>Dcm_DefaultRoutineStart()</code>, <code>Dcm_DefaultRoutineStop()</code>, <code>Dcm_DefaultRoutineRequestRoutineResults()</code>. For every Routine which is not supported (not configured or not used) Dcm module shall call the above Application functions according to the requested sub-service. ▶ FALSE: Every requested Routine which is not supported (not configured or not used) shall be rejected by Dcm module with <code>DCM_E_REQUESTOUT-OF RANGE Nrc.</code> 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	FALSE	
Configuration class	VariantPreCompile:	VariantPreCompile



Origin	Elektrobit Automotive GmbH
Parameter Name	DcmDspRoutineEnableRoutineInfoByte
Label	RoutineControl Enable Routine Info Byte
Description	<p>Note: This is a Vendor Specific parameter of the Dcm.</p> <p>Defines whether the DCM Module shall call Application interface for AddRoutineInfoByte, in order to add one more byte to the routine response.</p> <ul style="list-style-type: none"> ▶ TRUE: The following callout function shall be present in Application software: Dcm_AddRoutineInfoByte(). For every Routine which returns E_OK, one more byte will be added to the message in the beginning of the response. ▶ FALSE: Every requested routine will response with the original configured message.
Multiplicity	1..1
Type	BOOLEAN
Default value	FALSE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspRoutineVariableLengthInBytes
Label	RoutineControl Variable Length Parameters in Bytes
Description	<p>Note: This is a Vendor Specific parameter of the Dcm.</p> <p>Defines whether the parameter expressing the length of a signal which is part of a RoutineControl (0x31) diagnostic service request or response and has the DcmDspRoutineSignalType parameter set to VARIABLE_LENGTH (in generated API calls, the currentDataLength parameter) is to express the length of the input/output signal in bits or Bytes.</p> <ul style="list-style-type: none"> ▶ TRUE: The VARIABLE_LENGTH signal length (the currentDataLength function parameter) is expressed in Bytes. ▶ FALSE: The VARIABLE_LENGTH signal length (the currentDataLength function parameter) is expressed in bits.
Multiplicity	1..1
Type	BOOLEAN
Default value	FALSE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH



Parameter Name	DcmDspDidEnableDefaultInterfaces	
Label	Did Enable Default Interface	
Description	<p>Note: This is a Vendor Specific parameter of the Dcm.</p> <p>Defines whether the DCM Module shall call Application default interfaces for each unsupported (unconfigured or unused) DID (Service 0x22 - ReadDataByIdentifier, Service 0x2E - WriteDataByIdentifier, Service 0x2F - InputOutputControlByIdentifier).</p> <ul style="list-style-type: none"> ▶ TRUE: DIDs which are not configured are considered supported and they are handled by the supplier. ▶ FALSE: DIDs which are not configured are considered not supported. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	FALSE	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmDspSessionControlTiming	
Label	Enable Timing values in positive response	
Description	<p>Defines whether the positive response for service DiagnosticSessionControl (0x10) contains P2 Timing values (P2ServerMax and P2*ServerMax, represented by the DcmDspSessionP2ServerMax and DcmDspSessionP2StarServerMax configuration parameters of the DcmDspSessionRow describing the entered Diagnostic Session) or not.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: The P2ServerMax and P2*ServerMax timing values are present in the positive response for service DiagnosticSessionControl (0x10). ▶ FALSE: The P2ServerMax and P2*ServerMax timing values are not present in the positive response for service DiagnosticSessionControl (0x10). 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	



5.2.1.25. DcmDspComControl

Containers included		
Container name	Multiplicity	Description
<u>DcmDspComControlAllChannel</u>	0..n	<p>Label: All Com Control Channel</p> <p>Defines a collection of ComMChannels which shall be controlled if all networks are addressed by the CommunicationControl (0x28) service request.</p> <p>If the CommunicationControl (0x28) is requested with the subnet parameter (bits 4 .. 7) of the communicationType parameter in the CommunicationControl (0x28) request message equal to 0x00, the Dcm shall apply the requested controlType on all the Channels referenced here.</p>
<u>DcmDspComControlSetting</u>	1..1	<p>Label: All Com Control Setting</p> <p>Defines the configuration of the Communication control setting mode arbitration.</p>
<u>DcmDspComControlSpecificChannel</u>	0..n	<p>Label: Com Control Specific Channel</p> <p>Defines the correspondence between the subnet parameter (bits 4 .. 7) of the communicationType parameter in the CommunicationControl (0x28) request message and a specific ComMChannel which shall be controlled as a result of this request.</p> <p>Configure the necessary ComMChannels to be controlled here.</p>
<u>DcmDspComControlSubNode</u>	0..n	<p>Label: Communication Control Sub Node</p> <p>This container gives information about the node identification number and the ComM channel used to address a sub-network.</p> <p>Configure the necessary ComMChannels to be controlled here.</p>

5.2.1.26. DcmDspComControlAllChannel

Parameters included	
Parameter name	Multiplicity



Parameters included

DcmDspAllComMChannelRef	1..1
---	------

Parameter Name	DcmDspAllComMChannelRef	
Label	All Com Channel Refrence	
Description	Reference to a ComMChannel to be considered as part of the 'All channels' collection.	
Multiplicity	1..1	
Type	CHOICE-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.27. DcmDspComControlSetting

Parameters included

Parameter name	Multiplicity
DcmDspComControlCommunicationReEnableModeRuleRef	0..1

Parameter Name	DcmDspComControlCommunicationReEnableModeRuleRef	
Label	Re-enabling Com Control Mode Rule Refrence	
Description	<p>Reference to the DcmDspModeRule which controls re-enabling of communication by the Dcm.</p> <p>If this DcmDspModeRule evaluates to FALSE, for each controlled network, the Dcm shall:</p> <ul style="list-style-type: none"> ▶ Switch the mode of the Dcm_CommunicationControl_Network ModeDeclarationGroupPrototype to DCM_ENABLE_RX_TX_NORM_NM ▶ Call the BswM_Dcm_CommunicationMode_CurrentState() API with: <ul style="list-style-type: none"> ▶ Network set to the ComMChannel of the controlled network ▶ RequestedMode set to DCM_ENABLE_RX_TX_NORM_NM 	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	



5.2.1.28. DcmDspComControlSpecificChannel

Parameters included	
Parameter name	Multiplicity
DcmDspSubnetNumber	1..1
DcmDspSpecificComMChannelRef	1..1

Parameter Name	DcmDspSubnetNumber	
Label	Subnet Number	
Description	<p>Defines the subnet number.</p> <p>Configure the expected subnet parameter (bits 4 .. 7) of the communication-Type parameter in the CommunicationControl (0x28) request message here.</p>	
Multiplicity	1..1	
Type	INTEGER	
Range	<p><=14</p> <hr/> <p>>=1</p>	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspSpecificComMChannelRef	
Label	ComM Channel Reference	
Description	Reference to the ComMChannel corresponding to the configured DcmDspSubnetNumber.	
Multiplicity	1..1	
Type	CHOICE-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.29. DcmDspComControlSubNode

Parameters included	
Parameter name	Multiplicity
DcmDspComControlSubNodeld	1..1



Parameters included

DcmDspComControlSubNodeComMChannelRef	1..1
---	------

Parameter Name	DcmDspComControlSubNodeId
Label	SubNode ID
Description	<p>Defines the subnode ID.</p> <p>The node identification number <code>DcmDspComControlSubNodeId</code> is addressed by the CommunicationControl (0x28) request.</p>
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspComControlSubNodeComMChannelRef
Label	ComM Channel Refrence
Description	Reference to the ComMChannel corresponding to the configured <code>DcmDspComControlSubNodeId</code> .
Multiplicity	1..1
Type	CHOICE-REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.2.1.30. DcmDspControlDTCSetting

Parameters included	
Parameter name	Multiplicity
DcmSupportDTCSettingControlOptionRecord	1..1
DcmDspControlDTCSettingReEnableModeRuleRef	0..1

Parameter Name	DcmSupportDTCSettingControlOptionRecord
Label	Enable ControlOptionRecord
Description	<p>This configuration switch defines if the DTCSettingControlOptionRecord is required in the request message or not.</p> <p>Range:</p>



	<ul style="list-style-type: none"> ▶ TRUE: all requests to the ControlDTCSetting (0x85) service shall contain the DTCSettingControlOptionRecord, i.e. requests to the ControlDTCSetting (0x85) service not containing the DTCSettingControlOptionRecord will be rejected with an IncorrectMessageLengthOrInvalidFormat (0x13) NRC. ▶ FALSE: all requests to the ControlDTCSetting (0x85) service shall contain nothing else except the DTCSettingType, i.e. requests to the ControlDTCSetting (0x85) containing any other data besides the DTCSettingType will be rejected with an IncorrectMessageLengthOrInvalidFormat (0x13) NRC.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspControlDTCSettingReEnableModeRuleRef
Description	Reference to the DcmDspModeRule which controls re-enabling of DTC Setting by the Dcm. If this DcmDspModeRule evaluates to FALSE, in case the DTC Setting, the Dcm shall: <ul style="list-style-type: none"> ▶ Switch the mode of the DcmControlDTCSetting ModeDeclarationGroupPrototype to the ENABLEDTCSETTING ModeDeclaration ▶ Call the Dem_EnableDTCSetting() API
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.2.1.31. DcmDspData

Parameters included	
Parameter name	Multiplicity
DcmDspDataConditionCheckReadFncUsed	1..1
DcmDspDataConditionCheckReadFnc	0..1

**Parameters included**

DcmDspDataEcuSignal	0..1
DcmDspDataFreezeCurrentStateFnc	0..1
DcmDspDataGetScalingInfoFnc	0..1
DcmDspDataReadDataLengthFnc	0..1
DcmDspDataReadEcuSignal	0..1
DcmDspDataReadFnc	0..1
DcmDspDataResetToDefaultFnc	0..1
DcmDspDataReturnControlToEcuFnc	0..1
DcmDspDataShortTermAdjustmentFnc	0..1
DcmDspDataSize	1..1
DcmDspDataType	0..1
DcmDspDataUsePort	1..1
DcmDspDataWriteFnc	0..1
DcmDspDataInfoRef	1..1
DcmDspDataBlockIdRef	0..1
DcmDspDataEndianness	0..1

Parameter Name	DcmDspDataConditionCheckReadFncUsed	
Label	Enable Condition Check Function	
Description	Enables/disables the condition check read function.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Bugzilla #53669	

Parameter Name	DcmDspDataConditionCheckReadFnc
Label	Read DID condition check function
Description	Defines the name of the Callout used to check for Application conditions before reading signal data for the DID. If, for one of the signals requested to be read, the DcmDspDataConditionCheckReadFnc returns E_NOT_OK, the Dcm shall respond with a NegativeResponseCode equal to the ErrorCode parameter.



	<p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DID Data Operation interface type (<code>DcmDspDataUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspDataUsePort</code> is set to <code>USE_DATA_ASYNC_FNC</code> or <code>USE_DATA_SYNCH_FNC</code>. <p>Note: The multiplicity of this parameter shall be equal to that of parameter <code>DcmDspDataReadFnc</code>.</p>
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DcmDspDataEcuSignal	
Label	ECU Signal access control function	
Description	<p>Defines the string identifying this signal as an ECU Signal for IOControl operations. This string will be used in order to generate the <code>IoHwAb_Dcm-&#60; DcmDspDataEcuSignal&#62;</code> APIs used for IOControl purposes in the context of a request for UDS Service <code>InputOutputControlByIdentifier</code> (0x2F).</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DID Data Operation interface type (<code>DcmDspDataUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspDataUsePort</code> is set to <code>USE_ECU_SIGNAL</code>. <p>Note: the functionality related to this parameter is not supported by the current implementation.</p>	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	PreCompile: VariantPreCompile	
Origin	AUTOSAR_ECU	

Parameter Name	DcmDspDataFreezeCurrentStateFnc	
Label	Freeze CurrentState function	
Description	<p>Defines the name of the callout used for requesting the Application to freeze the current state of an IOControl signal in the context of a request for UDS Service <code>InputOutputControlByIdentifier</code> (0x2F) with a</p>	



	<p>controlOptionRecord<inputoutputcontrolparameter> parameter value of freezeCurrentState (0x02)</inputoutputcontrolparameter></p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DID Data Operation interface type (<code>DcmDspDataUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspDataUsePort</code> is set to <code>USE_DATA_ASYNCH_FNC</code> or <code>USE_DATA_SYNCH_FNC</code>.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDataGetScalingInfoFnc
Label	Read ScalingInformation function
Description	<p>Defines the name of the callout used for requesting the scaling information of a DID from the Application in the context of a request for UDS service <code>ReadScalingDataByIdentifier</code> (0x24)</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DID Data Operation interface type (<code>DcmDspDataUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspDataUsePort</code> is set to <code>USE_DATA_ASYNCH_FNC</code> or <code>USE_DATA_SYNCH_FNC</code>.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDataReadDataLengthFnc
Label	Read DID data length function
Description	<p>Defines the name of the callout used for requesting the length of a DID signal from the application.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DID Data Operation interface type (<code>DcmDspDataUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspDataUsePort</code> is set to <code>USE_DATA_ASYNCH_FNC</code> or <code>USE_DATA_SYNCH_FNC</code>.
Multiplicity	0..1



Type	FUNCTION-NAME	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspDataReadEcuSignal	
Label	Read ECU Signal data function	
Description	<p>Defines the string identifying this signal as an ECU Signal with Read access. This string will be used in order to generate the <code>IoHwAb_Dcm - Read&#60;EcuSignalName&#62;</code> APIs used when reading the signal data.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DID Data Operation interface type (<code>DcmDspDataUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspDataUsePort</code> is set to <code>USE_ECU_SIGNAL</code>. 	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspDataReadFnc	
Label	Read DID data function	
Description	<p>Defines the name of the callout used for reading the data of this signal from the Application.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DID Data Operation interface type (<code>DcmDspDataUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspDataUsePort</code> is set to <code>USE_DATA_ASYNCH_FNC</code> or <code>USE_DATA_SYNCH_FNC</code>. <p>Note: The multiplicity of this parameter shall be equal to that of parameter <code>DcmDspDataConditionCheckReadFnc</code>.</p>	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspDataResetToDefaultFnc	
----------------	-----------------------------	--



Label	Reset IOControl function
Description	<p>Defines the name of the callout used for requesting the Application to reset an IOControl signal to its default value in the context of a request for UDS Service <code>InputOutputControlByIdentifier</code> (0x2F) with a <code>controlOptionRecord[inputOutputControlParameter]</code> parameter value of <code>resetToDefault</code> (0x01).</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DID Data Operation interface type (<code>DcmDspDataUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspDataUsePort</code> is set to <code>USE_DATA_ASYNCH_FNC</code> or <code>USE_DATA_SYNCH_FNC</code>.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DcmDspDataReturnControlToEcuFnc
Label	Return Control function
Description	<p>Defines the name of the callout used for requesting the Application to return control of an IOControl signal to the ECU in the context of a request for UDS Service <code>InputOutputControlByIdentifier</code> (0x2F) with a <code>controlOptionRecord[inputOutputControlParameter]</code> parameter value of <code>returnControlToEcu</code> (0x00).</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DID Data Operation interface type (<code>DcmDspDataUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspDataUsePort</code> is set to <code>USE_DATA_ASYNCH_FNC</code> or <code>USE_DATA_SYNCH_FNC</code>.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DcmDspDataShortTermAdjustmentFnc
Label	Adjust IOSignal function
Description	Defines the name of the callout used for requesting the Application to adjust an IOControl signal to a given value in the context of a request for UDS Service <code>InputOutputControlByIdentifier</code> (0x2F) with a



	<p>controlOptionRecord<inputoutputcontrolparameter> parameter value of shortTermAdjustment (0x03).</inputoutputcontrolparameter></p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DID Data Operation interface type (<code>DcmDspDataUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspDataUsePort</code> is set to <code>USE_DATA_ASYNCH_FNC</code> or <code>USE_DATA_SYNCH_FNC</code>.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDataSize	
Label	Data Length (bits)	
Description	<p>Defines the length of the data comprising this signal.</p> <p>If this signal has a variable length (its <code>DcmDspDataInfo</code> referenced via <code>DcmDspDataInfoRef</code> has parameter <code>DcmDspDataFixedLength</code> configured to FALSE), then this value represents the maximum length of the signal.</p> <p>If this signal is maximum length (65535 bits) and the AccessInterface is function callout or client-server (DCM_USE_DATA_ASYNCH_CLIENT_SERVER, DCM_USE_DATA_ASYNCH_FNC, DCM_USE_DATA_SYNCH_CLIENT_SERVER or DCM_USE_DATA_SYNCH_FNC), when Endian conversion is required an extra byte is added to the length of the signal.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 65535 bits 	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<p><=65535</p> <hr/> <p>>=1</p>	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	
Parameter Name	DcmDspDataType	
Label	Data Type	



Description	<p>Provide the implementation data type of data belonging to a DID.</p> <p>Note: This configuration is used in order to facilitate Endianness conversion of the data, if this is required (see configuration parameter <code>DcmDslProtocolEndiannessConvEnabled</code>).</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ BOOLEAN ▶ SINT16 ▶ SINT16_N ▶ SINT32 ▶ SINT32_N ▶ SINT8 ▶ SINT8_N ▶ UINT16 ▶ UINT16_N ▶ UINT32 ▶ UINT32_N ▶ UINT8 ▶ UINT8_DYN ▶ UINT8_N
Multiplicity	0..1
Type	ENUMERATION
Range	<p>BOOLEAN</p> <p>SINT16</p> <p>SINT16_N</p> <p>SINT32</p> <p>SINT32_N</p> <p>SINT8</p> <p>SINT8_N</p> <p>UINT16</p> <p>UINT16_N</p> <p>UINT32</p> <p>UINT32_N</p>



	UINT8
	UINT8_DYN
	UINT8_N
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDataUsePort
Label	Data Access Interface
Description	<p>Defines which interface shall be used to access the data of this signal.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ USE_BLOCK_ID: This signal data shall be read from and written to an NvM Block, referenced in DcmDspDataBlockIdRef; ▶ USE_DATA_SYNCH_FNC: This signal data shall be read, written, controlled and the scaling information retrieved by using a synchronous API callout (the operation shall be completed in one single call to the API). Depending on the required functionality, the following callout names shall be configured: <ul style="list-style-type: none"> ▶ For reading data of the signal: <ul style="list-style-type: none"> ▶ DcmDspDataConditionCheckReadFnc ▶ DcmDspDataReadDataLengthFnc ▶ DcmDspDataReadFnc ▶ For writing data of the signal: <ul style="list-style-type: none"> ▶ DcmDspDataWriteFnc ▶ DcmDspDataReadDataLengthFnc ▶ For IOControl operations of the signal: <ul style="list-style-type: none"> ▶ DcmDspDataReturnControlToEcuFnc ▶ DcmDspDataResetToDefaultFnc ▶ DcmDspDataShortTermAdjustmentFnc ▶ DcmDspDataFreezeCurrentStateFnc ▶ For retrieving the scaling information of the signal: <ul style="list-style-type: none"> ▶ DcmDspDataGetScalingInfoFnc ▶ USE_DATA_ASYNC_FNC: This signal data shall be read, written, controlled and the scaling information retrieved by using an asynchronous AUTOSAR ClientServerInterface with asynchronous behaviour (the operation shall be completed in a series of calls to the API, the first call being execut-



ed with an OpStatus set to DCM_INITIAL, and as long as the interface returns E_PENDING, it shall be called with an OpStatus set to DCM_PENDING for all subsequent executions). Depending on the required functionality, the following callout names shall be configured:

- ▶ For reading data of the signal:
 - ▶ DcmDspDataConditionCheckReadFnc
 - ▶ DcmDspDataReadDataLengthFnc
 - ▶ DcmDspDataReadFnc
- ▶ For writing data of the signal:
 - ▶ DcmDspDataWriteFnc
 - ▶ DcmDspDataReadDataLengthFnc
- ▶ For IOControl operations of the signal:
 - ▶ DcmDspDataReturnControlToEcuFnc
 - ▶ DcmDspDataResetToDefaultFnc
 - ▶ DcmDspDataShortTermAdjustmentFnc
 - ▶ DcmDspDataFreezeCurrentStateFnc
- ▶ For retrieving the scaling information of the signal:
 - ▶ DcmDspDataGetScalingInfoFnc
- ▶ USE_DATA_SENDER_RECEIVER: The DCM will access the Data using an Port requiring a SenderReceiverInterface (with isService=false) DataServices_<Data>. The Port is named DataServices_<Data> where <Data> is the name of the container DcmDspData.
- ▶ USE_DATA_SENDER_RECEIVER_AS_SERVICE: The DCM will access the Data using an Port requiring a SenderReceiverInterface (with isService=true) DataServices_<Data>. The Port is named DataServices_<Data> where <Data> is the name of the container DcmDspData.
- ▶ USE_DATA_SYNCH_CLIENT_SERVER: This signal shall be read, written, controlled and the scaling information retrieved by using by using a synchronous AUTOSAR ClientServerInterface with synchronous behaviour (the operation shall be completed in one single call to the operation of the generated interface). The Dcm shall generate the DataServices_<Data> ClientServerInterface, where <Data> is the name of this DcmDspData container. This ClientServerInterface contains the necessary operations for this signal. A Require Port is generated for this interface and shall be connected to a



	<p>compatible Provide Port of an Application Software Component. The name of this Require Port is the same as the name of the ClientServerInterface.</p> <ul style="list-style-type: none"> ▶ USE_DATA_ASYNCH_CLIENT_SERVER: This signal shall be read, written, controlled and the scaling information retrieved by using an asynchronous API (the operation shall be completed in a series of calls to the API, the first call being executed with an OpStatus set to DCM_INITIAL, and as long as the interface returns E_PENDING, it shall be called with an OpStatus set to DCM_PENDING for all subsequent executions). The Dcm shall generate the DataServices_Data; ClientServerInterface, where Data is the name of the container DcmDspData. This ClientServerInterface contains the necessary operations for this signal. A Require Port is generated for this interface and shall be connected to a compatible Provide Port of an Application Software Component. The name of this Require Port is the same as the name of the ClientServerInterface. ▶ USE_ECU_SIGNAL: <i>The functionality related to this option is not supported by the current implementation.</i> This signal shall be read, written and controlled as an ECU Signal by accessing the related interfaces of the Io-HwAb module.
Multiplicity	1..1
Type	ENUMERATION
Range	USE_BLOCK_ID USE_DATA_ASYNCH_CLIENT_SERVER USE_DATA_ASYNCH_FNC USE_DATA_SENDER_RECEIVER USE_DATA_SENDER_RECEIVER_AS_SERVICE USE_DATA_SYNCH_CLIENT_SERVER USE_DATA_SYNCH_FNC USE_ECU_SIGNAL
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDataWriteFnc
Label	Write DID Data function
Description	Defines the name of the callout used for writing the data of this signal to the Application.



	<p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DID Data Operation interface type (<code>DcmDspDataUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspDataUsePort</code> is set to <code>USE_DATA_ASYNC_FNC</code> or <code>USE_DATA_SYNCH_FNC</code>.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	<code>DcmDspDataInfoRef</code>
Label	Reference to the DID Data Information
Description	<p>Reference to the <code>DcmDspDataInfo</code> containing:</p> <ul style="list-style-type: none"> ▶ Information on whether the length of this signal is fixed or not ▶ If Scaling information is available for this signal, the size of the scaling information
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	<code>DcmDspDataBlockIdRef</code>
Label	Reference to the NVRAM blockId
Description	<p>Reference to the NvM Block used to access the data of this signal, if <code>DcmDspDataUsePort</code> is configured to <code>USE_BLOCK_ID</code>.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DID Data Operation interface type (<code>DcmDspDataUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspDataUsePort</code> is set to <code>USE_BLOCK_ID</code>.
Multiplicity	0..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC V1.0.0

Parameter Name	<code>DcmDspDataEndianness</code>
-----------------------	-----------------------------------



Label	DcmDspDataEndianness	
Description	<p>Defines the endianness of the data belonging to a DID in a diagnostic request or response message.</p> <ul style="list-style-type: none"> ▶ BIG_ENDIAN = Most significant byte shall be stored at the lowest address. ▶ LITTLE_ENDIAN = Most significant byte shall be stored at the highest address ▶ OPAQUE = Opaque data endianness 	
Multiplicity	0..1	
Type	ENUMERATION	
Range	BIG_ENDIAN LITTLE_ENDIAN OPAQUE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.32. DcmDspDataInfo

Parameters included	
Parameter name	Multiplicity
DcmDspDataFixedLength	1..1
DcmDspDataScalingInfoSize	0..1

Parameter Name	DcmDspDataFixedLength
Label	Fixed Data Length
Description	<p>This parameter indicates whether the length of the data of the referencing DcmDspData signal is fixed.</p> <p>If the length of data for a signal is variable, then for interfaces related to this signal for reading, writing and control, additional parameters shall be included in the signatures to convey information on:</p> <ul style="list-style-type: none"> ▶ The amount of data received from the Application, in case of a Read operation. ▶ The amount of data received in the service request message, in case of a Write or IOControl operation.



	Range:
	<ul style="list-style-type: none"> ▶ TRUE: The length of the data is fixed. ▶ FALSE: The length of the data is variable.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDataScalingInfoSize	
Label	Size of scaling information	
Description	Defines the size in bytes of the scaling information of the referenced DcmDsp-Data if the ReadScalingDataByIdentifier (0x24) service is used.	
Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Range	<=4294967295 >=0	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.33. DcmDspDid

Containers included		
Container name	Multiplicity	Description
DcmDspDidExtRoe	1..1	<p>Label: External ROE-DID Handling</p> <p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Defines the parameters needed to configure this DID to handle an OnChangeOfDataIdentifier ROE event which is externally generated.</p> <p>If this container is not configured, an ROE event on this DID will be managed internally (the comparison of the previous</p>



Containers included

		<p>and the current values of the DID will be executed internally). If this container is configured, the Application may generate an event by calling either the <code>Dcm_TriggerOnEvent()</code> API. Furthermore, the <code>Dcm</code> shall signal to the Application the activation or deactivation of the ROE event by executing either the <code>DcmDspDidRoeActivateFnc</code> callout, if configured, or by calling the <code>xxx_ActivateEvent</code> operation of the <code>ROEServices ClientServerInterface</code> on the corresponding generated Require Port if the <code>DcmDspDidRoeActivateFnc</code> callout is not configured. (please refer to the <code>DcmDspDidRoeActivateFnc</code> parameter for more details).</p> <p>The generated Require Port for the <code>ROEServices ClientServerInterface</code> is named <code>ROEServices_&#60;SWC&#62;</code> where <code>&#60;SWC&#62;</code> is the name of this container.</p>
DcmDspDidSignal	0..n	<p>Label: DID Signal Configuration</p> <p>Defines:</p> <ul style="list-style-type: none"> ▶ The reference to one <code>DcmDspData</code> container representing the data of this signal. ▶ The position of this signal within the DID.

Parameters included

Parameter name	Multiplicity
DcmDspDidIdentifier	1..1
DcmDspDidUsed	1..1
DcmDspDidInfoRef	1..1
DcmDspDidRef	0..n

Parameter Name	DcmDspDidIdentifier
Label	DID Identifier
Description	<p>Defines the numeric ID of the Data Identifier according to ISO 14229-1.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0x00FF .. 0xF1FF, 0xF900 .. 0xFFFF: Static defined DIDs, any type of access allowed



	<ul style="list-style-type: none"> ▶ 0xF200 .. 0xF2FF: Periodic DIDs, can defined as static or dynamic. If defined as static any type of access allowed, if defined as dynamic only read access allowed ▶ 0xF300 .. 0xF3FF: Dynamic defined DIDs, only read access allowed ▶ 0xF400 .. 0xFEFF: OBD DIDs (partially reserved for future use), only read access allowed <p>Note: When OBD support is not enabled, the DIDs in ranges 0xF400-0xF4FF, 0xF600-0xF6FF and 0xF800-0xF8FF shall be read as usual UDS DIDs, from the values of their configured signals.</p>
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDidUsed	
Label	Enable DID usage	
Description	<p>Defines whether this DID is available for use in the current configuration (for multi-purpose ECUs).</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: This DID is available. ▶ FALSE: This DID is not available. The <code>Dcm</code> behaves as if this DID was not configured. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspDidInfoRef	
Label	Reference to DID Info	
Description	Reference to the DcmDspDidInfo structure for this DID.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile



Origin	AUTOSAR_ECUC
Parameter Name	DcmDspDidRef
Label	Reference to other DIDs
Description	<p>Reference to another <code>DcmDspDid</code>. In case a DID is read and references other DIDs, the referenced DIDs' data shall be read and concatenated to the first DID's data.</p> <p>Note: A referenced DID shall not, in turn, reference any other DIDs.</p>
Multiplicity	0..n
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.2.1.34. `DcmDspDidExtRoe`

Parameters included	
Parameter name	Multiplicity
DcmDspDidRoeActivateFnc	0..1
DcmDspDidRoeEventId	1..1

Parameter Name	DcmDspDidRoeActivateFnc
Label	ROE Activate/Deactivate Function
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Defines the name of the callout used in order to activate/deactivate an ROE event which is managed externally.</p> <p>If this parameter is configured, the <code>Dcm</code> shall signal to the Application the activation or deactivation of the ROE event by executing either the <code>DcmDspDidRoeActivateFnc</code> callout, if configured, or by calling the <code>xxx_ActivateEvent</code> operation of the <code>ROEServices ClientServerInterface</code> on the corresponding generated Require Port if the <code>DcmDspDidRoeActivateFnc</code> callout is not configured.</p> <p>The generated Require Port for the <code>ROEServices ClientServerInterface</code> is named <code>ROEServices_&#60;SWC&#62;</code>; where <code>&#60;SWC&#62;</code> is the name of this container.</p>



Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDidRoeEventId
Label	ROE Event ID
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Defines EventId to be used within the <code>Dcm_TriggerOnEvent()</code> API and the calls to either:</p> <ul style="list-style-type: none"> ▶ The <code>DcmDspDidRoeActivateFnc</code> callout, if configured. ▶ The <code>xxx_ActivateEvent()</code> operation of the ROEServices ClientServerInterface if the <code>DcmDspDidRoeActivateFnc</code> callout is not configured. <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 255
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<p><=255</p> <p>>=0</p>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.2.1.35. DcmDspDidSignal

Parameters included	
Parameter name	Multiplicity
DcmDspDidDataPos	1..1
DcmDspDidDataRef	1..1
Parameter Name	DcmDspDidDataPos



Label	DID Data Position
Description	Defines the position(in bits) of the signal data defined by the <code>DcmDspData</code> referenced via <code>DcmDspDidDataRef</code> within the DID. Range: ▶ 0 .. 65535 bits
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<=65535 >=0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDidDataRef
Label	DID Data Reference
Description	Reference to the <code>DcmDspData</code> container defining the methods for obtaining/writing this DID signal's data.
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.2.1.36. DcmDspDidInfo

Containers included		
Container name	Multiplicity	Description
DcmDspDidAccess	1..1	Label: DID Access Defines the access types (Read, Write or IOControl) available to the referencing <code>DcmDspDid</code> DID and, for each access type, the permissions required.

Parameters included	
Parameter name	Multiplicity

**Parameters included**

DcmDspDD DID Max Elements	1..1
DcmDspDidDynamicallyDefined	1..1

Parameter Name	DcmDspDD DID Max Elements	
Label	Maximum number of source elements	
Description	Maximum number of source elements of a DD DID.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=255 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspDidDynamicallyDefined	
Label	Is the DID dynamically defined?	
Description	Defines if the referencing DcmDspDid DID is dynamically defined. Configure this parameter to TRUE if the referencing DcmDspDid DID is to be defined by a DynamicallyDefineDataIdentifier (0x2C) service request. Range: <ul style="list-style-type: none">▶ TRUE: The referencing DcmDspDid DID is dynamically defined.▶ FALSE: The referencing DcmDspDid DID is statically defined.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.37. DcmDspDidAccess**Containers included**

Container name	Multiplicity	Description



Containers included

DcmDspDidControl	0..1	<p>Label: DID Control</p> <p>Defines the availability of IOControl access for the referencing DcmDspDid DID, IOControl access types and permissions for IOControl access.</p> <p>The available IOControl access types are:</p> <ul style="list-style-type: none"> ▶ FreezeCurrentState: Freezes the current state of the referencing DcmDspDid DID signals. ▶ ReturnControlToEcu: Returns control over the referencing DcmDspDid DID signals to the ECU. ▶ ResetToDefault: Resets the referencing DcmDspDid DID signals to their default states. ▶ ShortTermAdjustment: Adjusts the referencing DcmDspDid DID signals to the values included in the controlOptionRecord. <p>Configure this container if the referencing DcmDspDid DID represents input or output signals or internal parameters that are to be controlled in the context of an InputOutputControlByIdentifier (0x2F) service request.</p>
DcmDspDidRead	0..1	<p>Label: DID Read</p> <p>Defines the availability of Read access for the referencing DcmDspDid DID and permissions for Read access.</p>
DcmDspDidWrite	0..1	<p>Label: DID Write</p> <p>Defines the availability of Write access for the referencing DcmDspDid DID and permissions for Write access.</p>
DcmDspDidReadScalingData	0..1	<p>Label: DID Read Scaling Data</p> <p>Defines the availability of Read Scaling Data access for the referencing DcmDspDid DID and permissions for Read Scaling Data access.</p>

5.2.1.38. DcmDspDidControl

Containers included

Container name	Multiplicity	Description



Containers included

DcmDspDidControlEnable-Mask	0..1	Label: DcmDspDidControlEnableMask The shortname of the container value defines the symbol of the controlMask.
---	------	---

Parameters included

Parameter name	Multiplicity
DcmDspDidControlMask	0..1
DcmDspDidControlMaskSize	0..1
DcmDspDidFreezeCurrentState	0..1
DcmDspDidResetToDefault	0..1
DcmDspDidReturnControlToEcu	0..1
DcmDspDidShortTermAdjustment	0..1
DcmDspDidControlModeRuleRef	0..1
DcmDspDidControlSecurityLevelRef	0..n
DcmDspDidControlSessionRef	0..n

Parameter Name	DcmDspDidControlMask
Label	DcmDspDidControlMask
Description	<p>This indicates the presence of "controlEnableMask" in SWC service interfaces and defines how the Dcm treats a service request.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DCM_CONTROLMASK_EXTERNAL: The control enable mask record shall be forwarded within each interface and is handled externally. ▶ DCM_CONTROLMASK_INTERNAL: The control enable mask record is handled internally and Dcm controls only the included signals. ▶ DCM_CONTROLMASK_NO: No control enable mask handling. <p>DCM_CONTROLMASK_INTERNAL is not supported. To obtain DCM_CONTROLMASK_INTERNAL functionality disable DcmDspDidControlMask and DcmDspDidControlMaskSize. In this case, the allowed mask size is computed internally as large enough to, considered as bitmask, represent all signals configured for the DID, one bit per signal.</p> <p>DCM_CONTROLMASK_NO is not supported. To obtain DCM_CONTROLMASK_NO functionality disable DcmDspDidControlMask and DcmDspDidControlMaskSize this use case, the number of signals of the controlled DID is limited to one.</p>



Multiplicity	0..1	
Type	ENUMERATION	
Default value	DCM_CONTROLMASK_EXTERNAL	
Range	DCM_CONTROLMASK_EXTERNAL DCM_CONTROLMASK_INTERNAL DCM_CONTROLMASK_NO	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspDidControlMaskSize	
Label	DcmDspDidControlMaskSize	
Description	<p>The value defines the size of the controlEnableMaskRecord in bytes.</p> <p>Note: the functionality related to this parameter is not supported by the current implementation.</p>	
Multiplicity	0..1	
Type	INTEGER	
Range	<=4 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspDidFreezeCurrentState	
Label	Enable FreezeCurrentState	
Description	<p>Defines whether FreezeCurrentState access is available for the referencing DcmDspDid DID.</p> <p>Configure this parameter to TRUE if the referencing DcmDspDid DID is to be used by the inputOutputControlByIdentifier (0x2F) service request with the controlOptionRecord[inputOutputControlParameter] parameter set to FreezeCurrentState (0x02)</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: FreezeCurrentState access is available for the referencing DcmDspDid DID. 	



	► FALSE: FreezeCurrentState access is not available for the referencing DcmDspDid DID.
Multiplicity	0..1
Type	BOOLEAN
Default value	true
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDidResetToDefault
Label	Enable ResetToDefault
Description	<p>Defines whether ResetToDefault access is available for the referencing DcmDspDid DID.</p> <p>Configure this parameter to TRUE if the referencing DcmDspDid DID is to be used by the inputOutputControlByIdentifier (0x2F) service request with the controlOptionRecord[inputOutputControlParameter] parameter set to ResetToDefault (0x01)</p> <p>Range:</p> <ul style="list-style-type: none"> ► TRUE: ResetToDefault access is available for the referencing DcmDspDid DID. ► FALSE: ResetToDefault access is not available for the referencing DcmDspDid DID.
Multiplicity	0..1
Type	BOOLEAN
Default value	true
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDidReturnControlToEcu
Label	Enable ReturnControlToEcu
Description	<p>Defines whether ReturnControlToEcu access is available for the referencing DcmDspDid DID.</p> <p>Configure this parameter to TRUE if the referencing DcmDspDid DID is to be used by the inputOutputControlByIdentifier (0x2F) service request with the controlOptionRecord[inputOutputControlParameter] parameter set to ReturnControlToEcu (0x00)</p>



	<p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: ReturnControlToEcu access is available for the referencing DcmDspDid DID. ▶ FALSE: ReturnControlToEcu access is not available for the referencing DcmDspDid DID.
Multiplicity	0..1
Type	BOOLEAN
Default value	true
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDidShortTermAdjustment
Label	Enable ShortTermAdjustment
Description	<p>Defines whether ShortTermAdjustment access is available for the referencing DcmDspDid DID.</p> <p>Configure this parameter to TRUE if the referencing DcmDspDid DID is to be used by the inputOutputControlByIdentifier (0x2F) service request with the controlOptionRecord[inputOutputControlParameter] parameter set to ShortTermAdjustment (0x03)</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: ShortTermAdjustment access is available for the referencing DcmDspDid DID. ▶ FALSE: ShortTermAdjustment access is not available for the referencing DcmDspDid DID.
Multiplicity	0..1
Type	BOOLEAN
Default value	true
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDidControlModeRuleRef
Description	Reference to the DcmDspModeRule to be checked before executing any IOControl operation on the referencing DcmDspDid DID.



	If this parameter is enabled and a <code>DcmDspModeRule</code> is referenced, this Mode Rule shall be evaluated before service execution is allowed to proceed. If the Mode Rule evaluates to <code>TRUE</code> , service processing will be allowed to proceed. If the Mode Rule evaluates to <code>FALSE</code> , service processing is not allowed to proceed and depending on the referenced mode rule a Negative Response Code is returned to the Tester. If this parameter is disabled, no mode rule permission check is performed.
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	<code>DcmDspDidControlSecurityLevelRef</code>
Description	Reference to the <code>DcmDspSecurityRows</code> defining the Security Levels in which IOControl access shall be allowed for the referencing <code>DcmDspDid</code> DID. Note: If there is no reference configured, no check of Security Level is performed, i.e. IOControl access is allowed in any Security Level.
Multiplicity	0..n
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	<code>DcmDspDidControlSessionRef</code>
Description	Reference to the <code>DcmDspSessionRows</code> defining the Diagnostic Sessions in which IOControl access shall be allowed for the referencing <code>DcmDspDid</code> DID. Note: If there is no reference configured, no check of Diagnostic Session is performed, i.e. IOControl access is allowed in any Diagnostic Session.
Multiplicity	0..n
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC



5.2.1.39. DcmDspDidControlEnableMask

Parameters included	
Parameter name	Multiplicity
DcmDspDidControlMaskBitPosition	0..1
Parameter Name	
DcmDspDidControlMaskBitPosition	
Label	
Did Control Mask Bit Position	
Description	
Defines the position of the bit in the controlMask starting from most significant bit (MSB first) to least significant bit. This Bit endianness is identical to the control-Mask in UDS. The DcmDspDidControlMaskSize should be considered for most significant bit.	
Multiplicity	
0..1	
Type	
INTEGER	
Range	
<=31	
>=0	
Configuration class	
PreCompile: VariantPreCompile	
Origin	
AUTOSAR_ECUC	

5.2.1.40. DcmDspDidRead

Parameters included	
Parameter name	Multiplicity
DcmDspDidReadModeRuleRef	0..1
DcmDspDidReadSecurityLevelRef	0..n
DcmDspDidReadSessionRef	0..n
Parameter Name	
DcmDspDidReadModeRuleRef	
Label	
DID Read Mode Rule Reference	
Description	
Reference to the DcmDspModeRule to be checked before executing a Read operation on the referencing DcmDspDid DID. If this parameter is enabled and a DcmDspModeRule is referenced, this Mode Rule shall be evaluated before service execution is allowed to proceed. If the Mode Rule evaluates to TRUE, service processing will be allowed to proceed. If the Mode Rule evaluates to FALSE, service processing is not allowed to pro-	



	ceed and depending on the referenced mode rule a Negative Response Code is returned to the Tester. If this parameter is disabled, no mode rule permission check is performed.
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDidReadSecurityLevelRef
Label	Reference to Security Levels
Description	Reference to the DcmDspSecurityRows defining the Security Levels in which Read access shall be allowed for the referencing DcmDspDid DID. Note: If there is no reference configured, no check of Security Level is performed, i.e. Read access is allowed in any Security Level.
Multiplicity	0..n
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDidReadSessionRef
Label	Reference to Diagnostic Sessions
Description	Reference to the DcmDspSessionRows defining the Diagnostic Sessions in which Read access shall be allowed for the referencing DcmDspDid DID. Note: If there is no reference configured, no check of Diagnostic Session is performed, i.e. Read access is allowed in any Diagnostic Session.
Multiplicity	0..n
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.2.1.41. DcmDspDidWrite

Parameters included	
Parameter name	Multiplicity

**Parameters included**

DcmDspDidWriteModeRuleRef	0..1
DcmDspDidWriteSecurityLevelRef	0..n
DcmDspDidWriteSessionRef	0..n

Parameter Name	DcmDspDidWriteModeRuleRef	
Label	DID Write Mode Rule Reference	
Description	<p>Reference to the <code>DcmDspModeRule</code> to be checked before executing a Read operation on the referencing <code>DcmDspDid</code> DID.</p> <p>If this parameter is enabled and a <code>DcmDspModeRule</code> is referenced, this Mode Rule shall be evaluated before service execution is allowed to proceed. If the Mode Rule evaluates to <code>TRUE</code>, service processing will be allowed to proceed. If the Mode Rule evaluates to <code>FALSE</code>, service processing is not allowed to proceed and depending on the referenced mode rule a Negative Response Code is returned to the Tester. If this parameter is disabled, no mode rule permission check is performed.</p>	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DcmDspDidWriteSecurityLevelRef	
Label	Reference to Security Levels	
Description	<p>Reference to the <code>DcmDspSecurityRows</code> defining the Security Levels in which Write access shall be allowed for the referencing <code>DcmDspDid</code> DID.</p> <p>Note: If there is no reference configured, no check of Security Level is performed, i.e. Write access is allowed in any Security Level.</p>	
Multiplicity	0..n	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DcmDspDidWriteSessionRef	
Label	Reference to Diagnostic Sessions	
Description	Reference to the <code>DcmDspSessionRows</code> defining the Diagnostic Sessions in which Write access shall be allowed for the referencing <code>DcmDspDid</code> DID.	



	Note: If there is no reference configured, no check of Diagnostic Session is performed, i.e. Write access is allowed in any Diagnostic Session.	
Multiplicity	0..n	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.42. DcmDspDidReadScalingData

Parameters included	
Parameter name	Multiplicity
DcmDspDidReadScalingDataSecurityLevelRef	0..n
DcmDspDidReadScalingDataSessionRef	0..n

Parameter Name	DcmDspDidReadScalingDataSecurityLevelRef	
Label	Reference to Security Levels	
Description	Reference to the <code>DcmDspSecurityRows</code> defining the Security Levels in which Read Scaling Data access shall be allowed for the referencing <code>DcmDspDid</code> DID. Note: If there is no reference configured, no check of Security Level is performed, i.e. Read Scaling Data access is allowed in any Security Level.	
Multiplicity	0..n	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspDidReadScalingDataSessionRef	
Label	Reference to Diagnostic Sessions	
Description	Reference to the <code>DcmDspSessionRows</code> defining the Diagnostic Sessions in which Read Scaling Data access shall be allowed for the referencing <code>DcmDspDid</code> DID. Note: If there is no reference configured, no check of Diagnostic Session is performed, i.e. Read Scaling Data access is allowed in any Diagnostic Session.	
Multiplicity	0..n	
Type	REFERENCE	



Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.43. DcmDspDidRange

Parameters included	
Parameter name	Multiplicity
DcmDspDidRangeHasGaps	1..1
DcmDspDidRangeldentifierLowerLimit	1..1
DcmDspDidRangeldentifierUpperLimit	1..1
DcmDspDidRangesDidAvailableFnc	0..1
DcmDspDidRangeMaxDataLength	1..1
DcmDspDidRangeReadDidFnc	0..1
DcmDspDidRangeUsePort	1..1
DcmDspDidRangeWriteDidFnc	0..1
DcmDspDidRangeInfoRef	1..1

Parameter Name	DcmDspDidRangeHasGaps	
Label	Did Range Has Gaps	
Description	<p>Defines whether the DID range contains gaps or not.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: The DID range contains gaps. ▶ FALSE: The DID range does not contain gaps. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspDidRangeldentifierLowerLimit
Label	Did Range Lower Limit Id
Description	Defines the lower limit of the DID range.



	► 0 .. 65535
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDidRangIdentifierUpperLimit	
Label	Did Range Upper Limit Id	
Description	<p>Defines the upper limit of the DID range.</p> <p>Range:</p> <p>► 0 .. 65535</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<p><=65535</p> <p>>=0</p>	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspDidRangesDidAvailableFnc	
Label	Did Available Function	
Description	<p>Defines the name of the callout used for checking if a certain DID is available within the range or not.</p> <p>Dependency on parameter(s):</p> <p>► DID Range Operation interface type (<code>xxx_IsDidAvailable</code>): The configuration of this parameter is only relevant if <code>DcmDspDidRangeUsePort</code> is set to FALSE.</p>	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	



Parameter Name	DcmDspDidRangeMaxDataLength	
Label	Max Data Length	
Description	Maximum data length in bytes. Range: ▶ 1 .. 65535 Bytes	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=65535 >=1	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DcmDspDidRangeReadDidFnc	
Label	Did Range Read Did Function	
Description	Defines the name of the callout used for requesting the data range value of a DID. Dependency on parameter(s): ▶ DID Range Operation interface type (<code>Xxx_ReadDidData</code>): The configuration of this parameter is only relevant if <code>DcmDspDidRangeUsePort</code> is set to FALSE.	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DcmDspDidRangeUsePort	
Label	Did Range Use Port	
Description	Defines the type of the interface that shall be used to implement the <code>IsDidAvailable</code> , <code>ReadDidData</code> and <code>WriteDidData</code> operations of this DID Range in the context of a <code>ReadDataByIdentifier</code> (0x22) or <code>WriteDataByIdentifier</code> (0x2E) diagnostic service requests. Range:	



	<ul style="list-style-type: none"> ▶ TRUE: The <code>IsDidAvailable</code>, <code>ReadDidData</code> and <code>WriteDidData</code> operations of this DID Range are realized using a synchronous AUTOSAR ClientServerInterface with asynchronous behaviour (the operation shall be completed in a series of calls to the API, the first call being executed with an <code>OpStatus</code> set to <code>DCM_INITIAL</code>, and as long as the interface returns <code>E_PENDING</code>, it shall be called with an <code>OpStatus</code> set to <code>DCM_PENDING</code> for all subsequent executions). The <code>Dcm</code> generates the <code>DataServices_DIDRange_&#60;Range&#62;</code> ClientServerInterface. This interface shall implement the <code>IsDidAvailable</code> (if the range has gaps), <code>ReadDidData</code> (if read access is enabled) and <code>WriteDidData</code> (if write access is enabled) operations of this DID Range. A Require Port is generated for this interface and shall be connected to a compatible Provide Port of an Application Software Component. The name of this Require Port is the same as the name of the ClientServerInterface. ▶ FALSE: The operations are realized using an asynchronous API callout (the operation shall be completed in a series of calls to the API, the first call being executed with an <code>OpStatus</code> set to <code>DCM_INITIAL</code>, and as long as the interface returns <code>E_PENDING</code>, it shall be called with an <code>OpStatus</code> set to <code>DCM_PENDING</code> for all subsequent executions). The following callout names shall be configured for each operation: <ul style="list-style-type: none"> ▶ For operation <code>IsDidAvailable</code>: <code>DcmDspDidRangeIsDidAvailableFnc</code> ▶ For operation <code>ReadDidData</code>: <code>DcmDspDidRangeReadDidFnc</code> ▶ For operation <code>WriteDidData</code>: <code>DcmDspDidRangeWriteDidFnc</code>
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDidRangeWriteDidFnc
Label	Did Range write Did Function
Description	<p>Defines the name of the callout used for writing the data range value of a DID.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DID Range Operation interface type (<code>xxx_WriteDidData</code>): The configuration of this parameter is only relevant if <code>DcmDspDidRangeUsePort</code> is set to FALSE.



Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDidRangeInfoRef	
Label	Did Range Information Reference	
Description	Reference to DcmDspDidInfo containing information on this DID Range.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.44. DcmDspMemory

Containers included		
Container name	Multiplicity	Description
DcmDspAddressAndLength-FormatIdentifier	0..1	<p>Defines the configuration for AddressAndLengthFormatIdentifiers.</p> <p>An AddressAndLengthFormatIdentifier defines the number of bytes interpreted as the <code>memoryAddress</code> and <code>memorySize</code> parameters in the request messages for services <code>ReadMemoryByAddress</code> (0x23), <code>WriteMemoryByAddress</code> (0x3D), <code>DynamicallyDefineDataIdentifier</code>, <code>sub-function defineByMemoryAddress</code> (0x2C-0x02), <code>RequestDownload</code> (0x34) and <code>RequestUpload</code> (0x35). This parameter is a one byte value where:</p> <ul style="list-style-type: none"> ▶ Bits 3 .. 0 encode how many subsequent bytes, following the <code>AddressAndLengthFormatIdentifier</code> in the request message, encode the <code>memoryAddress</code> and ▶ Bits 7 .. 4 encode how many subsequent bytes, following the <code>memoryAddress</code> in the request message encode the <code>memorySize</code>.
DcmDspMemoryIdInfo	0..n	Label: Memory Id Info



Containers included

		<p>Defines a subunit of memory, identified by the the <code>DcmDspMemoryIdValue</code> identifier and contains the configuration of Write/Read Memory Ranges and permissions to access these ranges. These can further be used to represent, per example, different memory devices.</p> <p>Configure <code>DcmDspMemoryIdInfo</code> elements in order to use the <code>ReadMemoryByAddress</code> (0x23), <code>WriteMemoryByAddress</code> (0x3D) or configure dynamic DIDs by using the <code>DynamicallyDefineDataIdentifier</code> (0x2C) service request with the <code>defineByMemoryAddress</code> (0x02) sub-function.</p>
--	--	---

Parameters included

Parameter name	Multiplicity
DcmDspUseMemoryId	1..1
DcmDspMemoryRangeHighNotIncluded	0..1

Parameter Name	DcmDspUseMemoryId
Label	Memory Use Id
Description	<p>Defines whether the <code>Dcm</code> shall use the <code>MemoryIdentifier</code> parameter in the calls to the <code>Dcm_WriteMemory</code> and <code>Dcm_ReadMemory</code> callouts in the context of servicing a <code>ReadMemoryByAddress</code> (0x23) or <code>WriteMemoryByAddress</code> (0x3D) diagnostic service request.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: The <code>Dcm_WriteMemory</code> and <code>Dcm_ReadMemory</code> callouts shall be called with the <code>MemoryIdentifier</code> parameter set to the configured <code>DcmDspMemoryIdValue</code> value. ▶ FALSE: The <code>Dcm_WriteMemory</code> and <code>Dcm_ReadMemory</code> callouts shall be called with the <code>MemoryIdentifier</code> parameter set to 0.
Multiplicity	1..1
Type	BOOLEAN
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspMemoryRangeHighNotIncluded
Label	Define ranges using border values



Description	<p>Defines whether if the defined memory high limits of the defined ranges are interpreted either as belonging to the defined memory range or not belonging to the defined memory range.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: the values of DcmDspReadMemoryRangeHigh, DcmDspWriteMemoryRangeHigh, DcmDspWriteMemoryRangeByLabelHigh and DcmDspReadMemoryRangeByLabelHigh are not part of the defined memory range, e.g. (DcmDspReadMemoryRangeHigh-1) is the last address of the defined read range. ▶ FALSE: the values of DcmDspReadMemoryRangeHigh, DcmDspWriteMemoryRangeHigh, DcmDspWriteMemoryRangeByLabelHigh and DcmDspReadMemoryRangeByLabelHigh are part of the defined memory range, e.g. DcmDspReadMemoryRangeHigh is the last address of the defined read range. <p>i.e: The consecutive memory ranges R1 from 100 to 199 and R2 from 200 to 299 can be described as:</p> <ul style="list-style-type: none"> ▶ DcmDspMemoryRangeHighNotIncluded = FALSE R1:[100 - 199], R2: [200 - 299], or ▶ DcmDspMemoryRangeHighNotIncluded = TRUE R1:[100 - 200), R2: [200 - 300) <p>Purpose of defining the ranges with borders instead of as a pair of values is to allow simplification by taking advantage of the functionality of defining memory ranges using labels (<code>DcmDspWriteMemoryRangeByLabelInfo</code> and <code>DcmDspReadMemoryRangeByLabelInfo</code>).</p> <p>In this way the possibility is open to define the same LABEL for a high read/write limit of a range R(n) and the low read/write limit of the consecutive range R(n+1).</p> <p>By doing this, a list can be prepared inside the customised header file configured under <code>DcmHeaderFileInclusion</code> which will contain only the addresses between the defined ranges.</p> <p>This has advantages such as ease of modification, and reducing the probability of configuring unused memory regions.</p>
Multiplicity	0..1
Type	BOOLEAN
Default value	false



Configuration class	PreCompile:	VariantPreCompile
----------------------------	--------------------	-------------------

5.2.1.45. DcmDspAddressAndLengthFormatIdentifier

Parameters included	
Parameter name	Multiplicity
DcmDspSupportedAddressAndLengthFormatIdentifier	1..n

Parameter Name	DcmDspSupportedAddressAndLengthFormatIdentifier	
Label	Supported AddressAndLengthFormatIdentifier	
Description	<p>Defines the supported AddressAndLengthFormatIdentifiers.</p> <p>An AddressAndLengthFormatIdentifier defines the number of bytes interpreted as the memoryAddress and memorySize parameters in the request messages for services ReadMemoryByAddress (0x23), WriteMemoryByAddress (0x3D), DynamicallyDefineDataIdentifier (0x2C), RequestDownload (0x34) and RequestUpload (0x35). This parameter is a one byte value where:</p> <ul style="list-style-type: none"> ▶ Bits 3 .. 0 encode how many subsequent bytes, following the AddressAndLengthFormatIdentifier in the request message, encode the memoryAddress and ▶ Bits 7 .. 4 encode how many subsequent bytes, following the memoryAddress in the request message encode the memorySize. <p>Configure the supported AddressAndLengthFormatIdentifiers here.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 255 	
Multiplicity	1..n	
Type	INTEGER	
Default value	0	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	



5.2.1.46. DcmDspMemoryIdInfo

Containers included		
Container name	Multiplicity	Description
DcmDspReadMemoryRangeInfo	0..n	<p>Label: Read Memory Range Information</p> <p>Defines the configuration of a range of memory addresses available for reading in the context of this <code>DcmDspMemoryIdInfo</code>, as well as the necessary permissions for performing a Read operation.</p> <p>If a Read operation needs to be performed on a certain memory address using the <code>ReadMemoryByAddress</code> (0x23) service, the Memory Range received in the request (<code>memoryAddress .. memoryAddress + memorySize -1</code>) will be checked if it is inside the allowed memory ranges, by checking all the configured <code>DcmDspReadMemoryRangeLow .. DcmDspReadMemoryRangeHigh</code> intervals or the <code>DcmDspReadMemoryRangeByLabelLow .. DcmDspReadMemoryRangeByLabelHigh</code> parameters for each <code>DcmDspReadMemoryRangeByLabelInfo</code> container. If not, a <code>GeneralReject</code> (0x31) Negative Response Code is transmitted.</p>
DcmDspReadMemoryRangeByLabelInfo	0..n	<p>Label: Read Memory Range By Label Information</p> <p>Defines the configuration of a range of labeled memory addresses available for reading in the context of this <code>DcmDspMemoryIdInfo</code>, as well as the necessary permissions for performing a Read operation.</p> <p>If a Read operation needs to be performed on a certain memory address using the <code>ReadMemoryByAddress</code> (0x23) service, the Memory Range received in the request (<code>memoryAddress .. memoryAddress + memorySize -1</code>) will be checked if it is inside the allowed memory ranges, by checking all the configured <code>DcmDspReadMemoryRangeLow .. DcmDspReadMemoryRangeHigh</code> intervals, or the <code>DcmDspReadMemoryRangeByLabelLow .. DcmDspReadMemoryRangeByLabelHigh</code> parameters for each <code>DcmDspReadMemoryRangeByLabelInfo</code> container. If not, a <code>GeneralReject</code> (0x31) Negative Response Code is transmitted.</p>
DcmDspWriteMemoryRangeInfo	0..n	<p>Label: Write Memory Range Information</p>



Containers included

		<p>Defines the configuration of a range of memory addresses available for writing in the context of this <code>DcmDspMemoryIdInfo</code>, as well as the necessary permissions for performing a Write operation.</p> <p>If a Write operation needs to be performed on a certain memory address using the <code>WriteMemoryByAddress</code> (0x3D), the Memory Range received in the request (<code>memoryAddress .. memoryAddress + memorySize -1</code>) will be checked if it is inside the allowed memory ranges, by checking all the configured <code>DcmDspWriteMemoryRangeLow .. DcmDspWriteMemoryRangeHigh</code> intervals or the <code>DcmDspWriteMemoryRangeByLabelLow .. DcmDspWriteMemoryRangeByLabelHigh</code> parameters for each <code>DcmDspReadMemoryRangeByLabelInfo</code> container. If not, a <code>GeneralReject</code> (0x31) Negative Response Code is transmitted.</p>
<u>DcmDspWriteMemoryRange-ByLabelInfo</u>	0..n	<p>Label: Write Memory Range Information</p> <p>Defines the configuration of a range of labeled memory addresses available for reading in the context of this <code>DcmDspMemoryIdInfo</code>, as well as the necessary permissions for performing a Write operation.</p> <p>If a Write operation needs to be performed on a certain memory address using the <code>WriteMemoryByAddress</code> (0x3D) service, the Memory Range received in the request (<code>memoryAddress .. memoryAddress + memorySize -1</code>) will be checked if it is inside the allowed memory ranges, by checking all the configured <code>DcmDspWriteMemoryRangeLow .. DcmDspWriteMemoryRangeHigh</code> intervals, or the <code>DcmDspWriteMemoryRangeByLabelLow .. DcmDspWriteMemoryRangeByLabelHigh</code> parameters for each <code>DcmDspWriteMemoryRangeByLabelInfo</code> container. If not, a <code>GeneralReject</code> (0x31) Negative Response Code is transmitted.</p>

Parameters included

Parameter name	Multiplicity
<u>DcmDspMemoryIdValue</u>	0..1

Parameter Name	DcmDspMemoryIdValue
Label	Memory Id Value



Description	Defines the memory device identifier used.
	Dependency on parameter(s):
	Usage of the Memory ID parameter (<code>DcmDspUseMemoryId</code>): The configuration of this parameter is only relevant if the usage of the Memory ID parameter is enabled (<code>DcmDspUseMemoryId</code> is set to TRUE).
	Range:
	<ul style="list-style-type: none"> ▶ 0 .. 255
Multiplicity	0..1
Type	INTEGER
Default value	0
Range	<p><=255</p> <hr/> <p>>=0</p>
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

5.2.1.47. DcmDspReadMemoryRangeInfo

Parameters included	
Parameter name	Multiplicity
DcmDspReadMemoryRangeHigh	1..1
DcmDspReadMemoryRangeLow	1..1
DcmDspReadMemoryRangeModeRuleRef	0..1
DcmDspReadMemoryRangeSecurityLevelRef	0..n

Parameter Name	DcmDspReadMemoryRangeHigh
Label	Upper Limit of Read Memory Range
Description	<p>Defines the upper limit of this Memory Range.</p> <p>The range for this value depends on the setting of the <code>DcmDspMemoryRangeHighNotIncluded</code> parameter</p> <p>Range (<code>DcmDspMemoryRangeHighNotIncluded</code> set to FALSE):</p> <ul style="list-style-type: none"> ▶ 0 .. 4294967294



	Range (DcmDspMemoryRangeHighNotIncluded set to TRUE): ► 1 .. 4294967295
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspReadMemoryRangeLow	
Label	Lower Limit of Read Memory Range	
Description	Defines the lower limit of this Memory Range. Range: ► 0 .. 4294967294	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=4294967294 >=0	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspReadMemoryRangeModeRuleRef	
Label	Reference to read DcmDspModeRule	
Description	Reference to the DcmDspModeRule to be checked before either: ► executing any Read access on an address within this Memory Range via the ReadMemoryByAddress (0x23) service. ► defining a Dynamically Defined DID via the DynamicallyDefineDataIdentifier (0x2C) service request with the defineByMemoryAddress (0x02) sub-function with an address within this range If a Memory Address within this Memory Range needs to be read from, if this parameter is enabled and a DcmDspModeRule is referenced, this Mode Rule shall be evaluated before service execution is allowed to proceed. If the Mode Rule evaluates to TRUE, service processing will be allowed to proceed. If the Mode	



	Rule evaluates to FALSE, service processing is not allowed to proceed and depending on the referenced mode rule a Negative Response Code is returned to the Tester. If this parameter is disabled, no mode rule permission check is performed.
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspReadMemoryRangeSecurityLevelRef	
Label	Reference to Security Levels	
Description	<p>Reference to the <code>DcmDspSecurityRow</code>s defining the Security Levels in which either:</p> <ul style="list-style-type: none"> ▶ executing a Read access on an address within this Memory Range via the <code>ReadMemoryByAddress</code> (0x23) service or ▶ defining a Dynamically Defined DID via the <code>DynamicallyDefineDataIdentifier</code> (0x2C) service request with the <code>defineByMemoryAddress</code> (0x02) sub-function with an address within this range <p>is allowed.</p> <p>Note: If there is no reference configured, no check of Security Level is performed, i.e. the mentioned operations are allowed in any Security Level.</p>	
Multiplicity	0..n	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.48. DcmDspReadMemoryRangeByLabelInfo

Parameters included	
Parameter name	Multiplicity
DcmDspReadMemoryRangeByLabelHigh	1..1
DcmDspReadMemoryRangeByLabelLow	1..1
DcmDspReadMemoryRangeModeRuleRef	0..1
DcmDspReadMemoryRangeSecurityLevelRef	0..n



Parameter Name	DcmDspReadMemoryRangeByLabelHigh	
Label	Upper Limit of Read Memory Range by Label	
Description	Defines the High memory address as label (string) of a range allowed for reading.	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspReadMemoryRangeByLabelLow	
Label	Lower Limit of Read Memory Range by Label	
Description	Defines the Low memory address as label (string) of a range allowed for reading.	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspReadMemoryRangeModeRuleRef	
Label	Reference to read DcmDspModeRule	
Description	<p>Reference to the <code>DcmDspModeRule</code> to be checked before either:</p> <ul style="list-style-type: none"> ▶ executing any Read access on an address within this Memory Range via the <code>ReadMemoryByAddress</code> (0x23) service. ▶ defining a Dynamically Defined DID via the <code>DynamicallyDefineDataIdentifier</code> (0x2C) service request with the <code>defineByMemoryAddress</code> (0x02) sub-function with an address within this range <p>If a Memory Address within this Memory Range needs to be read from, if this parameter is enabled and a <code>DcmDspModeRule</code> is referenced, this Mode Rule shall be evaluated before service execution is allowed to proceed. If the Mode Rule evaluates to <code>TRUE</code>, service processing will be allowed to proceed. If the Mode Rule evaluates to <code>FALSE</code>, service processing is not allowed to proceed and depending on the referenced mode rule a Negative Response Code is returned to the Tester. If this parameter is disabled, no mode rule permission check is performed.</p>	
Multiplicity	0..1	



Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspReadMemoryRangeSecurityLevelRef	
Label	Reference to Security Levels	
Description	<p>Reference to the DcmDspSecurityRows defining the Security Levels in which either:</p> <ul style="list-style-type: none"> ▶ executing a Read access on an address within this Memory Range via the ReadMemoryByAddress (0x23) service or ▶ defining a Dynamically Defined DID via the DynamicallyDefineDataIdentifier (0x2C) service request with the defineByMemoryAddress (0x02) sub-function with an address within this range <p>is allowed.</p> <p>Note: If there is no reference configured, no check of Security Level is performed, i.e. the mentioned operations are allowed in any Security Level.</p>	
Multiplicity	0..n	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.49. DcmDspWriteMemoryRangeInfo

Parameters included	
Parameter name	Multiplicity
DcmDspWriteMemoryRangeHigh	1..1
DcmDspWriteMemoryRangeLow	1..1
DcmDspWriteMemoryRangeModeRuleRef	0..1
DcmDspWriteMemoryRangeSecurityLevelRef	0..n

Parameter Name	DcmDspWriteMemoryRangeHigh
Label	Upper Limit of Write Memory Range
Description	Defines the upper limit of this Memory Range.



	<p>The range for this value depends on the setting of the DcmDspMemoryRangeHighNotIncluded parameter</p> <p>Range (DcmDspMemoryRangeHighNotIncluded set to FALSE):</p> <ul style="list-style-type: none"> ▶ 0 .. 4294967294 <p>Range (DcmDspMemoryRangeHighNotIncluded set to TRUE):</p> <ul style="list-style-type: none"> ▶ 1 .. 4294967295
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspWriteMemoryRangeLow
Label	Lower Limit of Write Memory Range
Description	<p>Defines the lower limit of this Memory Range.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 4294967294
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	≤ 4294967294 ≥ 0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspWriteMemoryRangeModeRuleRef
Label	Reference to write DcmDspModeRule
Description	<p>Reference to the DcmDspModeRule to be checked before executing any Write access on an address belonging to this Memory Range via the WriteMemory-ByAddress (0x3D) service.</p> <p>If a Memory Address within this Memory Range needs to be written to, if this parameter is enabled and a DcmDspModeRule is referenced, this Mode Rule shall</p>



	be evaluated before service execution is allowed to proceed. If the Mode Rule evaluates to <code>TRUE</code> , service processing will be allowed to proceed. If the Mode Rule evaluates to <code>FALSE</code> , service processing is not allowed to proceed and depending on the referenced mode rule a Negative Response Code is returned to the Tester. If this parameter is disabled, no mode rule permission check is performed.
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspWriteMemoryRangeSecurityLevelRef	
Label	Reference to Security Levels	
Description	Reference to the <code>DcmDspSecurityRow</code> s defining the Security Levels in which executing a Write access on an address within this Memory Range via the <code>ReadMemoryByAddress</code> (0x23) service is allowed. Note: If there is no reference configured, no check of Security Level is performed, i.e. this operation is allowed in any Security Level.	
Multiplicity	0..n	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.50. DcmDspWriteMemoryRangeByLabelInfo

Parameters included	
Parameter name	Multiplicity
DcmDspWriteMemoryRangeByLabelHigh	1..1
DcmDspWriteMemoryRangeByLabelLow	1..1
DcmDspWriteMemoryRangeModeRuleRef	0..1
DcmDspWriteMemoryRangeSecurityLevelRef	0..n

Parameter Name	DcmDspWriteMemoryRangeByLabelHigh
Label	Upper Limit of Write Memory Range by Label



Description	Defines the High memory address as label (string) of a range allowed for reading.	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DcmDspWriteMemoryRangeByLabelLow	
Label	Lower Limit of Write Memory Range by Label	
Description	Defines the Low memory address as label (string) of a range allowed for reading.	
Multiplicity	1..1	
Type	STRING	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DcmDspWriteMemoryRangeModeRuleRef	
Label	Reference to read DcmDspModeRule	
Description	<p>Reference to the <code>DcmDspModeRule</code> to be checked before either:</p> <ul style="list-style-type: none"> ▶ executing any Write access on an address within this Memory Range via the <code>WriteMemoryByAddress</code> (0x23) service. ▶ defining a Dynamically Defined DID via the <code>DynamicallyDefineDataIdentifier</code> (0x2C) service request with the <code>defineByMemoryAddress</code> (0x02) sub-function with an address within this range <p>If a Memory Address within this Memory Range needs to be read from, if this parameter is enabled and a <code>DcmDspModeRule</code> is referenced, this Mode Rule shall be evaluated before service execution is allowed to proceed. If the Mode Rule evaluates to <code>TRUE</code>, service processing will be allowed to proceed. If the Mode Rule evaluates to <code>FALSE</code>, service processing is not allowed to proceed and depending on the referenced mode rule a Negative Response Code is returned to the Tester. If this parameter is disabled, no mode rule permission check is performed.</p>	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile



Origin	AUTOSAR_ECUC
Parameter Name	DcmDspWriteMemoryRangeSecurityLevelRef
Label	Reference to Security Levels
Description	<p>Reference to the <code>DcmDspSecurityRow</code>s defining the Security Levels in which either:</p> <ul style="list-style-type: none"> ▶ executing a Write access on an address within this Memory Range via the <code>WriteMemoryByAddress</code> (0x23) service or ▶ defining a Dynamically Defined DID via the <code>DynamicallyDefineDataIdentifier</code> (0x2C) service request with the <code>defineByMemoryAddress</code> (0x02) sub-function with an address within this range <p>is allowed.</p> <p>Note: If there is no reference configured, no check of Security Level is performed, i.e. the mentioned operations are allowed in any Security Level.</p>
Multiplicity	0..n
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.2.1.51. DcmDspPeriodicDidTransmission

Parameters included	
Parameter name	Multiplicity
DcmDspMaxPeriodicDidScheduler	0..1
DcmDspPeriodicDidSamplingBufferSize	0..1
DcmDspPeriodicDidTransmissionRingEntries	0..1
DcmDspPeriodicDDIDMaxLength	0..1

Parameter Name	DcmDspMaxPeriodicDidScheduler
Label	DcmDspMaxPeriodicDidScheduler
Description	Defines the maximum number of <code>periodicDataIdentifiers</code> that can be scheduled concurrently
Multiplicity	0..1
Type	INTEGER



Range	<=255
	>=1
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspPeriodicDidSamplingBufferSize
Label	DcmDspPeriodicDidSamplingBufferSize
Description	Defines the size of a single buffer to which periodic DIDs are sampled. A buffer can hold at most a complete UUDT message and is therefore limited to 8 octets, the first octet being the lower part of the periodic data identifier and the remaining 7 bytes the actual data sample generated by reading that data identifier. These buffers are part of a transmission ring. The number of individual buffers in that transmission ring is given by DcmDspPeriodicDidTransmissionRingEntries
Multiplicity	0..1
Type	INTEGER
Default value	8
Range	<=8
	>=2
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspPeriodicDidTransmissionRingEntries
Label	DcmDspPeriodicDidTransmissionRingEntries
Description	Defines the number of entries in the transmission ring used to store the periodic DID samples. One such entry represents a single buffer of size DcmDspPeriodicDidSamplingBufferSize
Multiplicity	0..1
Type	INTEGER
Range	<=65535
	>=1
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspPeriodicDDIDMaxLength
-----------------------	------------------------------------



Label	Maximum data length of periodic DDDID
Description	<p>Specified the maximum data length that dynamically defined periodic DID can have. This is relevant when sending periodic DDDIDs as UUDT messages directly to the IF layer of the communication stack. The data generated by a periodic DDDID must fit into one frame of the bus on which it is sent. For the CAN bus for example the max size of a frame is 8 octets and because the periodic UUDT message carries the PDID as the first byte max length for the payload data can be 7 octets. Therefore, when sending UUDT on CAN <code>DcmDspPeriodicDD DIDMaxLength</code> may be configured as 7</p> <p>Note: This limitation is applicable only to dynamically defined data identifiers that will be scheduled to be periodically ready via UDS service <code>ReadDataByPeriodicIdentifier</code> (0x2A).</p>
Multiplicity	0..1
Type	INTEGER
Default value	7
Range	<p><=65535</p> <p>>=1</p>
Configuration class	<p>PreCompile: VariantPreCompile</p>
Origin	Elektrobit Automotive GmbH

5.2.1.52. DcmDspPeriodicTransmission

Parameters included	
Parameter name	Multiplicity
DcmDspPeriodicTransmissionSlowRate	0..1
DcmDspPeriodicTransmissionMediumRate	0..1
DcmDspPeriodicTransmissionFastRate	0..1

Parameter Name	DcmDspPeriodicTransmissionSlowRate
Label	DcmDspPeriodicTransmissionSlowRate
Description	<p>Defines the transmission rate of the requested <code>periodicDataIdentifiers</code> to be used if the parameter <code>transmissionMode</code> given in the <code>ReadDataByPeriodicID</code> request is equal to "0x01" ("sendAtSlowRate").</p> <p>Note: This parameter value must be given in seconds and has to be configured as a multiple of <code>DcmTaskTime</code>. A negative value or zero are invalid inputs.</p>



Multiplicity	0..1
Type	FLOAT
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspPeriodicTransmissionMediumRate
Label	DcmDspPeriodicTransmissionMediumRate
Description	Defines the transmission rate of the requested periodicDataIdentifiers to be used if the parameter transmissionMode given in the ReadDataByPeriodicID request is equal to "0x02" ("sendAtMediumRate"). Note: This parameter value must be given in seconds and has to be configured as a multiple of DcmTaskTime. A negative value or zero are invalid inputs.
Multiplicity	0..1
Type	FLOAT
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspPeriodicTransmissionFastRate
Label	DcmDspPeriodicTransmissionFastRate
Description	Defines the transmission rate of the requested periodicDataIdentifiers to be used if the parameter transmissionMode given in the ReadDataByPeriodicID request is equal to "0x03" ("sendAtFastRate"). Note: This parameter value must be given in seconds and has to be configured as a multiple of DcmTaskTime. A negative value or zero are invalid inputs.
Multiplicity	0..1
Type	FLOAT
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.2.1.53. DcmDspPid

Containers included		
Container name	Multiplicity	Description



Containers included

DcmDspPidData	1..255	Label: PID Data Defines the parameters that control how the PID data is obtained.
DcmDspPidSupportInfo	0..256	Label: PID Support Information Defines the support information for the PID.

Parameters included

Parameter name	Multiplicity
DcmDspPidIdentifier	1..1
DcmDspPidService	1..1
DcmDspPidSize	1..1
DcmDspPidUsed	1..1

Parameter Name	DcmDspPidIdentifier
Label	PID Identifier
Description	<p>Defines the unique identifier of the PID according to ISO_14229-1 and ISO 15031-5.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 255 <p>Note: PID 02 is only supported in Service 0x02, it should not be configured since it is always available.</p> <p>Note : For PIDs where Dem provides the function, DcmDspPidDataUsePort for that PID should be set to USE_DATA_SYNCH_FNC and DcmDspPid-DataReadFnc shall point to the function Dem_DcmReadDataOfPID<NN> where <NN> represents the Id of the PID.</p>
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspPidService
Label	PID Service



Description	Defines whether this PID is used for OBD service \$01 or/and \$02.
	Range:
	<ul style="list-style-type: none"> ▶ DCM_SERVICE_01: This PID is only accessible via service \$01 ▶ DCM_SERVICE_01_02: This PID is accessible via both service \$01 and \$02 ▶ DCM_SERVICE_02: This PID is only accessible via service \$02
Multiplicity	1..1
Type	ENUMERATION
Range	DCM_SERVICE_01 DCM_SERVICE_01_02 DCM_SERVICE_02
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspPidSize
Label	PID Length
Description	Defines the length of the PID in Bytes. Range:
	<ul style="list-style-type: none"> ▶ 0 .. 255 Bytes
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspPidUsed
Label	Enable PID
Description	Defines whether this PID is available for use in the current configuration (for multi-purpose ECUs). Range:

- ▶ TRUE: This PID is available.
- ▶ FALSE: This PID is not available. The `Dcm` behaves as if this DID was not configured.



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

5.2.1.54. DcmDspPidData

Containers included		
Container name	Multiplicity	Description
DcmDspPidDataSupportInfo	0..1	<p>Label: PID Data Support Information</p> <p>Defines the supported information for this PID Data.</p>
DcmDspPidService01	0..1	<p>Label: PID Service 01 Configuration</p> <p>Defines the configuration parameters of the PID for use with OBD service \$01 - Request Current Powertrain diagnostic Data.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ PID service usage (DcmDspPidService): This container is only available if <code>DcmDspPidService</code> is configured to either <code>DCM_SERVICE_01</code> or <code>DCM_SERVICE_01_02</code>.
DcmDspPidService02	0..1	<p>Label: PID Service 02 Configuration</p> <p>Defines the configuration parameters of the PID for use with OBD service \$02 - Request Power Train FreezeFrame Data.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ PID service usage (DcmDspPidService): This container is only available if <code>DcmDspPidService</code> is configured to either <code>DCM_SERVICE_02</code> or <code>DCM_SERVICE_01_02</code>. <p>Note: Configuration of this parameter has no bearing on functionality of service \$02.</p>

Parameters included

Parameter name	Multiplicity

**Parameters included**

DcmDspPidDataPos	1..1
DcmDspPidDataSize	1..1

Parameter Name	DcmDspPidDataPos
Label	PID Data Position
Description	<p>Defines the position of PID Data in the response.</p> <p>Note: The data will not start at position 0 in case a support information is available(for bit-mapped PIDs).</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 2040 bits
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspPidDataSize
Label	PID Data Size
Description	<p>Defines the length of data associated to the PID in bits.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 2040 bits
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.2.1.55. DcmDspPidDataSupportInfo**Parameters included**

Parameter name	Multiplicity
-----------------------	---------------------



Parameters included

DcmDspPidDataSupportInfoBit	1..1
DcmDspPidDataSupportInfoRef	1..1

Parameter Name	DcmDspPidDataSupportInfoBit	
Label	Supported Information Bit	
Description	Defines the bit of the SupportInfo. Range: ► 0 .. 255 bits	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspPidDataSupportInfoRef	
Label	Reference to Supported Information	
Description	Reference to the supported information DcmDspPidSupportInfo.	
Multiplicity	1..1	
Type	REFERENCE	
Range	node:paths(..../..../..//DcmDspPidSupportInfo/*)	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.56. DcmDspPidService01

Parameters included	
Parameter name	Multiplicity
DcmDspPidDataReadFnc	0..1
DcmDspPidDataUsePort	1..1

Parameter Name	DcmDspPidDataReadFnc
-----------------------	-----------------------------



Label	Read PID Data Function	
Description	<p>Defines the name of the callout used for reading the PID data.</p> <p>Note : This parameter is relevant only if DcmDspPidDataUsePort = USE_DATA_SYNCH_FNC.</p> <p>Note : For PIDs where Dem provides the function, DcmDspPidDataUsePort for that PID should be set to USE_DATA_SYNCH_FNC and DcmDspPidDataReadFnc shall point to the function Dem_DcmReadDataOfPID<NN> where <NN> represents the Id of the PID.</p>	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DcmDspPidDataUsePort	
Label	Read PID Data using Port interface	
Description	<p>Indicates if DCM uses the RTE port or DSP function to get the PID data value.</p> <p>Note : The DCM module currently does not support the usage of a sender receiver port to get the PID data values, therefore the option USE_DATA_SENDER_RECEIVER can not be configured.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ USE_DATA_SYNCH_FNC: The DCM will use the function defined in DcmDspPidDataReadFnc to get the PID data value. ▶ USE_DATA_SYNCH_CLIENT_SERVER: The DCM will have an R-Port requiring the interface DataServices_{Data}. The R-Port is named DataServices_{Data} where {Data} is the name of the container DcmDspPidData. <p>Note : For PIDs where Dem provides the function, DcmDspPidDataUsePort for that PID should be set to USE_DATA_SYNCH_FNC and DcmDspPidDataReadFnc shall point to the function Dem_DcmReadDataOfPID<NN> where <NN> represents the Id of the PID.</p>	
Multiplicity	1..1	
Type	ENUMERATION	
Range	USE_DATA_SYNCH_CLIENT_SERVER USE_DATA_SYNCH_FNC	
Configuration class	VariantPreCompile:	VariantPreCompile



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

5.2.1.57. DcmDspPidService02

Parameters included	
Parameter name	Multiplicity
DcmDspPidDataDemRef	0..1

Parameter Name	DcmDspPidDataDemRef	
Label	DEM PID Data Reference	
Description	<p>Reference to the <code>DemPidDataElement</code> holding the PID Data in <code>Dem</code> configuration.</p> <p>This parameter links the <code>Dcm</code> PID and <code>Dem</code> PID configuration for Mode \$02.</p> <p>Note: Configuration of this parameter has no bearing on functionality of service 02.</p>	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.58. DcmDspPidSupportInfo

Parameters included	
Parameter name	Multiplicity
DcmDspPidSupportInfoLen	1..1
DcmDspPidSupportInfoPos	1..1

Parameter Name	DcmDspPidSupportInfoLen	
Label	Length of Support Information	
Description	<p>Defines the length of the supported information in Bytes.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 255 Bytes 	



Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspPidSupportInfoPos	
Label	Position of Support Information	
Description	Defines the position of the supported information in the response in Bytes. Range: ► 0 .. 255 Bytes	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.59. DcmDspReadGenericInformation

Parameters included		
Parameter name		Multiplicity
DcmDspMaxDemGetDiagnosticDataConsecutiveExecutions		0..1

Parameter Name	DcmDspMaxDemGetDiagnosticDataConsecutiveExecutions	
Label	Maximum allowed consecutive Data fetch from DEM	
Description	This configuration parameter sets the maximum number of consecutive executions of the function fetching diagnostic data, in order to allow for a break-down of the execution time of service 0xAF (Read Generic Information) and avoid blocking conditions when requested to fetch all of the Data Records.	
Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Range	<=20	



	>=1
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.2.1.60. DcmDspRequestControl

Parameters included	
Parameter name	Multiplicity
DcmDspRequestControlFnc	0..1
DcmDspRequestControlInBufferSize	1..1
DcmDspRequestControlOutBufferSize	1..1
DcmDspRequestControlTestId	1..1
DcmDspRequestControlUsed	1..1
DcmDspRequestControlUsePort	1..1

Parameter Name	DcmDspRequestControlFnc
Label	Request control function
Description	<p>Defines the name of the callout used for the request control functionality of OBD Service \$08.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Request control operation interface type (<code>DcmDspRequestControlUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspRequestControlUsePort</code> is set to <code>USE_REQUESTCONTROL_SYNCH_FNC</code>
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspRequestControlInBufferSize
Label	Input Buffer Size
Description	Defines the size of the input buffer for the <code>RequestControl</code> operation in Bytes.



	<ul style="list-style-type: none"> ▶ 0 .. 5 Bytes <p>Note: The available buffer length is restricted from the AUTOSAR-defined range of 0 .. 4294967295 to 0..5. The ISO specifies that no RequestControl function takes more than 5 parameters.</p>
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<p><=5</p> <hr/> <p>>=0</p>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspRequestControlOutBufferSize
Label	Output Buffer Size
Description	<p>Defines the size of the output buffer for the RequestControl operation in Bytes.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 5 Bytes <p>Note: The available buffer length is restricted from the AUTOSAR-defined range of 0 .. 4294967295 to 0..5. The ISO specifies that no RequestControl function takes more than 5 parameters.</p>
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<p><=5</p> <hr/> <p>>=0</p>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspRequestControlTestId
Label	TestId
Description	Defines the TestId requested in the \$08 - Request Control of On-Board System, Test or Component service request represented by this DcmDspRequestControl



	Note: According to the SWS the TestId is with the range of 0..255 That range however also covers the availability TIDs {0x00 0x020 0x40 0x60 0x80 0xA0 0xC0 0xE0}. The availability TIDs serve to report what RequestControl TIDs are available and should therefore not be part of the configured identifiers.
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspRequestControlUsed
Label	Enable RequestControl TID usage
Description	Note : This is a Vendor Specific parameter for Request Control in the Dcm. Allow to activate or deactivate the usage of a service true = service available, false = service not available
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspRequestControlUsePort
Label	Request Control Interface
Description	Defines the type of the interface that shall be used to implement the RequestControl operation defining the sub-function of OBD service RequestControl (0x08) Range: <ul style="list-style-type: none"> ▶ USE_REQUESTCONTROL_SYNCH_FNC: The operation shall be implemented using a synchronous API. The following callout names shall be configured: <ul style="list-style-type: none"> ▶ DcmDspRequestControlFnc ▶ USE_REQUESTCONTROL_SYNCH_CLIENT_SERVER: The operation shall be implemented using a synchronous AUTOSAR ClientServerInterface. The Dcm generates the RequestControl_&#60;TID&#62; ClientServerInterface where &#60;TID&#62; is the name of this DcmDspRequestControl container. This ClientServerInterface contains



	the RequestControl operation. A Require Port is generated for this interface and shall be connected to a compatible Provide Port of an Application Software Component. The name of this Require Port is the same as the name of the ClientServerInterface.
Multiplicity	1..1
Type	ENUMERATION
Default value	USE_REQUESTCONTROL_SYNCH_CLIENT_SERVER
Range	USE_REQUESTCONTROL_SYNCH_CLIENT_SERVER USE_REQUESTCONTROL_SYNCH_FNC
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.2.1.61. DcmDspRoe

Containers included		
Container name	Multiplicity	Description
DcmDspRoeEvent	1..255	This container contains a list of all supported Roe eventType-Records which are accepted by this ECU. At most one DcmDspRoeEvent container is allowed to define a DcmDspRoeEventProperties container with the choice DcmDspRoeOnDTCStatusChange.
DcmDspRoeEventWindow-Time	1..255	This container configures the available EventWindowTime in this Ecu. This container contains a sub-set of EventWindow-Times supported by the Dcm, to limit the usage of the Ecu resources.

Parameters included	
Parameter name	Multiplicity
DcmDspRoeBlockIdRef	0..1
DcmDspRoeInterMessageTime	1..1

Parameter Name	DcmDspRoeBlockIdRef
Label	Reference to the NVRAM blockId to store ROE data
Description	Reference to the NvM block used to store persistent ROE configuration data.
Multiplicity	0..1



Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmDspRoeInterMessageTime			
Label	Inter-Message Time			
Description	<p>Provide the minimum time in seconds between two transmissions of ROE event. It is used for the delay between two different consecutive Roe transmissions.</p> <p>In the current implementation, if a new event triggers a transmission before the DcmDspRoeInterMessageTime expires, the new event is ignored.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 5 seconds 			
Multiplicity	1..1			
Type	FLOAT			
Default value	0.0			
Range	<table border="0"> <tr> <td><=5</td> </tr> <tr> <td>>=0</td> </tr> </table>		<=5	>=0
<=5				
>=0				
Configuration class	VariantPreCompile:	VariantPreCompile		
Origin	AUTOSAR_ECUC			

5.2.1.62. DcmDspRoeEvent

Containers included		
Container name	Multiplicity	Description
DcmDspRoeEventProperties	1..1	<p>Label: Roe Pre Configuration Service To Respond</p> <p>This container contains the properties of Roe eventTypeRecords. In one DcmDspRoeEventProperties container one DcmDspRoeOnDTCstatusChange or DcmDspRoeOnChangeOfDataIdentifier container shall be defined.</p>

Parameters included	
Parameter name	Multiplicity

**Parameters included**

DcmDspRoeEventId	1..1
DcmDspRoeInitialEventStatus	1..1

Parameter Name	DcmDspRoeEventId
Label	EventId
Description	EventId for a global identification of this ROE event. It is used within APIs Dcm_TriggerOnEvent (), ModeDeclarationGroup DcmResponseOn-Event_<RoeEventID> and ModeSwitchInterface SchM_Switch_<bsnp>_ - DcmResponseOnEvent_<RoeEventId> The Event Ids should be zero-based and consecutive.
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile

Parameter Name	DcmDspRoeInitialEventStatus
Label	Initial Roe status
Description	<p>Initial Roe status of this RoeEvent</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DCM_ROE_CLEARED: If the EventWindowTime is active over power cycles and not timed out, the Dcm shall reactivate all RoeEvents which were active in the default session during the last power cycle as soon as the non-volatile information is available. If a valid StartResponseOnEvent request is received with a storageState set to StoreEvent and the EventWindowTime supports the StorageState in a previous power cycle, the RoeEvent shall change to 'ROE started' state as soon as the non-volatile data is available. ▶ DCM_ROE_STOPPED: switch to 'ROE stopped' state immediately in the initialisation. The Dcm will behave like this RoeEvent was set-up with StorageState set to 'StoreEvent' and EventWindowTime set to infinity <p>NOTE: At most one event can be in STOPPED state for each event type (one for OnChangeOfDataIdentifier and one for onDTCStatusChange).</p>
Multiplicity	1..1
Type	ENUMERATION
Range	DCM_ROE_CLEARED DCM_ROE_STOPPED



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

5.2.1.63. DcmDspRoeEventProperties

Containers included		
Container name	Multiplicity	Description
DcmDspRoeOnChangeOfDataIdentifier	1..1	This container contains configuration of a eventTypeRecord <code>onChangeOfDataIdentifier</code> accepted by this ECU.
DcmDspRoeOnDTCStatusChange	1..1	This container contains configuration of a eventTypeRecord <code>onDTCStatusChange</code> accepted by this ECU. Please note that currently are no additional parameters for <code>DcmDspRoeOnDTCStatusChange</code> are defined. Therefore the existence of the container denotes the choice.

5.2.1.64. DcmDspRoeOnChangeOfDataIdentifier

Parameters included	
Parameter name	Multiplicity
DcmDspRoeDidRef	1..1

Parameter Name	DcmDspRoeDidRef	
Description	Reference to a Did which is watched.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile

5.2.1.65. DcmDspRoeOnDTCStatusChange

Parameters included	
Parameter name	Multiplicity
DcmDspRoeEventPreconfiguredDTCMask	1..1



Parameter Name	DcmDspRoeEventPreconfiguredDTCMask	
Label	Pre-configured DTC Mask for DTC Events	
Description	Defines the default value used for DTCMaks when an OnDTCstatusChange event is preconfigured Range: ▶ 0x00 .. 0xFF	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=255 >=0	
Configuration class	VariantPreCompile:	VariantPreCompile

5.2.1.66. DcmDspRoeEventWindowTime

Parameters included	
Parameter name	Multiplicity
DcmDspRoeEventWindowTime	1..1
DcmDspRoeStorageState	1..1

Parameter Name	DcmDspRoeEventWindowTime
Description	Value of the EventWindowTime Range: ▶ DCM_ROE_EVENT_WINDOW_CURRENT_AND_FOLLOWING_CYCLE: ▶ DCM_ROE_EVENT_WINDOW_CURRENT_CYCLE: ▶ DCM_ROE_EVENT_WINDOW_INFINITE: <i>Note: In the current release only DCM_ROE_EVENT_WINDOW_INFINITE is supported.</i>
Multiplicity	1..1
Type	ENUMERATION
Range	DCM_ROE_EVENT_WINDOW_CURRENT_AND_FOLLOWING_CYCLE



	DCM_ROE_EVENT_WINDOW_CURRENT_CYCLE
	DCM_ROE_EVENT_WINDOW_INFINITE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspRoeStorageState
Label	Evaluate StorageStateBit
Description	If this parameter is set to TRUE the StorageStateBit will be evaluated if this EventWindowTime is requested. Range: <ul style="list-style-type: none">▶ TRUE: StorageStateBit will be evaluated.▶ FALSE: StorageStateBit will not be evaluated.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.2.1.67. DcmDspRoutine

Parameters included	
Parameter name	Multiplicity
DcmDspRequestResultsRoutineFnc	0..1
DcmDspRequestResultsRoutineSupported	1..1
DcmDspRoutineFixedLength	1..1
DcmDspRoutineIdentifier	1..1
DcmDspRoutineUsed	1..1
DcmDspRoutineUsePort	1..1
DcmDspRoutineUsePortASRVersion	0..1
DcmDspStartRoutineFnc	0..1
DcmDspStopRoutineFnc	0..1
DcmDspStopRoutineSupported	1..1

**Parameters included**

DcmDspRoutineInfoRef	1..1
DcmDspRoutineTidRef	0..1

Parameter Name	DcmDspRequestResultsRoutineFnc
Label	Function for requestRoutineResults
Description	<p>Defines the name of the Callout used for requesting the results of running a Diagnostic Routine from the Application, in the context of a RoutineControl (0x31) diagnostic service request with the <code>requestRoutineResults</code> (0x03) sub-function.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Routine Operation interface type (<code>DcmDspRoutineUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspRoutineUsePort</code> is set to FALSE.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspRequestResultsRoutineSupported
Label	requestRoutineResults Supported
Description	<p>Defines whether the <code>requestRoutineResults</code> (0x03) sub-function of the in the RoutineControl (0x31) service and its associated operations are available for this Diagnostic Routine.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: the <code>requestRoutineResults</code> (0x03) sub-function of the in the RoutineControl (0x31) service and its associated operations are available for this Diagnostic Routine. ▶ FALSE: the <code>requestRoutineResults</code> (0x03) sub-function of the in the RoutineControl (0x31) service and its associated operations are not available for this Diagnostic Routine. <p>Note: Further configuration of sub-functions under <code>DcmDsdSubService</code> is not necessary.</p>
Multiplicity	1..1



Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspRoutineFixedLength	
Label	Fixed Length Optional Record	
Description	<p>Defines if the length of data in the <code>routineControlOptionRecord</code> in <code>RoutineControl</code> (0x31)</p> <ul style="list-style-type: none"> ▶ <code>startRoutine</code> (0x01) ▶ <code>stopRoutine</code> (0x02) <p>diagnostic service requests and the <code>routineStatusRecord</code> in <code>RoutineControl</code> (0x31)</p> <ul style="list-style-type: none"> ▶ <code>startRoutine</code> (0x01) ▶ <code>stopRoutine</code> (0x02) ▶ <code>requestRoutineResults</code> (0x03) <p>diagnostic service responses is fixed.</p> <p>This configuration parameter is deprecated by RFC 57860 and does no longer exists in AUTOSAR version 4.2.1</p> <p>This configuration parameter controls whether the <code>currentDataLength</code> parameter is expected or not in interfaces called for Diagnostic Routine Operations.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: Length of the <code>routineControlOptionRecord</code> and the <code>routineStatusRecord</code> is fixed. The parameter <code>currentDataLength</code> is not expected in the signatures of Diagnostic Routine Operation APIs. ▶ FALSE: Length of the <code>routineControlOptionRecord</code> and the <code>routineStatusRecord</code> is variable. The parameter <code>currentDataLength</code> is not expected in the signatures of Diagnostic Routine Operation APIs. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile



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

Parameter Name	DcmDspRoutineIdentifier
Label	Routine Identifier
Description	Defines the unique numeric identifier of the Diagnostic Routine (to be used as the <code>routineIdentifier</code> in <code>RoutineControl</code> (0x31) diagnostic service requests according to ISO_14229-1). Range: <ul style="list-style-type: none">▶ 0 .. 65535
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspRoutineUsed
Label	Enable Routine
Description	Defines whether this Diagnostic Routine is available for use in the current configuration (for multi-purpose ECUs). Range: <ul style="list-style-type: none">▶ TRUE: This Diagnostic Routine is available.▶ FALSE: This Diagnostic Routine is not available. The Dcm behaves behave as if this Diagnostic Routine was not configured.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspRoutineUsePort
Label	Enable port interface
Description	Defines the type of the interface that shall be used to implement the <code>start</code> , <code>stop</code> and <code>requestResults</code> operations of this Diagnostic Routine in the context of a <code>RoutineControl</code> (0x31) diagnostic service request. Range:



	<ul style="list-style-type: none"> ▶ TRUE: The start, stop and requestResults operations of this Diagnostic Routine shall be implemented using by using a synchronous AUTOSAR ClientServerInterface with asynchronous behaviour(the operation shall be completed in a series of calls to the API, the first call being executed with an OpStatus set to DCM_INITIAL, and as long as the interface returns E_PENDING, it shall be called with an OpStatus set to DCM_PENDING for all subsequent executions). The Dcm generates the RoutineServices_&#60;RoutineName&#62; ClientServerInterface, where &#60;RoutineName&#62; is the name of this DcmDspRoutine container. This interface shall implement the start and stop and requestResults operations if they are configured for this Diagnostic Routine. A Require Port is generated for this interface and shall be connected to a compatible Provide Port of an Application Software Component. The name of this Require Port is the same as the name of the ClientServerInterface. ▶ FALSE: The Dcm uses an asynchronous API (the operation shall be completed in a series of calls to the API, the first call being executed with an OpStatus set to DCM_INITIAL, and as long as the interface returns E_PENDING, it shall be called with an OpStatus set to DCM_PENDING for all subsequent executions). The following callout names shall be configured for each operation, if used: <ul style="list-style-type: none"> ▶ For operation startRoutine: DcmDspStartRoutineFnc ▶ For operation stopRoutine: DcmDspStopRoutineFnc ▶ For operation requestRoutineResults: DcmDspRequestResultsRoutineFnc
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspRoutineUsePortASRVersion
Label	Autosar Version
Description	Selects the AUTOSAR version to be used by the RoutineServices interface.
Multiplicity	0..1
Type	ENUMERATION
Default value	Default
Range	Default



	AUTOSAR_40
	AUTOSAR_42
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspStartRoutineFnc
Label	Function for startRoutine
Description	<p>Defines the name of the Callout used for starting a Diagnostic Routine in the Application, in the context of a <code>RoutineControl</code> (0x31) diagnostic service request with the <code>startRoutine</code> (0x01) sub-function.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Routine Operation interface type (<code>DcmDspRoutineUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspRoutineUsePort</code> is set to FALSE.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DcmDspStopRoutineFnc
Label	Function for stopRoutine
Description	<p>Defines the name of the Callout used for stopping a Diagnostic Routine in the Application, in the context of a <code>RoutineControl</code> (0x31) diagnostic service request with the <code>stopRoutine</code> (0x02) sub-function.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Routine Operation interface type (<code>DcmDspRoutineUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspRoutineUsePort</code> is set to FALSE
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DcmDspStopRoutineSupported
-----------------------	-----------------------------------



Label	Stop Routine Supported
Description	<p>Defines whether the <code>stopRoutine</code> (0x02) sub-function of the <code>RoutineControl</code> (0x31) service and its associated operations are available for this Diagnostic Routine.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: the <code>requestRoutineResults</code> (0x02) sub-function of the <code>RoutineControl</code> (0x31) service and its associated operations are available for this Diagnostic Routine. ▶ FALSE: the <code>requestRoutineResults</code> (0x02) sub-function of the <code>RoutineControl</code> (0x31) service and its associated operations are not available for this Diagnostic Routine. <p>Note: Further configuration of sub-functions under <code>DcmDsdSubService</code> is not necessary.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspRoutineInfoRef
Label	Reference to Routine Information
Description	<p>Reference to the <code>DcmDspRoutineInfo</code> container defining the signals and permissions of this Diagnostic Routine:</p> <ul style="list-style-type: none"> ▶ Diagnostic Routine Security Level, Diagnostic Session and Mode Rule authorization settings ▶ <code>startRoutine</code> (0x01) sub-function input and output signal configuration ▶ <code>stopRoutine</code> (0x02) sub-function input and output signal configuration ▶ <code>requestRoutineResults</code> (0x03) sub-function output signal configuration
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspRoutineTidRef
-----------------------	----------------------------



Label	RequestControl Reference	
Description	<p>Reference to a <code>DcmDspRequestControl</code> instance to be used when reading data through the <code>RoutineControl</code> interface. <code>DcmDspRequestControl</code> contains the configuration (parameters) of the "Request control of on-board system, test or component" service (Service \$08).</p> <p>Note: If this parameter is disabled, the parameters <code>DcmDspRoutineAuthorization</code>, <code>DcmDspStartRoutine</code> and <code>DcmDspRoutineUsePort</code> shall be mandatory. If this parameter is enabled and a <code>DcmDspRequestControl</code> is referenced, the parameters <code>DcmDspRoutineAuthorization</code>, <code>DcmDspRoutineRequest</code>, <code>DcmDspRoutineStop</code>, <code>DcmDspStartRoutine</code> and <code>DcmDspRoutineUsePort</code> shall be disabled.</p>	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

5.2.1.68. DcmDspRoutineInfo

Containers included		
Container name	Multiplicity	Description
DcmDspRoutineAuthorization	1..1	<p>Label: Routine Authorization</p> <p>Defines the configuration for authorization of operations on the referencing <code>DcmDspRoutine</code> Diagnostic Routine.</p>
DcmDspRoutineRequestResIn	0..1	<p>Label: RequestRoutineResults Input Configuration</p> <p>Defines the input parameter of <code>requestResults</code> sub-service for <code>routineControl</code> service.</p> <p>Note : This is a Vendor Specific parameter.</p>
DcmDspRoutineRequestRe-sOut	0..1	<p>Label: RequestResult Output Configuration</p> <p>Defines the configuration for the <code>routineStatusRecord</code> output parameter present in the response to the <code>requestRoutineResults</code> (0x03) sub-function of the <code>RoutineControl</code> (0x31) diagnostic service for the referencing <code>DcmDspRoutine</code> Diagnostic Routine.</p>
DcmDspRoutineStopIn	0..1	Label: Stop Routine Input Configuration



Containers included		
		Defines the configuration for the <code>routineControlOptionRecord</code> input parameter present in the request for the <code>stopRoutine</code> (0x02) sub-function of the <code>RoutineControl</code> (0x31) diagnostic service for the referencing <code>DcmDspRoutine</code> Diagnostic Routine.
DcmDspRoutineStopOut	0..1	<p>Label: Stop Routine Output Configuration</p> <p>Defines the configuration for the <code>routineStatusRecord</code> output parameter present in the response to the <code>stopRoutine</code> (0x02) sub-function of the <code>RoutineControl</code> (0x31) diagnostic service for the referencing <code>DcmDspRoutine</code> Diagnostic Routine.</p>
DcmDspStartRoutineIn	0..1	<p>Label: Start Routine Input Configuration</p> <p>Defines the configuration for the <code>routineControlOptionRecord</code> input parameter present in the request for the <code>startRoutine</code> (0x01) sub-function of the <code>RoutineControl</code> (0x31) diagnostic service for the referencing <code>DcmDspRoutine</code> Diagnostic Routine.</p>
DcmDspStartRoutineOut	0..1	<p>Label: Start Routine Output Configuration</p> <p>Defines the configuration for the <code>routineStatusRecord</code> output parameter present in the response to the <code>startRoutine</code> (0x01) sub-function of the <code>RoutineControl</code> (0x31) diagnostic service for the referencing <code>DcmDspRoutine</code> Diagnostic Routine.</p>

5.2.1.69. DcmDspRoutineAuthorization

Parameters included	
Parameter name	Multiplicity
DcmDspRoutineModeRuleRef	0..1
DcmDspRoutineSecurityLevelRef	0..n
DcmDspRoutineSessionRef	0..n

Parameter Name	DcmDspRoutineModeRuleRef
Label	Mode Rule Reference



Description	Reference to the <code>DcmDspModeRule</code> to be checked before executing any operation on the referencing <code>DcmDspRoutine</code> Diagnostic Routine.
	If this parameter is enabled and a <code>DcmDspModeRule</code> is referenced, this Mode Rule shall be evaluated before service execution is allowed to proceed. If the Mode Rule evaluates to <code>TRUE</code> , service processing will be allowed to proceed. If the Mode Rule evaluates to <code>FALSE</code> , service processing is not allowed to proceed and depending on the referenced mode rule a Negative Response Code is returned to the Tester. If this parameter is disabled, no mode rule permission check is performed.
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspRoutineSecurityLevelRef
Label	Reference to Security Levels
Description	Reference to the <code>DcmDspSecurityRows</code> defining the Security Levels in which operations shall be allowed on the referencing <code>DcmDspRoutine</code> Diagnostic Routine. Note: If there is no reference configured, no check of Security Level is performed, i.e. operations are allowed in any Security Level.
Multiplicity	0..n
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspRoutineSessionRef
Label	Reference to Diagnostic Sessions
Description	Reference to the <code>DcmDspSessionRows</code> defining the Diagnostic Sessions in which operations shall be allowed on the referencing <code>DcmDspRoutine</code> Diagnostic Routine. Note: If there is no reference configured, no check of Diagnostic Session is performed, i.e. operations are allowed in any Diagnostic Session.
Multiplicity	0..n
Type	REFERENCE



Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.70. DcmDspRoutineRequestResIn

Containers included		
Container name	Multiplicity	Description
DcmDspRoutineRequestResInSignal	1..n	Label: RequestResult Signal Description Defines the description of input signals used in routineControl service.

5.2.1.71. DcmDspRoutineRequestResInSignal

Parameters included	
Parameter name	Multiplicity
DcmDspRoutineSignalLength	1..1
DcmDspRoutineSignalPos	1..1
DcmDspRoutineSignalType	1..1

Parameter Name	DcmDspRoutineSignalLength
Label	Signal Length
Description	Defines, in number of bits, the signal length in the routineControl request/response.
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<=65535 >=0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspRoutineSignalPos
-----------------------	-------------------------------



Label	Signal Position	
Description	Configures the signal position in the routineControl request/response, in bits.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=65535 >=0	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspRoutineSignalType
Label	Signal Type
Description	<p>Configures the signal type in the routineControl request/response.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ BOOLEAN ▶ SINT16 ▶ SINT32 ▶ SINT8 ▶ UINT16 ▶ UINT32 ▶ UINT8 ▶ SINT16_N ▶ SINT32_N ▶ SINT8_N ▶ UINT16_N ▶ UINT32_N ▶ UINT8_N ▶ VARIABLE_LENGTH
Multiplicity	1..1
Type	ENUMERATION
Range	BOOLEAN SINT16



	SINT32
	SINT8
	UINT16
	UINT32
	UINT8
	SINT16_N
	SINT32_N
	SINT8_N
	UINT16_N
	UINT32_N
	UINT8_N
	VARIABLE_LENGTH
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.2.1.72. DcmDspRoutineRequestResOut

Containers included		
Container name	Multiplicity	Description
DcmDspRoutineRequestResOutSignal	1..n	<p>Label: RequestResult Output Signal Description</p> <p>Defines the configuration of a single signal of the <code>routineStatusRecord</code> output parameter which is present in the response to the <code>requestRoutineResults</code> (0x03) sub-function of the <code>RoutineControl</code> (0x31) diagnostic service, for the referencing <code>DcmDspRoutine</code> Diagnostic Routine.</p>

5.2.1.73. DcmDspRoutineRequestResOutSignal

Parameters included	
Parameter name	Multiplicity
DcmDspRoutineSignalLength	1..1
DcmDspRoutineSignalPos	1..1

**Parameters included**

DcmDspRoutineSignalType	1..1
---	------

Parameter Name	DcmDspRoutineSignalLength		
Label	Signal Length		
Description	<p>Defines the length of the signal, when the "signal length" is not variable, if DcmDspRoutineSignalType is not configured to VARIABLE_LENGTH, or the maximum length the variable "length signal" is allowed to reach, when DcmDspRoutineSignalType is configured to VARIABLE_LENGTH</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 1 .. 65535 bits <p>Note: The available length range is restricted from the AUTOSAR-defined range of 0 .. 65535 as signals of length 0 are not usable.</p>		
Multiplicity	1..1		
Type	INTEGER		
Default value	1		
Range	<table border="0"> <tr> <td><=65535</td> </tr> <tr> <td>>=1</td> </tr> </table>	<=65535	>=1
<=65535			
>=1			
Configuration class	VariantPreCompile: VariantPreCompile		
Origin	AUTOSAR_ECUC		

Parameter Name	DcmDspRoutineSignalPos		
Label	Signal Position		
Description	<p>Defines the starting position of this signal in the routineStatusRecord output parameter of the diagnostic service response.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 65535 bits 		
Multiplicity	1..1		
Type	INTEGER		
Default value	0		
Range	<table border="0"> <tr> <td><=65535</td> </tr> <tr> <td>>=0</td> </tr> </table>	<=65535	>=0
<=65535			
>=0			
Configuration class	VariantPreCompile: VariantPreCompile		



Origin	AUTOSAR_ECUC
Parameter Name	DcmDspRoutineSignalType
Label	Signal Type
Description	<p>Defines the data type of data belonging to this signal.</p> <p>This configuration is used in order to facilitate Endianness conversion of the data, if this is required (see configuration parameter <code>DcmDslProtocolEndiannessConvEnabled</code>).</p> <p>This parameter also configures whether the length of this signal is variable or not. If the type of the signal is <code>VARIABLE_LENGTH</code>:</p> <ul style="list-style-type: none"> ▶ also configure <code>DcmDspRoutineFixedLength</code> to TRUE for the <code>DcmDspRoutine</code> referencing this <code>DcmDspRoutineInfo</code>. ▶ the length of the signal is supplied by the Application via the <code>currentDataLength</code> parameter of the <code>requestRoutineResults</code> operation interface. <p>Range:</p> <ul style="list-style-type: none"> ▶ BOOLEAN ▶ SINT16 ▶ SINT32 ▶ SINT8 ▶ UINT16 ▶ UINT32 ▶ UINT8 ▶ SINT16_N ▶ SINT32_N ▶ SINT8_N ▶ UINT16_N ▶ UINT32_N ▶ UINT8_N ▶ VARIABLE_LENGTH
Multiplicity	1..1
Type	ENUMERATION
Range	BOOLEAN



	SINT16
	SINT32
	SINT8
	UINT16
	UINT32
	UINT8
	SINT16_N
	SINT32_N
	SINT8_N
	UINT16_N
	UINT32_N
	UINT8_N
	VARIABLE_LENGTH
Configuration class	VariantPreCompile:
Origin	AUTOSAR_ECUC

5.2.1.74. DcmDspRoutineStopIn

Containers included		
Container name	Multiplicity	Description
DcmDspRoutineStopInSignal	1..n	<p>Label: Stop Routine Input Signal Description</p> <p>Defines the configuration for a signal present in the <code>routineControlOptionRecord</code> input parameter present in the request for the <code>stopRoutine</code> (0x02) sub-function of the <code>RoutineControl</code> (0x31) diagnostic service for the referencing <code>DcmDspRoutine</code> Diagnostic Routine.</p>

5.2.1.75. DcmDspRoutineStopInSignal

Parameters included	
Parameter name	Multiplicity
DcmDspRoutineSignalLength	1..1

**Parameters included**

DcmDspRoutineSignalPos	1..1
DcmDspRoutineSignalType	1..1

Parameter Name	DcmDspRoutineSignalLength			
Label	Signal Length			
Description	<p>Defines the length of the signal, when the "signal length" is not variable, if <code>DcmDspRoutineSignalType</code> is not configured to <code>VARIABLE_LENGTH</code>, or the maximum length the variable "length signal" is allowed to reach, when <code>DcmDspRoutineSignalType</code> is configured to <code>VARIABLE_LENGTH</code></p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 1 .. 65535 bits <p>Note: The available length range is restricted from the AUTOSAR-defined range of 0 .. 65535 as signals of length 0 are not usable.</p>			
Multiplicity	1..1			
Type	INTEGER			
Default value	1			
Range	<table border="0"> <tr> <td><=65535</td> </tr> <tr> <td>>=1</td> </tr> </table>		<=65535	>=1
<=65535				
>=1				
Configuration class	VariantPreCompile:	VariantPreCompile		
Origin	AUTOSAR_ECUC			

Parameter Name	DcmDspRoutineSignalPos			
Label	Signal Position			
Description	<p>Defines the starting position of this signal in the <code>routineControlOptionRecord</code> input parameter of the diagnostic service request.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 65535 bits 			
Multiplicity	1..1			
Type	INTEGER			
Default value	0			
Range	<table border="0"> <tr> <td><=65535</td> </tr> <tr> <td>>=0</td> </tr> </table>		<=65535	>=0
<=65535				
>=0				



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DcmDspRoutineSignalType
Label	Signal Type
Description	<p>Defines the data type of data belonging to this signal.</p> <p>This configuration is used in order to facilitate Endianness conversion of the data if this is required (see configuration parameter <code>DcmDslProtocolEndianness-ConvEnabled</code>).</p> <p>This parameter also configures whether the length of this signal is variable or not. If the type of the signal is <code>VARIABLE_LENGTH</code>:</p> <ul style="list-style-type: none"> ▶ also configure <code>DcmDspRoutineFixedLength</code> to TRUE for the <code>DcmDspRoutine</code> referencing this <code>DcmDspRoutineInfo</code>. ▶ the length of the signal is supplied by the <code>Dcm</code> via the <code>currentDataLength</code> parameter of the <code>stopRoutine</code> operation interface. <p>Range:</p> <ul style="list-style-type: none"> ▶ BOOLEAN ▶ SINT16 ▶ SINT32 ▶ SINT8 ▶ UINT16 ▶ UINT32 ▶ UINT8 ▶ SINT16_N ▶ SINT32_N ▶ SINT8_N ▶ UINT16_N ▶ UINT32_N ▶ UINT8_N ▶ VARIABLE_LENGTH
Multiplicity	1..1
Type	ENUMERATION



Range	BOOLEAN
	SINT16
	SINT32
	SINT8
	UINT16
	UINT32
	UINT8
	SINT16_N
	SINT32_N
	SINT8_N
	UINT16_N
	UINT32_N
	UINT8_N
	VARIABLE_LENGTH
Configuration class	VariantPreCompile:
Origin	AUTOSAR_ECUC

5.2.1.76. DcmDspRoutineStopOut

Containers included		
Container name	Multiplicity	Description
DcmDspRoutineStopOutSignal	1..n	<p>Label: Stop Routine Output Signal Description</p> <p>Defines the configuration for a signal in the routineStatusRecord output parameter present in the response to the stopRoutine (0x02) sub-function of the RoutineControl (0x31) diagnostic service for the referencing DcmDspRoutine Diagnostic Routine.</p>

5.2.1.77. DcmDspRoutineStopOutSignal

Parameters included	
Parameter name	Multiplicity

**Parameters included**

DcmDspRoutineSignalLength	1..1
DcmDspRoutineSignalPos	1..1
DcmDspRoutineSignalType	1..1

Parameter Name	DcmDspRoutineSignalLength		
Label	Signal Length		
Description	<p>Defines the length of the signal, when the "signal length" is not variable, if DcmDspRoutineSignalType is not configured to VARIABLE_LENGTH, or the maximum length the variable "length signal" is allowed to reach, when DcmDspRoutineSignalType is configured to VARIABLE_LENGTH</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 1 .. 65535 bits <p>Note: The available length range is restricted from the AUTOSAR-defined range of 0 .. 65535 as signals of length 0 are not usable.</p>		
Multiplicity	1..1		
Type	INTEGER		
Default value	1		
Range	<table border="0"> <tr> <td><=65535</td> </tr> <tr> <td>>=1</td> </tr> </table>	<=65535	>=1
<=65535			
>=1			
Configuration class	<table border="0"> <tr> <td>VariantPreCompile:</td> <td> VariantPreCompile</td> </tr> </table>	VariantPreCompile:	VariantPreCompile
VariantPreCompile:	VariantPreCompile		
Origin	AUTOSAR_ECUC		

Parameter Name	DcmDspRoutineSignalPos
Label	Signal Position
Description	Defines the starting position of this signal in the routineStatusRecord output parameter of the diagnostic service response.
	<p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 65535 bits
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<=65535



	>=0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspRoutineSignalType
Label	Signal Type
Description	<p>Defines the data type of data belonging to this signal.</p> <p>This configuration is used in order to facilitate Endianness conversion of the data if this is required (see configuration parameter <code>DcmDslProtocolEndiannessConvEnabled</code>).</p> <p>This parameter also configures whether the length of this signal is variable or not. If the type of the signal is <code>VARIABLE_LENGTH</code>:</p> <ul style="list-style-type: none"> ▶ also configure <code>DcmDspRoutineFixedLength</code> to TRUE for the <code>DcmDspRoutine</code> referencing this <code>DcmDspRoutineInfo</code>. ▶ the length of the signal is supplied by the Application via the <code>currentDataLength</code> parameter of the <code>stopRoutine</code> operation interface. <p>Range:</p> <ul style="list-style-type: none"> ▶ BOOLEAN ▶ SINT16 ▶ SINT32 ▶ SINT8 ▶ UINT16 ▶ UINT32 ▶ UINT8 ▶ SINT16_N ▶ SINT32_N ▶ SINT8_N ▶ UINT16_N ▶ UINT32_N ▶ UINT8_N ▶ VARIABLE_LENGTH
Multiplicity	1..1



Type	ENUMERATION	
Range	BOOLEAN	
	SINT16	
	SINT32	
	SINT8	
	UINT16	
	UINT32	
	UINT8	
	SINT16_N	
	SINT32_N	
	SINT8_N	
	UINT16_N	
	UINT32_N	
	UINT8_N	
	VARIABLE_LENGTH	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.78. DcmDspStartRoutineIn

Containers included		
Container name	Multiplicity	Description
DcmDspStartRoutineInSignal	1..n	Defines the configuration for a signal in the <code>routineControlOptionRecord</code> input parameter present in the request for the <code>startRoutine</code> (0x01) sub-function of the <code>RoutineControl</code> (0x31) diagnostic service for the referencing <code>DcmDspRoutine</code> Diagnostic Routine.

5.2.1.79. DcmDspStartRoutineInSignal

Parameters included	
Parameter name	Multiplicity
DcmDspRoutineSignalLength	1..1

**Parameters included**

DcmDspRoutineSignalPos	1..1
DcmDspRoutineSignalType	1..1

Parameter Name	DcmDspRoutineSignalLength			
Label	Signal Length			
Description	<p>Defines the length of the signal, when the "signal length" is not variable, if <code>DcmDspRoutineSignalType</code> is not configured to <code>VARIABLE_LENGTH</code>, or the maximum length the variable "length signal" is allowed to reach, when <code>DcmDspRoutineSignalType</code> is configured to <code>VARIABLE_LENGTH</code></p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 1 .. 65535 bits <p>Note: The available length range is restricted from the AUTOSAR-defined range of 0 .. 65535 as signals of length 0 are not usable.</p>			
Multiplicity	1..1			
Type	INTEGER			
Default value	1			
Range	<table border="0"> <tr> <td><=65535</td> </tr> <tr> <td>>=1</td> </tr> </table>		<=65535	>=1
<=65535				
>=1				
Configuration class	VariantPreCompile:	VariantPreCompile		
Origin	AUTOSAR_ECUC			

Parameter Name	DcmDspRoutineSignalPos			
Label	Signal Position			
Description	<p>Defines the starting position of this signal in the <code>routineControlOptionRecord</code> input parameter of the diagnostic service request.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 65535 bits 			
Multiplicity	1..1			
Type	INTEGER			
Default value	0			
Range	<table border="0"> <tr> <td><=65535</td> </tr> <tr> <td>>=0</td> </tr> </table>		<=65535	>=0
<=65535				
>=0				



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DcmDspRoutineSignalType
Label	Signal Type
Description	<p>Defines the data type of data belonging to this signal.</p> <p>This configuration is used in order to facilitate Endianness conversion of the data if this is required (see configuration parameter <code>DcmDslProtocolEndianness-ConvEnabled</code>).</p> <p>This parameter also configures whether the length of this signal is variable or not. If the type of the signal is <code>VARIABLE_LENGTH</code>:</p> <ul style="list-style-type: none"> ▶ also configure <code>DcmDspRoutineFixedLength</code> to TRUE for the <code>DcmDspRoutine</code> referencing this <code>DcmDspRoutineInfo</code>. ▶ the length of the signal is supplied by the <code>Dcm</code> via the <code>currentDataLength</code> parameter of the <code>startRoutine</code> operation interface. <p>Range:</p> <ul style="list-style-type: none"> ▶ BOOLEAN ▶ SINT16 ▶ SINT32 ▶ SINT8 ▶ UINT16 ▶ UINT32 ▶ UINT8 ▶ SINT16_N ▶ SINT32_N ▶ SINT8_N ▶ UINT16_N ▶ UINT32_N ▶ UINT8_N ▶ VARIABLE_LENGTH
Multiplicity	1..1
Type	ENUMERATION



Range	BOOLEAN
	SINT16
	SINT32
	SINT8
	UINT16
	UINT32
	UINT8
	SINT16_N
	SINT32_N
	SINT8_N
	UINT16_N
	UINT32_N
	UINT8_N
	VARIABLE_LENGTH
Configuration class	VariantPreCompile:
Origin	AUTOSAR_ECUC

5.2.1.80. DcmDspStartRoutineOut

Containers included		
Container name	Multiplicity	Description
DcmDspStartRoutineOutSignal	1..n	<p>Label: Start Routine Output Signal Description</p> <p>Defines the configuration of a signal used in the <code>routineStatusRecord</code> output parameter present in the response to the <code>startRoutine</code> (0x01) sub-function of the <code>RoutineControl</code> (0x31) diagnostic service for the referencing <code>DcmDspRoutine</code> Diagnostic Routine.</p>

5.2.1.81. DcmDspStartRoutineOutSignal

Parameters included	
Parameter name	Multiplicity
DcmDspRoutineSignalLength	1..1

**Parameters included**

DcmDspRoutineSignalPos	1..1
DcmDspRoutineSignalType	1..1

Parameter Name	DcmDspRoutineSignalLength			
Label	Signal Length			
Description	<p>Defines the length of the signal, when the "signal length" is not variable, if <code>DcmDspRoutineSignalType</code> is not configured to <code>VARIABLE_LENGTH</code>, or the maximum length the variable "length signal" is allowed to reach, when <code>DcmDspRoutineSignalType</code> is configured to <code>VARIABLE_LENGTH</code></p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 1 .. 65535 bits <p>Note: The available length range is restricted from the AUTOSAR-defined range of 0 .. 65535 as signals of length 0 are not usable.</p>			
Multiplicity	1..1			
Type	INTEGER			
Default value	1			
Range	<table border="0"> <tr> <td><=65535</td> </tr> <tr> <td>>=1</td> </tr> </table>		<=65535	>=1
<=65535				
>=1				
Configuration class	VariantPreCompile:	VariantPreCompile		
Origin	AUTOSAR_ECUC			

Parameter Name	DcmDspRoutineSignalPos			
Label	Signal Position			
Description	<p>Defines the starting position of this signal in the <code>routineStatusRecord</code> output parameter of the diagnostic service response.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 65535 bits 			
Multiplicity	1..1			
Type	INTEGER			
Default value	0			
Range	<table border="0"> <tr> <td><=65535</td> </tr> <tr> <td>>=0</td> </tr> </table>		<=65535	>=0
<=65535				
>=0				



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DcmDspRoutineSignalType
Label	Signal Type
Description	<p>Defines the data type of data belonging to this signal.</p> <p>This configuration is used in order to facilitate Endianness conversion of the data if this is required (see configuration parameter <code>DcmDslProtocolEndianness-ConvEnabled</code>).</p> <p>This parameter also configures whether the length of this signal is variable or not. If the type of the signal is <code>VARIABLE_LENGTH</code>:</p> <ul style="list-style-type: none"> ▶ also configure <code>DcmDspRoutineFixedLength</code> to TRUE for the <code>DcmDspRoutine</code> referencing this <code>DcmDspRoutineInfo</code>. ▶ the length of the signal is supplied by the Application via the <code>currentDataLength</code> parameter of the <code>startRoutine</code> operation interface. <p>Range:</p> <ul style="list-style-type: none"> ▶ BOOLEAN ▶ SINT16 ▶ SINT32 ▶ SINT8 ▶ UINT16 ▶ UINT32 ▶ UINT8 ▶ SINT16_N ▶ SINT32_N ▶ SINT8_N ▶ UINT16_N ▶ UINT32_N ▶ UINT8_N ▶ VARIABLE_LENGTH
Multiplicity	1..1
Type	ENUMERATION



Range	BOOLEAN
	SINT16
	SINT32
	SINT8
	UINT16
	UINT32
	UINT8
	SINT16_N
	SINT32_N
	SINT8_N
	UINT16_N
	UINT32_N
	UINT8_N
	VARIABLE_LENGTH
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.2.1.82. DcmDspSecurity

Containers included		
Container name	Multiplicity	Description
DcmDspSecurityRow	0..31	<p>Defines the Security Levels available to the Dcm. Each Security Level is configured per row.</p> <p>The Dcm generates the ClientServerInterface SecurityAccess_&#60;LEVEL&#62; and the Require Port for this interface, also named SecurityAccess_&#60;LEVEL&#62; where &#60;LEVEL&#62; is the name of the container DcmDspSecurityRow. This interface will be used (if parameter DcmDspSecurityUsePort is USE_SECURITY_ASYNC_CLIENT_SERVER or USE_SECURITY_SYNC_CLIENT_SERVER) in order to realize the GetSeed and CompareKey operations.</p>
Parameters included		
Parameter name		Multiplicity



Parameters included

DcmDspSecurityLegacyAttemptCountersHandling	1..1
DcmDspSecurityMaxAttemptCounterReadoutTime	0..1
DcmDspSecurityMaxAttemptCounterWriteTime	0..1
DcmDspSecurityUnifiedDelayTimer	1..1
DcmDspSecurityResetAttemptCounterOnTimeout	1..1
DcmDspSecurityAttemptCounterReadoutFailInfiniteDelay	1..1
DcmDspSecurityIncrementAttemptCounterPastLimit	1..1

Parameter Name	DcmDspSecurityLegacyAttemptCountersHandling	
Label	Enable legacy AttemptCounter handling	
Description	<p>This flag enables legacy handling of AttemptCounters.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ FALSE: No saving of AttemptCounters to non-volatile memory is performed. ▶ TRUE: AttemptCounters are saved to non-volatile memory using the legacy Rte_DcmSecSetNumAtt interface. AttemptCounters are retrieved from non-volatile memory with every SendKey operation using the legacy Rte_DcmSecGetNumAtt interface. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmDspSecurityMaxAttemptCounterReadoutTime	
Label	Max. time to read out all attempt counters after PowerON	
Description	<p>Configures the delay, in seconds, from startup (measured from the first call of the Dcm_MainFunction), allowed for all AttemptCounter values to be obtained from the Application. Shall be a multiple of DcmTaskTime min: A value equal to the DcmTaskTime</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 65535 seconds 	
Multiplicity	0..1	
Type	FLOAT	



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU V1.0.0	

Parameter Name	DcmDspSecurityMaxAttemptCounterWriteTime	
Label	Max. time to write an attempt counter	
Description	<p>Configures the maximal time, in seconds, allowed for one attempt counter value to be written by the API <code>xxx_SetSecurityAttemptCounter</code>. This parameter only applies when the attempt counter is reset as a result of a delay timer elapsing when the parameter <code>DcmDspSecurityResetAttemptCounterOnTimeout</code> is set to TRUE. The timing-out behaviour in the case of writing the counter upon a failed <code>CompareKey</code> request is handled with the usual P2/P2* mechanism. Shall be a multiple of <code>DcmTaskTime</code> min: A value equal to <code>DcmTaskTime</code> Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 65535 seconds 	
Multiplicity	0..1	
Type	FLOAT	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmDspSecurityUnifiedDelayTimer	
Label	Enable unified delay timer	
Description	<p>This flag enables a unified delay timer.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ FALSE: Starting one SecurityLevel's DelayTimer only affects that SecurityLevel. ▶ TRUE: Starting one SecurityLevel's DelayTimer starts the DelayTimers of ALL SecurityLevels with the DelayTimer configured for that level. In case more than one DelayTimer value applies (as in the case of more than one DelayTimer being applied at startup as a result of retrieving AttemptCounter values), the highest value shall apply. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile



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

Parameter Name	DcmDspSecurityResetAttemptCounterOnTimeout	
Label	Reset attempt counter on security access delay timeout	
Description	<p>This flag enables resetting the security access attempt counter when the delay timer expires.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ FALSE: The attempt counter is not reset. ISO compatible! ▶ TRUE: If <code>DcmDspSecurityUnifiedDelayTimer</code> is disabled: When one SecurityLevel's DelayTimer expires, reset its AttemptCounter to 0 ▶ TRUE: If <code>DcmDspSecurityUnifiedDelayTimer</code> is enabled: When the unified DelayTimer expires, reset ALL AttemptCounters to 0 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmDspSecurityAttemptCounterReadoutFailInfiniteDelay	
Label	Enable infinite delay when reading out an attempt counter fails	
Description	<p>This flag forces infinite delay if the read out of any security access attempt counter fails.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ FALSE: If any of the Xxx_GetSecurityAttemptCounter operations for getting the AttemptCounters fails, the DelayTimer is started, as a consequence of this failure. ▶ TRUE: If any of the Xxx_GetSecurityAttemptCounter operations for getting the AttemptCounters fails, the DelayTimer is started, as a consequence of this failure, with infinite value. <p>Note: In case both this parameter and <code>DcmDspSecurityUnifiedDelayTimer</code> are configured to TRUE, this will cause all SecurityLevels to be permanently locked.</p>	
Multiplicity	1..1	
Type	BOOLEAN	



Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspSecurityIncrementAttemptCounterPastLimit
Label	Increment the security access attempt counter past the configured limit
Description	<p>This flag indicates if the security attempt counter is incremented any time a security access fails or just up to the number of times configured in DcmDspSecurityNumAttDelay.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ FALSE: Do not stop incrementing the attempt counter at the limit configured in DcmDspSecurityNumAttDelay (ISO 14229-1:2013 conforming behaviour) ▶ TRUE: Stop incrementing the attempt counter at the limit configured in DcmDspSecurityNumAttDelay (non-ISO 14229-1:2013 conforming behaviour)
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.2.1.83. DcmDspSecurityRow

Parameters included	
Parameter name	Multiplicity
DcmDspSecurityAttemptCounterEnabled	0..1
DcmDspSecurityDelayTimeOnBootOverride	1..1
DcmDspSecurityDelayTimeInfinite	1..1
DcmDspSecurityADRSIZE	0..1
DcmDspSecurityGetAttemptCounterFnc	0..1
DcmDspSecuritySetAttemptCounterFnc	0..1
DcmDspSecurityCompareKeyFnc	0..1

**Parameters included**

DcmDspSecurityDelayTime	0..1
DcmDspSecurityDelayTimeOnBoot	0..1
DcmDspSecurityGetSeedFnc	0..1
DcmDspSecurityKeySize	1..1
DcmDspSecurityLevel	1..1
DcmDspSecurityNumAttDelay	0..1
DcmDspSecuritySeedSize	1..1
DcmDspSecurityUsePort	1..1

Parameter Name**DcmDspSecurityAttemptCounterEnabled****Label**

Enable external handling of the security attempt counter

Description

Allows to enable the external handling of the security attempt counter (e.g. to survive a reset of the ECU).

Range:

- ▶ TRUE: The security attempt counter is handled externally
- ▶ FALSE: The security attempt counter is not handled externally

Multiplicity

0..1

Type

BOOLEAN

Default value

false

Configuration class**VariantPreCompile:**

VariantPreCompile

Origin

Elektrobit Automotive GmbH

Parameter Name**DcmDspSecurityDelayTimeOnBootOverride****Label**

Force security access delay on boot

Description

This flag allows to force delay time on boot regardless of the value of the security access attempt counter.

Range:

- ▶ TRUE: Upon startup, the DelayTimer for this security level is started with the value of DcmDspSecurityDelayTimeOnBoot regardless of the number of attempts retrieved from non-volatile memory for this security level.
- ▶ FALSE: Upon startup, the DelayTimer for this security level is started with the maximum value among DcmDspSecurityDelayTimeOnBoot or DcmD-



	<p>spSecurityDelayTime if the number of attempts retrieved from non-volatile memory for this security level has been reached.</p> <p>Note: In case both this parameter and <code>DcmDspSecurityUnifiedDelayTimer</code> are configured to TRUE, this will cause all SecurityLevels to start their Delay-Timers at startup with the highest value of all configured <code>DcmDspSecurityDelayTimeOnBoot</code> delays, for SecurityLevels for which the retrieval of the AttemptCounter from non-volatile memory succeeds or is unneeded. In case a higher unified DelayTimer value applies as a result of an incorrect retrieval, or as a result of the retrieved AttemptCounter of another SecurityLevel surpassing the configured limit, the higher value shall apply.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspSecurityDelayTimeInfinite
Label	Force infinite security access delay time
Description	<p>This flag forces infinite delay time after failed security access attempt.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: <code>DcmDspSecurityDelayTime</code> is considered infinite. When <code>DcmDspSecurityNumAttDelay</code> failed attempts are reached, this security level becomes blocked/inaccessible. ▶ FALSE: The value configured under <code>DcmDspSecurityDelayTime</code> applies. <p>Note: If <code>DcmDspSecurityUnifiedDelayTimer</code> is configured to TRUE, if such an infinite DelayTimer is started, this will cause all SecurityLevels to remain permanently locked.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspSecurityADRSIZE
Label	Size of securityAccessDataRecord



Description	Defines the size of the <code>securityAccessDataRecord</code> parameter expected in <code>requestSeed</code> (<code>DcmDspSecurityLevel * 2 - 1</code>) sub-function of the <code>SecurityAccess</code> (0x27) service request for this particular security level.
	<p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 4294967293 Bytes <p>Note 1: The available size range is restricted to the AUTOSAR-specified maximum minus 2 Bytes, as otherwise the full request, which contains two more Bytes besides this parameter, would not be able to fit into any <code>DcmDslBuffer</code>, which has a maximum size of 4294967295 Bytes.</p> <p>Note 2: If this parameter is not enabled, function parameters of the <code>requestSeed</code> interface representing the <code>securityAccessDataRecord</code> parameter are not present.</p>
Multiplicity	0..1
Type	INTEGER
Default value	0
Range	<p><=4294967293</p> <hr/> <p>>=1</p>
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DcmDspSecurityGetAttemptCounterFnc	
Label	Get attempt counter function	
Description	<p>Defines the name of the callout used to retrieve the security access attempt counter for this particular Security Level.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ External attempt counter handling being enabled by setting <code>DcmDspSecurityAttemptCounterEnabled</code> to TRUE ▶ SecurityAccess Operation interface type (<code>DcmDspSecurityUsePort</code>): The configuration of this parameter is only relevant if parameter <code>DcmDspSecurityUsePort</code> is <code>USE_SECURITY_ASYNC_FNC</code> or <code>USE_SECURITY_SYNCH_FNC</code>. 	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	PreCompile: VariantPreCompile	



Origin	Elektrobit Automotive GmbH	
Parameter Name	DcmDspSecuritySetAttemptCounterFnc	
Label	Set attempt counter function	
Description	<p>Defines the name of the callout used to set the security access attempt counter for this particular security level.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ External attempt counter handling being enabled by setting <code>DcmDspSecurityAttemptCounterEnabled</code> to TRUE ▶ SecurityAccess Operation interface type (<code>DcmDspSecurityUsePort</code>): The configuration of this parameter is only relevant if parameter <code>DcmDspSecurityUsePort</code> is <code>USE_SECURITY_ASYNC_FNC</code> or <code>USE_SECURITY_SYNCH_FNC</code>. 	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	
Parameter Name	DcmDspSecurityCompareKeyFnc	
Label	Compare Key function	
Description	<p>Defines the name of the Callout used to supply the Security Key generated from the provided Security Seed to the Server, in the context of the <code>sendKey</code> (<code>DcmDspSecurityLevel * 2</code>) sub-function of the SecurityAccess (0x27) service request for this particular Security Level.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ SecurityAccess Operation interface type (<code>DcmDspSecurityUsePort</code>): The configuration of this parameter is only relevant if parameter <code>DcmDspSecurityUsePort</code> is <code>USE_SECURITY_ASYNC_FNC</code> or <code>USE_SECURITY_SYNCH_FNC</code>. 	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	
Parameter Name	DcmDspSecurityDelayTime	
Label	Delay time for SecurityAccess	



Description	Defines the delay time after a failed SecurityAccess (0x27) request sequence, before another attempt to execute the service is allowed.
	<p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 65535 seconds <p>Note: This timer is started after a <code>DcmDspSecurityNumAttDelay</code> number of failed request sequences.</p>
Multiplicity	0..1
Type	FLOAT
Default value	0.0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspSecurityDelayTimeOnBoot	
Label	Delay time after PowerON	
Description	Defines the delay time after power-on until a SecurityAccess (0x27) service request will be allowed to be executed.	
	<p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 65535 seconds <p>Note: This timer is started after a <code>DcmDspSecurityNumAttDelay</code> number of failed request sequences.</p>	
Multiplicity	0..1	
Type	FLOAT	
Default value	0.0	
Range	<p><=65535</p> <hr/> <p>>=0</p>	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC V1.0.0	

Parameter Name	DcmDspSecurityGetSeedFnc	
Label	Security Get Seed function	
Description	Defines the name of the Callout used to request the Security Seed from the Server, and, if configured (<code>DcmDspSecurityADRSIZE</code> > 0), also to supply the <code>securityAccessDataRecord[]</code> from the request to the Server in the	



	<p>context of the <code>sendKey (DcmDspSecurityLevel * 2)</code> sub-function of the <code>SecurityAccess (0x27)</code> service request for this particular security level.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ SecurityAccess Operation interface type (<code>DcmDspSecurityUsePort</code>): The configuration of this parameter is only relevant if parameter <code>DcmDspSecurityUsePort</code> is <code>USE_SECURITY_ASYNC_FNC</code> or <code>USE_SECURITY_SYNCH_FNC</code>.
Multiplicity	0..1
Type	FUNCTION-NAME
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspSecurityKeySize	
Label	Size of securityKey	
Description	<p>Defines the size of the <code>securityKey</code> parameter expected in <code>sendKey (DcmDspSecurityLevel * 2)</code> sub-function of the <code>SecurityAccess (0x27)</code> service request for this particular security level.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 4294967293 Bytes <p>Note: the available size range is restricted to the AUTOSAR-specified maximum minus 2 Bytes, as otherwise the full request, which contains two more Bytes besides this parameter, would not be able to fit into any <code>DcmDslBuffer</code>, which has a maximum size of 4294967295 Bytes</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Range	<=4294967293	
	>=1	
Configuration class	VariantPreCompile: VariantPreCompile	
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspSecurityLevel	
Label	Security Level	
Description	Defines the numerical value of the Security Level.	



	<p>For each security level, the subservice handling corresponding to the two operations <code>sendKey</code> and <code>requestSeed</code>, is generated as follows:</p> <ul style="list-style-type: none"> ▶ The corresponding <code>requestSeed</code> sub-function of the <code>SecurityAccess</code> (0x27) service shall have the ID <code>DcmDspSecurityLevel * 2 -1</code> ▶ The corresponding <code>sendKey</code> sub-function of the <code>SecurityAccess</code> (0x27) service shall have the ID <code>DcmDspSecurityLevel * 2</code> <p>. Further configuration of sub-functions under <code>DcmDsdSubService</code> is not necessary.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 1 .. 63 <p>Note : The locked state cannot be configured explicitly.</p>
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DcmDspSecurityNumAttDelay	
Label	Maximum number of failed <code>SecurityAccess</code> attempts	
Description	Defines the number of failed attempts of <code>SecurityAccess</code> sequences after which the delay timer <code>DcmDspSecurityDelayTime</code> is started. Before this timer elapses, no other attempt to execute the service is allowed.	
Multiplicity	0..1	
Type	INTEGER	
Default value	0	
Range	<=255 >=1	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DcmDspSecuritySeedSize	
Label	Size of the <code>securitySeed</code>	
Description	Defines the size of the <code>securitySeed</code> parameter expected in the response to the <code>getSeed (DcmDspSecurityLevel * 2 -1)</code> sub-function of the <code>SecurityAccess</code> (0x27) service request for this particular security level.	



	<p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 4294967293 Bytes <p>Note: the available size range is restricted to the AUTOSAR-specified maximum minus 2 Bytes, as otherwise the full request, which contains two more Bytes besides this parameter, would not be able to fit into any <code>DcmDslBuffer</code>, which has a maximum size of 4294967295 Bytes</p>
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<p><=4294967293</p> <hr/> <p>>=1</p>
Configuration class	<p>VariantPreCompile: VariantPreCompile</p>
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspSecurityUsePort
Label	Security Access Interface
Description	<p>Defines the type of the interface that shall be used to implement the <code>sendKey</code> and <code>requestSeed</code> operations defining the sub-functions of service Security-Access (0x27)</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ USE_SECURITY_SYNCH_FNC: The two operations shall be implemented using a synchronous API (the operation shall be completed in one single call to the API). The following callout names shall be configured: <ul style="list-style-type: none"> ▶ DcmDspSecurityGetSeedFnc ▶ DcmDspSecurityCompareKeyFnc ▶ USE_SECURITY_ASYNC_FNC: The two operations shall be implemented using an asynchronous API (the operation shall be completed in a series of calls to the API, the first call being executed with an <code>OpStatus</code> set to <code>DCM_INITIAL</code>, and as long as the interface returns <code>E_PENDING</code>, it shall be called with an <code>OpStatus</code> set to <code>DCM_PENDING</code> for all subsequent executions). The following callout names shall be configured: <ul style="list-style-type: none"> ▶ DcmDspSecurityGetSeedFnc ▶ DcmDspSecurityCompareKeyFnc ▶ USE_SECURITY_SYNCH_CLIENT_SERVER: The two operations shall be implemented using by using a synchronous AUTOSAR ClientServerIn-



	<p>terface with synchronous behaviour (the operation shall be completed in one single call to the operation of the generated interface). The <code>Dcm</code> generates the <code>SecurityAccess_&#60;LEVEL&#62;</code> <code>ClientServerInterface</code> where <code>&#60;LEVEL&#62;</code> is the name of this <code>DcmDspSecurityRow</code> container. This <code>ClientServerInterface</code> contains the <code>getSeed</code> and <code>compareKey</code> operations. A Require Port is generated for this interface and shall be connected to a compatible Provide Port of an Application Software Component. The name of this Require Port is the same as the name of the <code>ClientServerInterface</code>.</p> <ul style="list-style-type: none"> ▶ <code>USE_SECURITY_ASYNC_CLIENT_SERVER</code>: The two operations shall be implemented using by using a synchronous AUTOSAR <code>ClientServerInterface</code> with asynchronous behaviour (the operation shall be completed in a series of calls to the API, the first call being executed with an <code>OpStatus</code> set to <code>DCM_INITIAL</code>, and as long as the interface returns <code>E_PENDING</code>, it shall be called with an <code>OpStatus</code> set to <code>DCM_PENDING</code> for all subsequent executions). The <code>Dcm</code> generates the <code>SecurityAccess_&#60;LEVEL&#62;</code> <code>ClientServerInterface</code> where <code>&#60;LEVEL&#62;</code> is the name of this <code>DcmDspSecurityRow</code> container. This <code>ClientServerInterface</code> contains the <code>getSeed</code> and <code>compareKey</code> operations. A Require Port is generated for this interface and shall be connected to a compatible Provide Port of an Application Software Component. The name of this Require Port is the same as the name of the <code>ClientServerInterface</code>. 	
Multiplicity	1..1	
Type	ENUMERATION	
Range	<p><code>USE_SECURITY_ASYNC_CLIENT_SERVER</code></p> <p><code>USE_SECURITY_ASYNC_FNC</code></p> <p><code>USE_SECURITY_SYNCH_CLIENT_SERVER</code></p> <p><code>USE_SECURITY_SYNCH_FNC</code></p>	
Configuration class	<p>VariantPreCompile:</p> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">VariantPreCompile</td> </tr> </table>	VariantPreCompile
VariantPreCompile		
Origin	Elektrobit Automotive GmbH	

5.2.1.84. DcmDspSession

Containers included		
Container name	Multiplicity	Description
DcmDspSessionRow	0..31	Defines the Diagnostic Sessions available to the <code>Dcm</code> . Each Diagnostic Session is configured per row.



Containers included

		<p>Note 1: If a functionality shall be allowed in for all Diagnostic Sessions, Diagnostic Session <code>ALL_SESSION_LEVEL</code> shall be added with value 0xFF.</p> <p>Note 2: If a Diagnostic Session is configured, the related Sub-function ID of the <code>DiagnosticSessionControl</code> (0x10) service also has to be configured in order for transitions into this Diagnostic Session to be possible.</p>
--	--	--

5.2.1.85. DcmDspSessionRow

Parameters included

Parameter name	Multiplicity
DcmDspSessionForBoot	1..1
DcmDspSessionLevel	1..1
DcmDspSessionP2ServerMax	1..1
DcmDspSessionP2StarServerMax	1..1
DcmDspDiagSesRespMaxNumRespPend	0..1

Parameter Name	DcmDspSessionForBoot
Label	Jump to BootLoader
Description	<p>Configures whether a transition to this Diagnostic Session is used in order to perform a jump to Bootloader software (OEM Bootloader or System Supplier Bootloader).</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ <code>DCM_NO_BOOT</code>: A transition to this Diagnostic Session does not trigger a jump to any Bootloader. ▶ <code>DCM_OEM_BOOT</code>: A transition to this Diagnostic Session triggers a jump to the OEM Bootloader. ▶ <code>DCM_SYS_BOOT</code>: A transition to this Diagnostic Session triggers a jump to the System Supplier Bootloader.
Multiplicity	1..1
Type	ENUMERATION
Range	<code>DCM_NO_BOOT</code>



	DCM_OEM_BOOT
	DCM_SYS_BOOT
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC V1.0.0

Parameter Name	DcmDspSessionLevel
Label	Session Level
Description	<p>Defines the numeric identifier of the Diagnostic Session.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0x01 ... 0x04: Diagnostic Sessions mandatory according to ISO14229-1 ▶ 0x05 ... 0x3F: ISOSAEReserved according to ISO14229-1 ▶ 0x40 ... 0x5F: Vehicle Manufacturer Specific Diagnostic Sessions ▶ 0x60 ... 0x7E: Vehicle Supplier Specific Diagnostic Sessions ▶ 0xFF: Defines that the functionality referencing this Diagnostic Session shall be allowed to run in any Diagnostic Session <p>Note 1: Values of the Diagnostic Sessions shall be configured according to ISO_14229-1.</p> <p>Note 2: Value 0xFF is reserved for the special ALL_SESSION_LEVEL Diagnostic Session, used in order to indicate that the functionality referencing it is to be allowed to run in any Diagnostic Session.</p>
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspSessionP2ServerMax
Label	P2ServerMax
Description	<p>Defines the value for the P2ServerMax in seconds as defined by ISO 15765-3. It denotes the maximum time available for the Dcm to start a response message transmission after the reception of a request message.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 1 seconds



Multiplicity	1..1
Type	FLOAT
Default value	0.0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	DcmDspSessionP2StarServerMax
Label	P2*ServerMax
Description	<p>Defines the value for the <code>P2ServerMax</code> in seconds as defined by ISO 15765-3. It denotes the maximum time available for the <code>Dcm</code> to start a response message transmission during enhanced response timing.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 100 seconds
Multiplicity	1..1
Type	FLOAT
Default value	0.0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	DcmDspDiagSesRespMaxNumRespPend
Label	Maximum number of ResponsePending messages
Description	<p>Configures the maximum number of consecutive <code>RequestCorrectlyReceived-ResponsePending</code> Negative Response Codes (NRC) allowed for one diagnostic request while in this session. When enabled and configured, the value of this parameter overrides the value of the <code>DcmDslDiagRespMaxNumRespPend</code> parameter while the ECU is in this diagnostic session.</p> <p>The <code>Dcm</code> shall send the <code>GeneralReject</code> (0x10) NRC when the number of <code>RequestCorrectlyReceived-ResponsePending</code> NRCs sent in the context of the current request processing has reached the limit defined by this parameter and if this parameter (<code>DcmDspDiagSesRespMaxNumRespPend</code>) is configured to less than the value interpreted as an infinite number of <code>RequestCorrectlyReceived-ResponsePending</code> (NRCs). The value interpreted as infinity is given by the configuration parameter <code>DcmDslDiagRespMaxNumRespPendInfinityLimit</code>:</p> <ul style="list-style-type: none"> ▶ 255 is interpreted as infinity if the configuration parameter <code>DcmDslDiagRespMaxNumRespPendInfinityLimit</code> is set to "255".



	<ul style="list-style-type: none"> ▶ 65535 is interpreted as infinity if the configuration parameter <code>DcmDslDiagRespMaxNumRespPendInfinityLimit</code> is set to "65535". <p>If this parameter (<code>DcmDspDiagSesRespMaxNumRespPend</code>) is configured to infinity, there is no limit imposed on the maximum number of consecutive RequestCorrectlyReceived-ResponsePending NRCs allowed.</p>
Multiplicity	0..1
Type	INTEGER
Default value	0
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.2.1.86. DcmDspTestResultByObdmid

Containers included		
Container name	Multiplicity	Description
DcmDspTestResultObdmidTid	0..255	<p>This parameter is enabled only if service 0x06 is supported.</p> <p>Defines the configuration parameters for an OBDMIDs (On-Board Diagnostic Monitor IDs) and the assigned Tids (Test IDs) to be used in the response to the \$06 - Request On-Board Monitoring Test-results for Specific Monitored Systems OBD service</p>
DcmDspTestResultTid	0..255	<p>This parameter is enabled only if service 0x06 is supported.</p> <p>Defines the configuration parameters for obtaining the data of a single TID.</p> <p>This operation is realized by using a synchronous AUTOSAR ClientServerInterface with synchronous behaviour (the operation shall be completed in one single call to the operation of the generated interface). The <code>Dcm</code> shall generate the <code>DTRServices ClientServerInterface</code>, is the name of this <code>DcmDspRequestControl</code> container. This ClientServerInterface contains the necessary <code>Get-DTRValue</code> operation. A Require Port, named <code>DtrServices_&#60;OBDMID&#62;_&#60;TID&#62;</code>, is generated for this interface for each TID (<code>DcmDspTestRe-</code></p>



Containers included

		sultTid) belonging to each OBDMID (DcmDspTestResultObdmidTid), where <OBDMID> is the name of the DcmDspTestResultObdmidTid container and <TID> is the name of the DcmDspTestResult-Tid container. This Require Port shall be connected to a compatible Provide Port of an Application Software Component.
--	--	---

Parameters included

Parameter name	Multiplicity
DcmOBDMIDSupportType	1..1
DcmWriteMidBeforeTid	1..1

Parameter Name	DcmOBDMIDSupportType	
Label	OBDMID Support	
Description	<p>This parameter is enabled only if service 0x06 is supported.</p> <p>DcmOBDMIDSupportType can select from where the OBDMID are read.</p> <p>DCM_OBDMID_SUPPORT_SWC will read the values from a SWC according to ASR 4.0.3</p> <p>DCM_OBDMID_SUPPORT_DEM will read the values from the Dem according to ASR 4.2.1</p>	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	DCM_OBDMID_SUPPORT_SWC	
Range	DCM_OBDMID_SUPPORT_SWC DCM_OBDMID_SUPPORT_DEM	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmWriteMidBeforeTid	
Label	Write MID before TID	
Description	This configuration switch defines whether the OBDMID shall be added in the response in front of every configured TID inside one OBDMID, in case it's requested by UDS service 0x22.	



	<p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: The OBDMID will be added before every TID inside the response message. ▶ FALSE: The OBDMID will only be added once as part of the DID requested via service 0x22
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.2.1.87. DcmDspTestResultObdmidTid

Containers included		
Container name	Multiplicity	Description
DcmDspTestResultObdmidTids	0..255	<p>Label: OBDMID TID Configuration</p> <p>Defines the configuration of the TIDs belonging to the OBDMID</p>

Parameters included	
Parameter name	Multiplicity
DcmDspTestResultObdmid	1..1
DcmDspTestResultObdmidUsed	1..1

Parameter Name	DcmDspTestResultObdmid
Label	OBDMID
Description	Defines the identifier of this OBDMID (On-Board Diagnostic Monitor ID), as expected in the On-Board Diagnostic Monitor ID parameter of the \$06 - Request On-Board Monitoring Test-results for Specific Monitored Systems OBD service request.
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC



Parameter Name	DcmDspTestResultObdmidUsed	
Label	Enable MID usage	
Description	By checking this you will enable this OBDMID to be requested by (On-Board Diagnostic Monitor IDs) and the assigned Tids (Test IDs) to be used in the response to the \$06 - Request On-Board Monitoring Test-results for Specific Monitored Systems OBD service	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

5.2.1.88. DcmDspTestResultObdmidTids

Containers included		
Container name	Multiplicity	Description
DcmDspTestResultObdmidTidRefs	1..n	<p>Label: Configure support of TID reference</p> <p>In this container we can find all references to TIDs that can be assigned to the same UaSID reported to an OBDMID(On-Board Diagnostic Monitor IDs) used in the response to the \$06 - Request On-Board Monitoring Test-results for Specific Monitored Systems OBD service</p>

Parameters included	
Parameter name	Multiplicity
DcmDspTestResultObdmidTidUaSid	1..1

Parameter Name	DcmDspTestResultObdmidTidUaSid
Label	Unit And Scaling ID
Description	Defines the Unit And Scaling ID for the TIDs referenced in DcmDspTestResultObdmidTidRef, to be used in the \$06 - Request On-Board Monitoring Test-results for Specific Monitored Systems OBD service response.
Multiplicity	1..1
Type	INTEGER
Default value	0



Range	<=255 >=0	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

5.2.1.89. DcmDspTestResultObdmidTidRefs

Parameters included		
Parameter name	Multiplicity	
DcmDspTestResultObdmidTidRef	1..1	
DcmDspTestResultTidUsed	1..1	
Parameter Name		
DcmDspTestResultObdmidTidRef		
Label	Reference to TIDs of OBDMID	
Description	Reference to <code>DcmDspTestResultTid</code> defining TIDs belonging to this OBDMID.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	
Parameter Name		
DcmDspTestResultTidUsed		
Label	Enable TIDRef usage	
Description	This parameter lets you to choose what TID is processed inside this OBDMID container.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

5.2.1.90. DcmDspTestResultTid

Parameters included		
Parameter name	Multiplicity	



Parameters included

DcmDspTestResultTestId	1..1
--	------

Parameter Name	DcmDspTestResultTestId
Label	TID
Description	Std./Manuf. TID of this TID to be used in the \$06 - Request On-Board Monitoring Test-results for Specific Monitored Systems OBD service response.
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.2.1.91. DcmDspVehInfo

Containers included

Container name	Multiplicity	Description
DcmDspVehInfoData	1..n	Label: Vehicle Information Data Defines the configuration parameters of a Vehicle Information Data element.

Parameters included

Parameter name	Multiplicity
DcmDspVehInfoInfoType	1..1
DcmDspVehInfoUsed	1..1
DcmDspVehInfoNODIProvResp	0..1

Parameter Name	DcmDspVehInfoInfoType
Label	InfoType
Description	Defines the InfoType requested in the \$09 - Request Vehicle Information service request represented by this DcmDspVehInfoInfoType
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC



Parameter Name	DcmDspVehInfoUsed	
Label	Enable Vehicle Info	
Description	Allow to activate or deactivate the usage of a service: true = service available, false = service not available Range: <ul style="list-style-type: none"> ▶ TRUE: This Vehicle Info is available. ▶ FALSE: This Vehicle Info is not available. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmDspVehInfoNODIProvResp	
Label	DcmDspVehInfoNODIProvResp	
Description	Indicate the Dcm, which side is responsible to fill the number of data items (NODI), Dcm or the provider of the InfoType data. In case the responsibility is on provider side, only one DcmDspVehInfoData container is allowed. Range <ul style="list-style-type: none"> ▶ TRUE: Provider is responsible for providing the number of data items parameter. ▶ FALSE: Dcm is responsible for providing the number of data items parameter 	
Multiplicity	0..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.2.1.92. DcmDspVehInfoData

Parameters included	
Parameter name	Multiplicity



Parameters included

DcmDspVehInfoDataOrder	1..1
DcmDspVehInfoDataReadFnc	0..1
DcmDspVehInfoDataSize	1..1
DcmDspVehInfoDataUsePort	1..1

Parameter Name	DcmDspVehInfoDataOrder	
Label	Data Index	
Description	<p>Defines the order which the data item will take in the <code>InfoType</code> for the response: the first data item having the order number 0, the next 1 and so on.</p> <p>Note: The configuration of data order needs to be unique per <code>DcmDspVehInfo</code>.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 0 .. 255 	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspVehInfoDataReadFnc	
Label	Function for reading the <code>InfoType</code>	
Description	<p>Defines the name of the Callout used for reading the associated <code>InfoType</code> data item.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Vehicle Info data retrieval interface type (<code>DcmDspVehInfoDataUsePort</code>): The configuration of this parameter is only relevant if <code>DcmDspVehInfoDataUsePort</code> is set to FALSE. 	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDspVehInfoDataSize	
-----------------------	------------------------------	--



Label	Size of the Data
Description	<p>Defines the size of the <code>InfoType</code> data item in Bytes.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ 1 .. 255 Bytes <p>Note : A <code>VehInfoDataSize</code> with the value 0 is invalid</p>
Multiplicity	1..1
Type	INTEGER
Default value	0
Range	<p><=255</p> <hr/> <p>>=1</p>
Configuration class	<code>VariantPreCompile:</code> <code>VariantPreCompile</code>
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspVehInfoDataUsePort
Label	Use port interface to access the <code>InfoType</code>
Description	<p>Defines the type of the interface that shall be used to implement the <code>GetInfotypeValueData</code> operation in order to obtain the data for this <code>vehInfo</code>.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: The <code>GetInfotypeValueData</code> operation for this <code>vehInfo</code> is implemented by using a synchronous AUTOSAR ClientServerInterface with asynchronous behaviour(the operation shall be completed in a series of calls to the API, the first call being executed with an <code>OpStatus</code> set to <code>DCM_INITIAL</code>, and as long as the interface returns <code>E_PENDING</code>, it shall be called with an <code>OpStatus</code> set to <code>DCM_PENDING</code> for all subsequent executions). The <code>Dcm</code> generates the <code>InfotypeServices_&#60;VEHINFODATA&#62;</code> ClientServerInterface, where <code>&#60;VEHINFODATA&#62;</code> is the name of this <code>DcmDspVehInfoData</code> container. This interface shall implement the <code>GetInfotypeValueData</code> operation. A Require Port is generated for this interface and shall be connected to a compatible Provide Port of an Application Software Component. The name of this Require Port is the same as the name of the ClientServerInterface. ▶ FALSE: The <code>GetInfotypeValueData</code> operation for this <code>vehInfo</code> is implemented asynchronous API (the operation shall be completed in a series of calls to the API, the first call being executed with an <code>OpStatus</code> set to <code>DCM_INITIAL</code>, and as long as the interface returns <code>E_PENDING</code>, it shall



	<p>be called with an <code>OpStatus</code> set to <code>DCM_PENDING</code> for all subsequent executions). The following callout name shall be configured:</p> <ul style="list-style-type: none"> ▶ For operation <code>GetInfotypeValueData: DcmDspVehInfo-DataReadFnc</code>
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

5.2.1.93. DcmGeneral

Containers included		
Container name	Multiplicity	Description
DcmServiceAPI	1..1	<p>Label: Service API Parameters</p> <p>Container for configuration of the service API of Dcm.</p>

Parameters included	
Parameter name	Multiplicity
DcmHeaderFileInclusion	0..n
DcmDD DIDStorage	1..1
DcmDevErrorDetect	1..1
DcmDslEnableSetActiveDiagnosis	1..1
DcmComMMultiCoreSupport	1..1
DcmDynamicallyDefinedDIDInitPolicy	0..1
DcmDynamicallyDefinedPeriodicDIDInitPolicy	0..1
DcmNRCPolicyISOVersion	1..1
DcmModeDeclarationSupport	1..1
DcmRequestManufacturerNotificationEnabled	1..1
DcmRequestSeedAfterFailedCompareKey	1..1
DcmRequestSupplierNotificationEnabled	1..1
DcmRespondAllRequest	1..1
DcmRteUsage	1..1

**Parameters included**

DcmRteTypesProtectionAtFileLevel	1..1
DcmGenericConnectionUsage	1..1
DcmTaskTime	1..1
DcmVersionInfoApi	1..1
DcmDspDD DID Block Id Ref	1..1
DcmDataTransferServicesASRVersion	1..1
DcmDspVehInfoASRVersion	1..1
DcmDataTransferServicesMemoryRangeAndSecurityCheck	1..1
DcmDataTransferServicesMemoryIdCheck	1..1
DcmDspDataDefaultEndianness	0..1
DcmDslProtocolDID Endianness Conv Enabled	1..1
DcmDslProtocolRoutineEndiannessConvEnabled	1..1

Parameter Name	DcmHeaderFileInclusion	
Label	Custom standard header files	
Description	Name of the header file(s) to be included by the Dcm module containing the used Memory Range definitions.	
Multiplicity	0..n	
Type	STRING	
Configuration class	PreCompile:	VariantPreCompile

Parameter Name	DcmDD DID Storage	
Label	Store DD DID definitions to non-volatile memory	
Description	Defines whether DD DID definitions are stored non-volatile memory or not.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DcmDevErrorDetect	
Label	Enable Development Error Detection	
Description	Enables usage of Development Error Detection for the Dcm.	



	Range: <ul style="list-style-type: none"> ▶ TRUE: Enables Development Error Detection. ▶ FALSE: Disables Development Error Detection.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDslEnableSetActiveDiagnosis
Label	Allow disabling calls to ComM_DCM_ActiveDiagnostic()
Description	This switch allows to enable or disable a functionality that can be used to activate or deactivate the call to the <code>ComM_DCM_ActiveDiagnostic()</code> function. It is intended to be used to allow the ECU to go to sleep even if the diagnosis is ongoing when ignition was turned off and the ECU is in the default diagnostic session.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmComMMultiCoreSupport
Label	Enable DCM and ComM on different partitions
Description	Allows the mapping of DCM and ComM to different partitions. Range: <ul style="list-style-type: none"> ▶ TRUE: DCM and ComM are located at different partitions. ▶ FALSE: DCM and ComM are located at the same partition.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH



Parameter Name	DcmDynamicallyDefinedDIDInitPolicy	
Label	Policy initialization type for DDDID	
Description	<p>Defines the policy initialization type for DDDID.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DDDID_CLEAR_ALWAYS: At Dcm initialisation phase DDDIDs will be always cleared ▶ DDDID_CLEAR_ONLY_POWERON: DDDIDs will be cleared by the integrator 	
Multiplicity	0..1	
Type	ENUMERATION	
Default value	DDDID_CLEAR_ALWAYS	
Range	<p>DDDID_CLEAR_ALWAYS</p> <p>DDDID_CLEAR_ONLY_POWERON</p>	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmDynamicallyDefinedPeriodicDIDInitPolicy	
Label	Policy initialization type for periodic DDDID	
Description	<p>Defines the policy initialization type for dynamically defined periodic DID.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ PRDI_PERIODIC_CLEAR_ALWAYS: Periodic DDDIDs will be always cleared at Dcm initialisation phase ▶ PRDI_PERIODIC_CLEAR_ONLY_POWERON: Periodic DDDIDs will be cleared by the integrator ▶ PRDI_PERIODIC_CLEAR_NEVER: Periodic DDDIDs will be restored from NvM 	
Multiplicity	0..1	
Type	ENUMERATION	
Default value	PRDI_PERIODIC_CLEAR_ALWAYS	
Range	<p>PRDI_PERIODIC_CLEAR_ALWAYS</p> <p>PRDI_PERIODIC_CLEAR_ONLY_POWERON</p> <p>PRDI_PERIODIC_CLEAR_NEVER</p>	
Configuration class	PreCompile:	VariantPreCompile



Origin	Elektrobit Automotive GmbH
Parameter Name	DcmNRCPolicyISOVersion
Label	Define ISO
Description	<p>Defines handling of NRCs (Suppression of NRCs of functional requests, NRC to use in negative responses) according to the selected ISO version.</p> <ul style="list-style-type: none"> ▶ ISO14229_2006 = Handle NRCs according to ISO-14229-1 / 2006 ▶ ISO14229_2013 = Handle NRCs according to ISO-14229-1 / 2013 <p>Note:If UDS service RoutineControl (0x31) is requested:</p> <ul style="list-style-type: none"> ▶ ISO14229_2006 = The NRC requestSequenceError(0x24) is return by DCM module according with ISO_14229-2006. ▶ ISO14229_2013 = The next checks must be handled by application "routineControlOptionRecord contains valid data for the requested RID", "Condition check" and "Request sequence respected for the RID?" to be according with ISO 14229-1:2013 and AUTOSAR 4.2.1.
Multiplicity	1..1
Type	ENUMERATION
Default value	ISO14229_2006
Range	ISO14229_2006 ISO14229_2013
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmModeDeclarationSupport
Label	Enable Mode Declaration Support
Description	<p>Enables Mode Declaration Support for the Dcm</p> <p>If enabled, the Dcm acts as a mode manager and is responsible for providing the modes of the ModeDeclarationGroupPrototypes it defines in its Basic Software Module Description:</p> <ul style="list-style-type: none"> ▶ DcmDiagnosticSessionControl ▶ DcmEcuReset ▶ DcmModeRapidPowerShutDown ▶ DcmCommunicationControl_<symbolic name of ComMChan-nelId>.



	<ul style="list-style-type: none"> ▶ DcmControlDTCSetting <p>SchM_Switch APIs enable the <code>Dcm</code> to initiate a mode switch. The Basic Software Scheduler then instantiates a <code>ModeMachineInstance</code>. Mode users (eg. the <code>BswM</code>) can read the value of these <code>ModeDeclarationGroupPrototypes</code> via SchM_Mode APIs.</p> <p>Note 1: Autosar 4.0.1 used BswM callouts whenever a mode was changed and Autosar 4.0.3 uses SchM switch APIs for notifying the mode changes</p> <p>Note 2: This parameter is subject to removal in further releases.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: Enables Mode Declaration Support ▶ FALSE: Disables Mode Declaration Support
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmRequestManufacturerNotificationEnabled
Label	Enable Diagnostic Request Notification for Manufacturer
Description	<p>Enables the Diagnostic Request Notification for Manufacturer.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: Enables Manufacturer Request Notification. ▶ FALSE: Disables Manufacturer Request Notification. <p>By enabling the Manufacturer Request Notification, the <code>Dcm</code> will generate a Require Port for the <code>ServiceRequestNotification</code> interface containing the following operations:</p> <ul style="list-style-type: none"> ▶ <code>Xxx_Indication</code> - this operation is called right before processing the diagnostic request in order to check permissions for processing. Depending on the implemented functionality, the diagnostic request will be further processed, ignored or be rejected with a Negative Response Code ▶ <code>Xxx_Confirmation</code> - this operation is called after confirmation of the diagnostic response transmission as a result of request processing, or at the end of request processing if the transmission of a response is not required.



Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	
Parameter Name	DcmRequestSeedAfterFailedCompareKey	
Label	Enable a strict SecurityAccess mechanism	
Description	<p>Enables strict sequencing for the execution of the SecurityAccess (0x27) service.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: The SecurityAccess service handler execution is reset to its default state by any SendKey sub-function failure. ▶ FALSE: The SecurityAccess service is tolerant to out-of-sequence SendKey requests. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	
Parameter Name	DcmRequestSupplierNotificationEnabled	
Label	Enable Diagnostic Request Notification for Supplier	
Description	<p>Enables the Diagnostic Request Notification for Supplier.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: Enables Supplier Request Notification. ▶ FALSE: Disables Supplier Request Notification. <p>By enabling the Supplier Request Notification, the Dcm will generate a Require Port for the ServiceRequestNotification interface containing the following operations:</p> <ul style="list-style-type: none"> ▶ Xxx_Indication - this operation is called right before processing the diagnostic request in order to check permissions for processing. Depending on the implemented functionality, the diagnostic request will be further processed, ignored or be rejected with a Negative Response Code 	



	► <code>Xxx_Confirmation</code> - this operation is called after confirmation of the diagnostic response transmission as a result of request processing, or at the end of request processing if the transmission of a response is not required.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmRespondAllRequest
Label	Enable Response for all Diagnostic Requests
Description	<p>Enables if the <code>Dcm</code> responds to all diagnostic requests.</p> <p>The <code>Dcm</code> can be configured not to respond to diagnostic service requests with service IDs in the ranges 0x40 .. 0x7F and 0xC0 .. 0x7F.</p> <p>Range:</p> <ul style="list-style-type: none"> ► TRUE: The <code>Dcm</code> sends the Negative Response Code <code>serviceNotSupported</code> (0x11). ► FALSE: The <code>Dcm</code> will not respond. <p>Note: the functionality related to this parameter is not supported by the current implementation.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmRteUsage
Label	Enable RTE Usage
Description	<p>Note : This is a Vendor Specific parameter for RTE usage by the Dcm.</p> <p>Enables RTE functionality usage by the <code>Dcm</code>.</p> <p>By disabling this configuration parameter, <code>Dcm</code> functionality which is dependent on the presence of the <code>RTE</code> is disabled.</p> <p>Range:</p>



	<ul style="list-style-type: none"> ▶ TRUE: Enables RTE feature usage. ▶ FALSE: Disables RTE feature usage.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmRteTypesProtectionAtFileLevel
Label	Enable protection of RTE types at file level
Description	<p>Note : This parameter is for choosing a type of protection of the types that are generated by RTE.</p> <p>By enabling this configuration parameter, protection at file level is used.</p> <p>By disabling this configuration parameter, double type definition protection is used.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: DCM uses protection at file level. ▶ FALSE: DCM uses double type definition protection at type level.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmGenericConnectionUsage
Label	Enable Generic Connection Usage
Description	<p>Note : This is a Vendor Specific parameter for Generic Connection usage by the Dcm.</p> <p>Enables EB custom feature for Generic Connections in order to obtain the MetaData information from connection using EB EcuC APIs and provide them to:</p> <ul style="list-style-type: none"> ▶ the Indication and Confirmation operations of the ServiceRequest-Notification client-server interface. ▶ the lower layer using the EcuC APIs



	<p>By disabling this configuration parameter, Dcm functionality which is dependent on the presence of the Generic Connection is disabled.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: Enables Generic Connection feature usage. ▶ FALSE: Disables Generic Connection feature usage.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmTaskTime
Label	Periodic Task time
Description	Defines the scheduling time for the periodic task running the Dcm_MainFunction(). The Dcm_MainFunction() is executed after every DcmTaskTime, and the value configured in this parameter defines the resolution of all timing dependent functionality of the Dcm_MainFunction().
	<p>Range:</p> <ul style="list-style-type: none"> ▶ 0.001 .. 1 seconds
Multiplicity	1..1
Type	FLOAT
Default value	0.005
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DcmVersionInfoApi
Label	Enable Version Info API
Description	Enables the Version Information API for the Dcm. The Dcm_GetVersionInfo() shall return the version information of this module. The version information includes: <ul style="list-style-type: none"> ▶ Module Id ▶ Vendor Id ▶ Module Major Version Number



	<ul style="list-style-type: none"> ▶ Module Minor Version Number ▶ Module Patch Version Number <p>Range:</p> <ul style="list-style-type: none"> ▶ TRUE: The <code>Dcm_GetVersionInfo()</code> API is available. ▶ FALSE: The <code>Dcm_GetVersionInfo()</code> API is not available.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmDspDD DIDBlockIdRef
Label	Reference to the NVRAM blockId to store DDDID data
Description	Reference to NVRAM block id to store DDDID data. Note : Only relevant if <code>DcmDD DIDStorage</code> = TRUE.
Multiplicity	1..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDataTransferServicesASRVersion
Label	TransferData services AUTOSAR version
Description	<p>Defines the AUTOSAR version compatibility for the following services: RequestDownload (0x34), RequestUpload (0x35), TransferData (0x36), RequestTransferExit (0x37), ReadMemoryByAddress (0x23) and WriteMemoryByAddress (0x3D) Valid options: AUTOSAR 4.0.3, AUTOSAR 4.2.2 or AUTOSAR 4.3.0</p> <ul style="list-style-type: none"> ▶ AUTOSAR_40 = The implementation and the callouts are AUTOSAR 4.0 compliant ▶ AUTOSAR_42 = The implementation and the callouts are AUTOSAR 4.2.2 compliant ▶ AUTOSAR_43 = The implementation and the callouts are AUTOSAR 4.3 compliant
Multiplicity	1..1



Type	ENUMERATION	
Default value	AUTOSAR_40	
Range	AUTOSAR_40 AUTOSAR_422 AUTOSAR_43	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmDspVehInfoASRVersion	
Label	Request Vehicle Information service AUTOSAR version	
Description	Defines the AUTOSAR version compatibility for the OBD RequestVehicleInformation (0x09) Service <ul style="list-style-type: none"> ► AUTOSAR_403 = The implementation and the callouts are AUTOSAR 4.0.3 compliant ► AUTOSAR_440 = The implementation and the callouts are AUTOSAR 4.4.0 compliant 	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	AUTOSAR_403	
Range	AUTOSAR_403 AUTOSAR_440	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmDataTransferServicesMemoryRangeAndSecurityCheck	
Label	Enable optional checks for TransferData services	
Description	Configure whether the following checks shall be done on receiving a RequestDownload (0x34) or RequestUpload (0x35) service request: <ul style="list-style-type: none"> ► If the range starting from the requested MemoryAddress until MemoryAddress + MemorySize fits into a configured DcmDspWriteMemoryRangeInfo in case RequestDownload was requested, respectively DcmDspReadMemoryRangeInfo in case RequestUpload was requested ► If the requested read or write operation is allowed in the current security level 	



	Range:
	<ul style="list-style-type: none"> ▶ TRUE = Enables the checks above ▶ FALSE = Disables the checks above
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDataTransferServicesMemoryIdCheck
Label	Enable MemoryId check for TransferData services
Description	<p>Check if the requested MemoryId identified as the most significant byte of the requested <code>memoryAddress</code> matches a configured <code>DcmDspMemoryIdValue</code> for a configured <code>DcmDspWriteMemoryRangeInfo</code> in case <code>RequestDownload</code> was requested, respectively <code>DcmDspReadMemoryRangeInfo</code> in case <code>RequestUpload</code> was requested</p> <ul style="list-style-type: none"> ▶ TRUE = Enables the check above ▶ FALSE = Disables the check above
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DcmDspDataDefaultEndianness
Label	DcmDspDataDefaultEndianness
Description	<p>Defines the default endianness belonging to a DID, RID or PID if the corresponding data does not define an endianness.</p> <ul style="list-style-type: none"> ▶ BIG_ENDIAN = Most significant byte shall be stored at the lowest address. ▶ LITTLE_ENDIAN = Most significant byte shall be stored at the highest address ▶ OPAQUE = Opaque data endianness
Multiplicity	0..1
Type	ENUMERATION



Default value	OPAQUE	
Range	BIG_ENDIAN	
	LITTLE_ENDIAN	
	OPAQUE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DcmDslProtocolDIDEndiannessConvEnabled	
Label	Enable endianness conversion for DID signals	
Description	<p>Note: This parameter has significance only when DcmDslProtocolEndianness-ConvEnabled is enabled for the protocol on which the request is processed.</p> <p>Enable endianness conversion for DID signals</p> <ul style="list-style-type: none"> ▶ TRUE = Enable endianness conversion for DID signals ▶ FALSE = Disable endianness conversion for DID signals 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmDslProtocolRoutineEndiannessConvEnabled	
Label	Enable endianness conversion for routine signals	
Description	<p>Note: This parameter has significance only when DcmDslProtocolEndianness-ConvEnabled is enabled for the protocol on which the request is processed.</p> <p>Enable endianness conversion for routine signals</p> <ul style="list-style-type: none"> ▶ TRUE = Enable endianness conversion for routine signals ▶ FALSE = Disable endianness conversion for routine signals 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	



5.2.1.94. DcmServiceAPI

Parameters included	
Parameter name	Multiplicity
DcmEnableASR40ServiceAPI	1..1
DcmEnableASR42ServiceAPI	1..1
DcmDefaultASRServiceAPI	1..1

Parameter Name	DcmEnableASR40ServiceAPI	
Label	Enable AUTOSAR 4.0 service API	
Description	Configures whether the AUTOSAR 4.0 service API shall be provided. <ul style="list-style-type: none"> ▶ TRUE = Enables AUTOSAR 4.0 service API. ▶ FALSE = Disables AUTOSAR 4.0 service API. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmEnableASR42ServiceAPI	
Label	Enable AUTOSAR 4.2 service API	
Description	Configures whether the AUTOSAR 4.2 service API shall be provided. <ul style="list-style-type: none"> ▶ TRUE = Enables AUTOSAR 4.2 service API. ▶ FALSE = Disables AUTOSAR 4.2 service API. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DcmDefaultASRServiceAPI	
Label	Default AUTOSAR service API	
Description	Defines the default AUTOSAR service API.	



	<ul style="list-style-type: none"> ▶ AUTOSAR_40 = AUTOSAR 4.0 service API is the default one. ▶ AUTOSAR_42 = AUTOSAR 4.2 service API is the default one.
Multiplicity	1..1
Type	ENUMERATION
Default value	AUTOSAR_40
Range	AUTOSAR_40 AUTOSAR_42
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.2.1.95. DcmPageBufferCfg

Parameters included	
Parameter name	Multiplicity
DcmPagedBufferEnabled	1..1
DcmPagedBufferTimeout	0..1

Parameter Name	DcmPagedBufferEnabled
Label	Enable Paged Buffering
Description	Enables the Paged Buffering mechanism for response transmission. Range: <ul style="list-style-type: none"> ▶ TRUE: Enables Paged Buffered transmission. ▶ FALSE: Disables Paged Buffered transmission.
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmPagedBufferTimeout
Label	Timeout for filling one page with data
Description	Defines the timeout for Paged Buffered transmissions in seconds.



	This timeout represents the maximum time to wait for a new page to be made available by the Diagnostic Service Processor (DSP) submodule of the <code>Dcm</code> for transmission before the processing of the request is cancelled.
	Range: <ul style="list-style-type: none">▶ 0 .. 1 seconds
Multiplicity	0..1
Type	FLOAT
Default value	0.0
Range	<=1 >=0
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.2.1.96. DcmProcessingConditions

Containers included		
Container name	Multiplicity	Description
DcmModeCondition	1..n	Label: Mode Condition Defines the configuration of a mode condition which can be used as an argument of <code>DcmModeRules</code> .
DcmModeRule	1..n	Label: Mode Rule Defines the configuration of mode rules, which represent logical expressions having <code>DcmModeConditions</code> and/or other <code>DcmModeRules</code> as arguments.

5.2.1.97. DcmModeCondition

Containers included		
Container name	Multiplicity	Description
DcmBswModeRef	0..1	Label: Bsw Mode Reference Defines the parameters needed in order to specify a mode rule using modes provided by Basic Software Modules.

**Containers included**

DcmSwcModeRef	0..1	Label: Swc Mode Reference Defines the parameters needed in order to specify a <code>DcmModeRule</code> using modes provided by Software Components.
-------------------------------	------	---

Parameters included

Parameter name	Multiplicity
DcmConditionType	1..1

Parameter Name	DcmConditionType
Label	Condition Type
Description	<p>Defines what kind of comparison is made for the evaluation of the mode condition.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ <code>DCM_EQUALS</code>: The <code>DcmModeCondition</code> evaluates to TRUE if, in <code>DcmBswModeRef</code> or <code>DcmSwcModeRef</code>, the current Mode of the <code>ModeDeclarationGroupPrototype</code> to which the referenced <code>ModeDeclaration</code> belongs equals the referenced <code>ModeDeclaration</code> ▶ <code>DCM_EQUALS_NOT</code>: The <code>DcmModeCondition</code> evaluates to TRUE if, in <code>DcmBswModeRef</code> or <code>DcmSwcModeRef</code>, the current Mode of the <code>ModeDeclarationGroupPrototype</code> to which the referenced <code>ModeDeclaration</code> belongs does not equal the referenced <code>ModeDeclaration</code>
Multiplicity	1..1
Type	ENUMERATION
Range	<code>DCM_EQUALS</code> <code>DCM_EQUALS_NOT</code>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

5.2.1.98. DcmBswModeRef**Parameters included**

Parameter name	Multiplicity
TARGET	1..1



Parameters included

CONTEXT	0..n
-------------------------	------

Parameter Name	TARGET
Description	References a ModeDeclaration belonging to a ModeDeclarationGroup-Prototype provided by a Basic Software Module, to be used for the DcmMode-Condition being configured.
Multiplicity	1..1
Type	REFERENCE
Origin	AUTOSAR_ECUC

Parameter Name

[CONTEXT](#)

Multiplicity

0..n

Type

REFERENCE

Origin

AUTOSAR_ECUC

5.2.1.99. DcmSwcModeRef

Parameters included

Parameter name	Multiplicity
TARGET	1..1
CONTEXT	0..n

Parameter Name

[TARGET](#)

[CONTEXT](#)

0..n

Parameter Name

[TARGET](#)

Description

References a ModeDeclaration belonging to a ModeDeclarationGroup-Prototype provided by a Software Component, to be used for the DcmMode-Condition being configured.

Multiplicity

1..1

Type

REFERENCE

Origin

AUTOSAR_ECUC

Parameter Name

[CONTEXT](#)

Multiplicity

0..n

Type

REFERENCE

Range

ROOT-SW-COMPOSITION-PROTOTYPE



	SW-COMPONENT-PROTOTYPE
	P-PORT-PROTOTYPE
	MODE-DECLARATION-GROUP-PROTOTYPE
Origin	AUTOSAR_ECUC

5.2.1.100. DcmModeRule

Parameters included	
Parameter name	Multiplicity
DcmLogicalOperator	1..1
DcmModeRuleNrcValue	0..1
DcmArgumentRef	1..n

Parameter Name	DcmLogicalOperator
Label	Logical Operator
Description	<p>Defines the logical operator to be used in the logical expression that defines a DcmModeRule. If the expression only consists of a single condition this parameter shall not be used.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DCM_AND: The value of the DcmModeRule shall be the value of the AND-ed operands ▶ DCM_OR: The value of the DcmModeRule shall be the value of the OR-ed operands
Multiplicity	1..1
Type	ENUMERATION
Range	DCM_AND DCM_OR
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DcmModeRuleNrcValue
Label	Mode Rule Nrc Value
Description	Defines the Negative Response Code to be sent in case the DcmModeRule does not evaluate to TRUE.



	This parameter is optional. In case it is disabled, 0x22 (ConditionsNotCorrect) is used as the default Negative Response Code.
Multiplicity	0..1
Type	INTEGER
Range	<=255 >=1
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DcmArgumentRef
Label	Argument Reference
Description	Defines a choice reference to either a DcmModeCondition or a DcmModeRule to be used as an operand of this DcmModeRule.
Multiplicity	1..n
Type	CHOICE-REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

5.2.1.101. PublishedInformation

Parameters included	
Parameter name	Multiplicity
PbcfgMSupport	1..1

Parameter Name	PbcfgMSupport
Label	PbcfgM support
Description	Specifies whether or not the Dcm can use the PbcfgM module for post-build support.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH



5.2.2. Recommended configurations

5.2.2.1. DcmRecConfigurationBasic

Containers included	
Container name	Container definition
DcmConfigSet_0	DcmConfigSet

Parameters included	
Parameter name	Value
IMPLEMENTATION_CONFIG_VARIANT	VariantPreCompile

5.2.2.1.1. DcmConfigSet_0

Containers included	
Container name	Container definition
DcmDsd	DcmDsd
DcmDsp	DcmDsp
DcmDsl	DcmDsl
DcmDefensiveProgramming	DcmDefensiveProgramming
DcmGeneral	DcmGeneral
DcmPageBufferCfg	DcmPageBufferCfg
DcmProcessingConditions	DcmProcessingConditions

Parameters included	
Parameter name	Value

5.2.2.1.2. DcmDsd

Containers included	
Container name	Container definition

**Containers included**

DcmDsdServiceTable_0	DcmDsdServiceTable
--------------------------------------	------------------------------------

Parameters included

Parameter name	Value
----------------	-------

5.2.2.1.3. DcmDsdServiceTable_0**Containers included**

Container name	Container definition
TesterPresent	DcmDsdService
DiagnosticSessionControl	DcmDsdService
EcuReset	DcmDsdService
SecurityAccess	DcmDsdService
ReadDid	DcmDsdService
WriteDid	DcmDsdService
ReadDTCInformation	DcmDsdService
ClearDiagnosticInformation	DcmDsdService
CommunicationControl	DcmDsdService
ControlDTCSetting	DcmDsdService

Parameters included

Parameter name	Value
DcmDsdSidTabId	0

5.2.2.1.4. TesterPresent**Parameters included**

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	62
DcmDsdSidTabSubfuncAvail	true



5.2.2.1.5. DiagnosticSessionControl

Containers included	
Container name	Container definition
defaultSession	DcmDsdSubService
programmingSession	DcmDsdSubService
extendedDiagnosticSession	DcmDsdSubService

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	16
DcmDsdSidTabSubfuncAvail	true

5.2.2.1.6. defaultSession

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	1

5.2.2.1.7. programmingSession

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	2

5.2.2.1.8. extendedDiagnosticSession

Parameters included	
Parameter name	Value

**Parameters included**

DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	3

5.2.2.1.9. EcuReset**Containers included**

Container name	Container definition
hardReset	DcmDsdSubService
keyOffOnReset	DcmDsdSubService
softReset	DcmDsdSubService
enableRapidPowerShutDown	DcmDsdSubService
disableRapidPowerShutDown	DcmDsdSubService

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	17
DcmDsdSidTabSubfuncAvail	true

5.2.2.1.10. hardReset**Parameters included**

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	1

5.2.2.1.11. keyOffOnReset**Parameters included**

Parameter name	Value



Parameters included

DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	2

5.2.2.1.12. softReset

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	3

5.2.2.1.13. enableRapidPowerShutDown

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	4

5.2.2.1.14. disableRapidPowerShutDown

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	5

5.2.2.1.15. SecurityAccess

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceld	39



Parameters included

DcmDsdSidTabSubfuncAvail	true
--	------

5.2.2.1.16. ReadDid

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	34
DcmDsdSidTabSubfuncAvail	false

5.2.2.1.17. WriteDid

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	46
DcmDsdSidTabSubfuncAvail	false

5.2.2.1.18. ReadDTCInformation

Containers included

Container name	Container definition
reportNumberOfDTCByStatusMask	DcmDsdSubService
reportDTCByStatusMask	DcmDsdSubService
reportDTCSnapshotRecordIdentification	DcmDsdSubService
reportDTCSnapshotRecordByDTCNumber	DcmDsdSubService
reportDTCExtendedDataRecordByDTCNumber	DcmDsdSubService
reportSupportedDTCs	DcmDsdSubService



Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	25
DcmDsdSidTabSubfuncAvail	true

5.2.2.1.19. reportNumberOfDTCByStatusMask

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	1

5.2.2.1.20. reportDTCByStatusMask

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	2

5.2.2.1.21. reportDTCSnapshotRecordIdentification

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	3

5.2.2.1.22. reportDTCSnapshotRecordByDTCNumber

Parameters included

Parameter name	Value



Parameters included

DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	4

5.2.2.1.23. reportDTCExtendedDataRecordByDTCNumber

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	6

5.2.2.1.24. reportSupportedDTCs

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	10

5.2.2.1.25. ClearDiagnosticInformation

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceld	20
DcmDsdSidTabSubfuncAvail	false

5.2.2.1.26. CommunicationControl

Containers included

Container name	Container definition
enableRxAndTx	DcmDsdSubService

**Containers included**

enableRxAndDisableTx	DcmDsdSubService
disableRxAndEnableTx	DcmDsdSubService
disableRxAndTx	DcmDsdSubService

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	40
DcmDsdSidTabSubfuncAvail	true

5.2.2.1.27. enableRxAndTx**Parameters included**

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	0

5.2.2.1.28. enableRxAndDisableTx**Parameters included**

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	1

5.2.2.1.29. disableRxAndEnableTx**Parameters included**

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	2



5.2.2.1.30. disableRxAndTx

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	3

5.2.2.1.31. ControlDTCSetting

Containers included	
Container name	Container definition
DcmDsdSubService_0	DcmDsdSubService
DcmDsdSubService_1	DcmDsdSubService

Parameters included

Parameter name	Value
DcmDsdServiceUsed	true
DcmDsdSidTabServiceId	133
DcmDsdSidTabSubfuncAvail	true
DcmAsyncServiceExecution	false
DcmDsdSidTabFnc	(DISABLED)

5.2.2.1.32. DcmDsdSubService_0

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	1

5.2.2.1.33. DcmDsdSubService_1

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)

**Parameters included**

DcmDsdSubServiceld	2
------------------------------------	---

5.2.2.1.34. DcmDsp**Containers included**

Container name	Container definition
DcmDspComControl	DcmDsp DcmDspComControl
DcmDspControlDTCSetting	DcmDsp DcmDspControlDTCSetting
DcmDspPeriodicDidTransmission	DcmDsp DcmDspPeriodicDidTransmission
DcmDspPeriodicTransmission	DcmDsp DcmDspPeriodicTransmission
DcmDspReadGenericInformation	DcmDsp DcmDspReadGenericInformation
DcmDspSession	DcmDspSession
DcmDspData_0x100	DcmDspData
DcmDspData_0xF186	DcmDspData
DcmDspDataInfo_0x100	DcmDspDataInfo
DcmDspDataInfo_0xF186	DcmDspDataInfo
DcmDspDid_0x100	DcmDspDid
DcmDspDid_0xF186	DcmDspDid
DcmDspDidInfo_0x100	DcmDspDidInfo
DcmDspDidInfo_0xF186	DcmDspDidInfo
DcmDspRoe	DcmDsp DcmDspRoe
DcmDspRoutine_0	DcmDspRoutine
DcmDspRoutineInfo_0	DcmDspRoutineInfo
DcmDspSecurity	DcmDsp DcmDspSecurity
DcmDspTestResultByObdmid	DcmDsp DcmDspTestResultByObdmid
DcmDspMemory	DcmDsp DcmDspMemory

Parameters included

Parameter name	Value
DcmDspDDIDCheckPerSourceDID	false
DcmDspGenerateOnlyNeededIOControlOperations	false



Parameters included

DcmDspMaxPeriodicDidToRead	1
DcmDspReturnControlToEcuTimeout	1
DcmDspSessionControlTiming	false
DcmDspMaxDidToRead	1
DcmDspPowerDownTime	10
DcmDspRoutineVariableLengthInBytes	false

5.2.2.1.35. DcmDspComControl

Containers included

Container name	Container definition
DcmDspComControlSetting	DcmDspComControl DcmDspComControlSetting

Parameters included

Parameter name	Value

5.2.2.1.36. DcmDspComControlSetting

Parameters included

Parameter name	Value

5.2.2.1.37. DcmDspControlDTCSetting

Parameters included

Parameter name	Value
DcmSupportDTCSSettingControlOptionRecord	false

5.2.2.1.38. DcmDspPeriodicDidTransmission

Parameters included

Parameter name	Value



Parameters included

DcmDspMaxPeriodicDidScheduler	(DISABLED)
DcmDspPeriodicDidSamplingBufferSize	(DISABLED)
DcmDspPeriodicDidTransmissionRingEntries	(DISABLED)
DcmDspPeriodicDDIDMaxLength	(DISABLED)

5.2.2.1.39. DcmDspPeriodicTransmission

Parameters included

Parameter name	Value
DcmDspPeriodicTransmissionSlowRate	(DISABLED)
DcmDspPeriodicTransmissionMediumRate	(DISABLED)
DcmDspPeriodicTransmissionFastRate	(DISABLED)

5.2.2.1.40. DcmDspReadGenericInformation

Parameters included

Parameter name	Value
DcmDspMaxDemGetDiagnosticDataConsecutive-Executions	(DISABLED)

5.2.2.1.41. DcmDspSession

Containers included

Container name	Container definition
DCM_DEFAULT_SESSION	DcmDspSessionRow
DCM_EXTENDED_DIAGNOSTIC_SESSION	DcmDspSessionRow
DCM_PROGRAMMING_SESSION	DcmDspSessionRow
DCM_ALL_SESSION_LEVEL	DcmDspSessionRow

Parameters included

Parameter name	Value



5.2.2.1.42. DCM_DEFAULT_SESSION

Parameters included	
Parameter name	Value
DcmDspSessionLevel	1
DcmDspSessionForBoot	DCM_NO_BOOT
DcmDspSessionP2ServerMax	0.05
DcmDspSessionP2StarServerMax	5.0

5.2.2.1.43. DCM_EXTENDED_DIAGNOSTIC_SESSION

Parameters included	
Parameter name	Value
DcmDspSessionForBoot	DCM_NO_BOOT
DcmDspSessionLevel	3
DcmDspSessionP2ServerMax	0.1
DcmDspSessionP2StarServerMax	2.0

5.2.2.1.44. DCM_PROGRAMMING_SESSION

Parameters included	
Parameter name	Value
DcmDspSessionForBoot	DCM_NO_BOOT
DcmDspSessionLevel	2
DcmDspSessionP2ServerMax	0.5
DcmDspSessionP2StarServerMax	7.0

5.2.2.1.45. DCM_ALL_SESSION_LEVEL

Parameters included	
Parameter name	Value
DcmDspSessionForBoot	DCM_NO_BOOT
DcmDspSessionLevel	255



Parameters included

DcmDspSessionP2ServerMax	0.5
DcmDspSessionP2StarServerMax	5.0

5.2.2.1.46. DcmDspData_0x100

Parameters included

Parameter name	Value
DcmDspDataConditionCheckReadFncUsed	true
DcmDspDataConditionCheckReadFnc	ReadDIDDataCondition0
DcmDspDataEcuSignal	(DISABLED)
DcmDspDataFreezeCurrentStateFnc	(DISABLED)
DcmDspDataGetScalingInfoFnc	(DISABLED)
DcmDspDataReadDataLengthFnc	ReadDIDDataLength0
DcmDspDataReadEcuSignal	(DISABLED)
DcmDspDataReadFnc	ReadDIDData0
DcmDspDataResetToDefaultFnc	(DISABLED)
DcmDspDataReturnControlToEcuFnc	(DISABLED)
DcmDspDataShortTermAdjustmentFnc	(DISABLED)
DcmDspDataSize	5
DcmDspDataType	UINT8
DcmDspDataUsePort	USE_DATA_ASYNCH_FNC
DcmDspDataWriteFnc	WriteDIDData0

5.2.2.1.47. DcmDspData_0xF186

Parameters included

Parameter name	Value
DcmDspDataConditionCheckReadFncUsed	false
DcmDspDataConditionCheckReadFnc	(DISABLED)
DcmDspDataEcuSignal	(DISABLED)
DcmDspDataFreezeCurrentStateFnc	(DISABLED)



Parameters included

DcmDspDataGetScalingInfoFnc	(DISABLED)
DcmDspDataReadDataLengthFnc	(DISABLED)
DcmDspDataReadEcuSignal	(DISABLED)
DcmDspDataReadFnc	Dcm_Internalf186Readout
DcmDspDataResetToDefaultFnc	(DISABLED)
DcmDspDataReturnControlToEcuFnc	(DISABLED)
DcmDspDataShortTermAdjustmentFnc	(DISABLED)
DcmDspDataSize	8
DcmDspDataType	UINT8
DcmDspDataUsePort	USE_DATA_SYNCH_FNC
DcmDspDataWriteFnc	(DISABLED)

5.2.2.1.48. DcmDspDataInfo_0x100

Parameters included

Parameter name	Value
DcmDspDataFixedLength	true
DcmDspDataScalingInfoSize	(DISABLED)

5.2.2.1.49. DcmDspDataInfo_0xF186

Parameters included

Parameter name	Value
DcmDspDataFixedLength	true
DcmDspDataScalingInfoSize	(DISABLED)

5.2.2.1.50. DcmDspDid_0x100

Containers included

Container name	Container definition
DcmDspDidSignal_0	DcmDspDidSignal

**Parameters included**

Parameter name	Value
DcmDspDidIdentifier	256
DcmDspDidUsed	true

5.2.2.1.51. DcmDspDidSignal_0**Parameters included**

Parameter name	Value
DcmDspDidDataPos	2

5.2.2.1.52. DcmDspDid_0xF186**Containers included**

Container name	Container definition
DcmDspDidSignal_0XF186	DcmDspDidSignal

Parameters included

Parameter name	Value
DcmDspDidIdentifier	61830
DcmDspDidUsed	true

5.2.2.1.53. DcmDspDidSignal_0XF186**Parameters included**

Parameter name	Value
DcmDspDidDataPos	0

5.2.2.1.54. DcmDspDidInfo_0x100**Containers included**

Container name	Container definition



Containers included

DcmDspDidAccess	DcmDspDidAccess
---------------------------------	---------------------------------

Parameters included

Parameter name	Value
DcmDspDDIDMaxElements	1
DcmDspDidDynamicallyDefined	false

5.2.2.1.55. DcmDspDidAccess

Containers included

Container name	Container definition
DcmDspDidControl	DcmDspDidAccess DcmDspDidControl
DcmDspDidRead	DcmDspDidAccess DcmDspDidRead
DcmDspDidWrite	DcmDspDidAccess DcmDspDidWrite

Parameters included

Parameter name	Value
----------------	-------

5.2.2.1.56. DcmDspDidControl

Parameters included

Parameter name	Value
DcmDspDidFreezeCurrentState	(DISABLED)
DcmDspDidResetToDefault	(DISABLED)
DcmDspDidReturnControlToEcu	(DISABLED)
DcmDspDidShortTermAdjustment	(DISABLED)

5.2.2.1.57. DcmDspDidRead

Parameters included

Parameter name	Value
----------------	-------



5.2.2.1.58. DcmDspDidWrite

Parameters included	
Parameter name	Value

5.2.2.1.59. DcmDspDidInfo_0xF186

Containers included	
Container name	Container definition
DcmDspDidAccess	DcmDspDidAccess
Parameters included	
Parameter name	Value
DcmDspDidDynamicallyDefined	false
DcmDspDDIDMaxElements	1

5.2.2.1.60. DcmDspDidAccess

Containers included	
Container name	Container definition
DcmDspDidControl	DcmDspDidAccess DcmDspDidControl
DcmDspDidRead	DcmDspDidAccess DcmDspDidRead
DcmDspDidWrite	DcmDspDidAccess DcmDspDidWrite
Parameters included	
Parameter name	Value

5.2.2.1.61. DcmDspDidControl

Parameters included	
Parameter name	Value
DcmDspDidFreezeCurrentState	(DISABLED)
DcmDspDidResetToDefault	(DISABLED)
DcmDspDidReturnControlToEcu	(DISABLED)
DcmDspDidShortTermAdjustment	(DISABLED)



5.2.2.1.62. DcmDspDidRead

Parameters included	
Parameter name	Value

5.2.2.1.63. DcmDspDidWrite

Parameters included	
Parameter name	Value

5.2.2.1.64. DcmDspRoe

Containers included	
Container name	Container definition
DcmDspRoeEvent_0	DcmDspRoeEvent
DcmDspRoeEventWindowTime_0	DcmDspRoeEventWindowTime

Parameters included	
Parameter name	Value
DcmDspRoeInterMessageTime	0.0

5.2.2.1.65. DcmDspRoeEvent_0

Parameters included	
Parameter name	Value
DcmDspRoeEventId	0
DcmDspRoeInitialEventStatus	DCM_ROE_CLEARED

5.2.2.1.66. DcmDspRoeEventProperties

Containers included	
Container name	Container definition
DcmDspRoeOnChangeOfDataIdentifier	DcmDspRoeEventProperties DcmDspRoeOnChangeOfDataIdentifier



Containers included

DcmDspRoeOnDTCStatusChange	DcmDspRoeEventProperties DcmDspRoeOnDTCStatusChange
--	---

Parameters included

Parameter name	Value
----------------	-------

5.2.2.1.67. DcmDspRoeOnChangeOfDataIdentifier

Parameters included

Parameter name	Value
----------------	-------

5.2.2.1.68. DcmDspRoeOnDTCStatusChange

Parameters included

Parameter name	Value
DcmDspRoeEventPreconfiguredDTCMask	0

5.2.2.1.69. DcmDspRoeEventWindowTime_0

Parameters included

Parameter name	Value
DcmDspRoeEventWindowTime	DCM_ROE_EVENT_WINDOW_INFINITE
DcmDspRoeStorageState	false

5.2.2.1.70. DcmDspRoutine_0

Parameters included

Parameter name	Value
DcmDspRequestResultsRoutineFnc	Routine0_RequestResult
DcmDspRoutineFixedLength	true
DcmDspRoutineIdentifier	65530



Parameters included

DcmDspRoutineUsePort	false
DcmDspRoutineUsed	true
DcmDspRoutineUsePortASRVersion	(DISABLED)
DcmDspStartRoutineFnc	Routine0_Start
DcmDspStopRoutineFnc	Routine0_Stop
DcmDspRequestResultsRoutineSupported	true
DcmDspStopRoutineSupported	true

5.2.2.1.71. DcmDspRoutineInfo_0

Containers included

Container name	Container definition
DcmDspRoutineAuthorization	DcmDspRoutineAuthorization
DcmDspRoutineRequestResIn	DcmDspRoutineRequestResIn
DcmDspRoutineRequestResOut	DcmDspRoutineRequestResOut
DcmDspRoutineStopIn	DcmDspRoutineStopIn
DcmDspRoutineStopOut	DcmDspRoutineStopOut
DcmDspStartRoutineIn	DcmDspStartRoutineIn
DcmDspStartRoutineOut	DcmDspStartRoutineOut

Parameters included

Parameter name	Value

5.2.2.1.72. DcmDspRoutineAuthorization

Parameters included

Parameter name	Value

5.2.2.1.73. DcmDspRoutineRequestResIn

Parameters included

Parameter name	Value



5.2.2.1.74. DcmDspRoutineRequestResOut

Containers included	
Container name	Container definition
DcmDspRoutineRequestResOutSignal_0	DcmDspRoutineRequestResOutSignal
DcmDspRoutineRequestResOutSignal_1	DcmDspRoutineRequestResOutSignal

Parameters included

Parameter name	Value
----------------	-------

5.2.2.1.75. DcmDspRoutineRequestResOutSignal_0

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	8
DcmDspRoutineSignalPos	0
DcmDspRoutineSignalType	UINT8

5.2.2.1.76. DcmDspRoutineRequestResOutSignal_1

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	1
DcmDspRoutineSignalPos	8
DcmDspRoutineSignalType	BOOLEAN

5.2.2.1.77. DcmDspRoutineStopIn

Containers included	
Container name	Container definition
DcmDspRoutineStopInSignal_0	DcmDspRoutineStopInSignal
DcmDspRoutineStopInSignal_1	DcmDspRoutineStopInSignal

Parameters included

Parameter name	Value
----------------	-------



5.2.2.1.78. DcmDspRoutineStopInSignal_0

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	16
DcmDspRoutineSignalPos	0
DcmDspRoutineSignalType	UINT16

5.2.2.1.79. DcmDspRoutineStopInSignal_1

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	8
DcmDspRoutineSignalPos	16
DcmDspRoutineSignalType	UINT8

5.2.2.1.80. DcmDspRoutineStopOut

Containers included	
Container name	Container definition
DcmDspRoutineStopOutSignal_0	DcmDspRoutineStopOutSignal
DcmDspRoutineStopOutSignal_1	DcmDspRoutineStopOutSignal

Parameters included

Parameter name	Value

5.2.2.1.81. DcmDspRoutineStopOutSignal_0

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	1
DcmDspRoutineSignalPos	0
DcmDspRoutineSignalType	BOOLEAN



5.2.2.1.82. DcmDspRoutineStopOutSignal_1

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	32
DcmDspRoutineSignalPos	1
DcmDspRoutineSignalType	UINT32

5.2.2.1.83. DcmDspStartRoutineIn

Containers included	
Container name	Container definition
DcmDspStartRoutineInSignal_0	DcmDspStartRoutineInSignal
DcmDspStartRoutineInSignal_1	DcmDspStartRoutineInSignal

Parameters included

Parameter name	Value

5.2.2.1.84. DcmDspStartRoutineInSignal_0

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	8
DcmDspRoutineSignalPos	0
DcmDspRoutineSignalType	SINT8

5.2.2.1.85. DcmDspStartRoutineInSignal_1

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	8
DcmDspRoutineSignalPos	8



Parameters included

DcmDspRoutineSignalType	UINT8
---	-------

5.2.2.1.86. DcmDspStartRoutineOut

Containers included

Container name	Container definition
DcmDspStartRoutineOutSignal_0	DcmDspStartRoutineOutSignal
DcmDspStartRoutineOutSignal_1	DcmDspStartRoutineOutSignal

Parameters included

Parameter name	Value
----------------	-------

5.2.2.1.87. DcmDspStartRoutineOutSignal_0

Parameters included

Parameter name	Value
DcmDspRoutineSignalLength	16
DcmDspRoutineSignalPos	0
DcmDspRoutineSignalType	SINT16

5.2.2.1.88. DcmDspStartRoutineOutSignal_1

Parameters included

Parameter name	Value
DcmDspRoutineSignalLength	32
DcmDspRoutineSignalPos	16
DcmDspRoutineSignalType	SINT32

5.2.2.1.89. DcmDspSecurity

Containers included

Container name	Container definition
----------------	----------------------

**Containers included**

DCM_SECLEV_L1	DcmDspSecurityRow
DCM_SECLEV_L3	DcmDspSecurityRow

Parameters included

Parameter name	Value

5.2.2.1.90. DCM_SECLEV_L1**Parameters included**

Parameter name	Value
DcmDspSecurityADRSIZE	1
DcmDspSecurityUsePort	USE_SECURITY_ASYNC_FNC
DcmDspSecurityGetSeedFnc	SecurityGetSeedADR0
DcmDspSecurityCompareKeyFnc	SecurityCompareKey0
DcmDspSecurityDelayTime	0.06
DcmDspSecurityDelayTimeOnBoot	0.0
DcmDspSecurityKeySize	2
DcmDspSecurityLevel	1
DcmDspSecurityNumAttDelay	1
DcmDspSecuritySeedSize	3

5.2.2.1.91. DCM_SECLEV_L3**Parameters included**

Parameter name	Value
DcmDspSecurityADRSIZE	1
DcmDspSecurityUsePort	USE_SECURITY_SYNCH_FNC
DcmDspSecurityGetSeedFnc	SecurityGetSeedADR0
DcmDspSecurityCompareKeyFnc	SecurityCompareKey0
DcmDspSecurityDelayTime	0.06
DcmDspSecurityDelayTimeOnBoot	0.0
DcmDspSecurityKeySize	4



Parameters included

DcmDspSecurityLevel	5
DcmDspSecurityNumAttDelay	2
DcmDspSecuritySeedSize	4

5.2.2.1.92. DcmDspTestResultByObdmid

Parameters included

Parameter name	Value
DcmWriteMidBeforeTid	false

5.2.2.1.93. DcmDspMemory

Containers included

Container name	Container definition
DcmDspAddressAndLengthFormatIdentifier	DcmDspMemory DcmDspAddressAndLengthFormatIdentifier
DcmDspMemoryIdInfo_0	DcmDspMemoryIdInfo

Parameters included

Parameter name	Value
DcmDspUseMemoryId	false

5.2.2.1.94. DcmDspAddressAndLengthFormatIdentifier

Parameters included

Parameter name	Value

5.2.2.1.95. DcmDspMemoryIdInfo_0

Containers included

Container name	Container definition
DcmDspReadMemoryRangeInfo_0	DcmDspReadMemoryRangeInfo
DcmDspWriteMemoryRangeInfo_0	DcmDspWriteMemoryRangeInfo



Parameters included

Parameter name	Value
DcmDspMemoryIdValue	(DISABLED)

5.2.2.1.96. DcmDspReadMemoryRangeInfo_0

Parameters included

Parameter name	Value
DcmDspReadMemoryRangeHigh	0
DcmDspReadMemoryRangeLow	0

5.2.2.1.97. DcmDspWriteMemoryRangeInfo_0

Parameters included

Parameter name	Value
DcmDspWriteMemoryRangeHigh	0
DcmDspWriteMemoryRangeLow	0

5.2.2.1.98. DcmDsl

Containers included

Container name	Container definition
DcmDslBuffer_Common	DcmDslBuffer
DcmDslDiagResp	DcmDsl DcmDslDiagResp
DcmDslProtocol	DcmDsl DcmDslProtocol

Parameters included

Parameter name	Value

5.2.2.1.99. DcmDslBuffer_Common

Parameters included

Parameter name	Value



Parameters included

DcmDsIBufferSize	256
----------------------------------	-----

5.2.2.1.100. DcmDsIDiagResp

Parameters included

Parameter name	Value
DcmDsIDiagRespMaxNumRespPend	5
DcmDsIDiagRespOnSecondDeclinedRequest	false

5.2.2.1.101. DcmDsIProtocol

Containers included

Container name	Container definition
DcmDsIProtocolRow_0	DcmDsIProtocolRow

Parameters included

Parameter name	Value
----------------	-------

5.2.2.1.102. DcmDsIProtocolRow_0

Containers included

Container name	Container definition
DcmDsIConnection_0	DcmDsIConnection

Parameters included

Parameter name	Value
DcmDsIProtocolPriority	0
DcmSendRespPendOnTransToBoot	true
DcmDsIProtocolEndiannessConvEnabled	false
DcmDsIProtocolID	DCM_UDS_ON_CAN
DcmDsIProtocolsParallelExecutab	false
DcmDsIProtocolPreemptTimeout	0.01
DcmDsIProtocolTransType	(DISABLED)



Parameters included

DcmTimStrP2ServerAdjust	0.0
DcmTimStrP2StarServerAdjust	0.0

5.2.2.1.103. DcmDsIConnection_0

Containers included

Container name	Container definition
DcmDsIMainConnection	DcmDsIMainConnection
DcmDsIPeriodicTransmission	DcmDsIPeriodicTransmission
DcmDsIResponseOnEvent	DcmDsIResponseOnEvent

Parameters included

Parameter name	Value

5.2.2.1.104. DcmDsIMainConnection

Containers included

Container name	Container definition
DcmDsIProtocolRx_0	DcmDsIProtocolRx
DcmDsIProtocolRx_1	DcmDsIProtocolRx
DcmDsIProtocolTx	DcmDsIMainConnection DcmDsIProtocolTx

Parameters included

Parameter name	Value
DcmDsIProtocolRxTesterSourceAddr	0

5.2.2.1.105. DcmDsIProtocolRx_0

Parameters included

Parameter name	Value
DcmDsIProtocolRxPduld	0
DcmDsIProtocolRxAddrType	DCM_PHYSICAL_TYPE
DcmDsIProtocolRxChannelId	0



5.2.2.1.106. DcmDslProtocolRx_1

Parameters included	
Parameter name	Value
DcmDslProtocolRxPduld	1
DcmDslProtocolRxAddrType	DCM_FUNCTIONAL_TYPE
DcmDslProtocolRxChannelId	0

5.2.2.1.107. DcmDslProtocolTx

Parameters included	
Parameter name	Value
DcmDslTxConfirmationPduld	0

5.2.2.1.108. DcmDslPeriodicTransmission

Parameters included	
Parameter name	Value

5.2.2.1.109. DcmDslResponseOnEvent

Parameters included	
Parameter name	Value
DcmDslRoeTxConfirmationPduld	(DISABLED)

5.2.2.1.110. DcmDefensiveProgramming

Parameters included	
Parameter name	Value
DcmDefProgEnabled	true
DcmPrecondAssertEnabled	true
DcmPostcondAssertEnabled	true
DcmStaticAssertEnabled	true
DcmUnreachAssertEnabled	true

**Parameters included**

DcmInvariantAssertEnabled	true
---	------

5.2.2.1.111. DcmGeneral**Containers included**

Container name	Container definition
DcmServiceAPI	DcmGeneral DcmServiceAPI

Parameters included

Parameter name	Value
DcmDD DIDStorage	false
DcmDevErrorDetect	true
DcmDSI EnableSetActiveDiagnosis	false
DcmDynamicallyDefinedDIDInitPolicy	(DISABLED)
DcmDynamicallyDefinedPeriodicDIDInitPolicy	(DISABLED)
DcmNRC PolicyISOVersion	ISO14229_2013
DcmModeDeclarationSupport	false
DcmRequestManufacturerNotificationEnabled	false
DcmRequestSupplierNotificationEnabled	false
DcmRespondAllRequest	true
DcmTaskTime	0.01
DcmVersionInfoApi	true
DcmRteUsage	false
DcmRequestSeedAfterFailedCompareKey	false

5.2.2.1.112. DcmServiceAPI**Parameters included**

Parameter name	Value
DcmEnableASR40ServiceAPI	false
DcmEnableASR42ServiceAPI	false
DcmDefaultASRServiceAPI	AUTOSAR_40



5.2.2.1.113. DcmPageBufferCfg

Parameters included	
Parameter name	Value
DcmPagedBufferEnabled	false
DcmPagedBufferTimeout	0.05

5.2.2.1.114. DcmProcessingConditions

Parameters included	
Parameter name	Value

5.2.2.2. DcmRecConfigurationDflt

Containers included	
Container name	Container definition
DcmConfigSet_0	DcmConfigSet

Parameters included	
Parameter name	Value
IMPLEMENTATION_CONFIG_VARIANT	VariantPreCompile

5.2.2.2.1. DcmConfigSet_0

Containers included	
Container name	Container definition
DcmDsd	DcmDsd
DcmDsp	DcmDsp
DcmDsl	DcmDsl
DcmDefensiveProgramming	DcmDefensiveProgramming
DcmGeneral	DcmGeneral
DcmPageBufferCfg	DcmPageBufferCfg

**Containers included**

DcmProcessingConditions	DcmProcessingConditions
---	---

Parameters included

Parameter name	Value
----------------	-------

5.2.2.2.2. DcmDsd**Containers included**

Container name	Container definition
DcmDsdServiceTable_0	DcmDsdServiceTable

Parameters included

Parameter name	Value
----------------	-------

5.2.2.2.3. DcmDsdServiceTable_0**Containers included**

Container name	Container definition
TesterPresent	DcmDsdService
DiagnosticSessionControl	DcmDsdService
EcuReset	DcmDsdService
SecurityAccess	DcmDsdService
ReadDid	DcmDsdService
WriteDid	DcmDsdService
ReadDTCInformation	DcmDsdService
ClearDiagnosticInformation	DcmDsdService
CommunicationControl	DcmDsdService
RoutineControl	DcmDsdService
RequestDownload	DcmDsdService
RequestUpload	DcmDsdService
TransferData	DcmDsdService
RequestTransferExit	DcmDsdService
LinkControl	DcmDsdService

**Containers included**

ControlDTCSetting	DcmDsdService
ReadMemoryByAddress	DcmDsdService
WriteMemoryByAddress	DcmDsdService
InputOutputControlByIdentifier	DcmDsdService
ResponseOnEvent	DcmDsdService

Parameters included

Parameter name	Value
DcmDsdSidTabId	0

5.2.2.2.4. TesterPresent**Parameters included**

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	62
DcmDsdSidTabSubfuncAvail	true

5.2.2.2.5. DiagnosticSessionControl**Containers included**

Container name	Container definition
defaultSession	DcmDsdSubService
programmingSession	DcmDsdSubService
extendedDiagnosticSession	DcmDsdSubService
oem_BI_ProgrammingSession	DcmDsdSubService
sys_BI_ProgrammingSession	DcmDsdSubService

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false



Parameters included

DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	16
DcmDsdSidTabSubfuncAvail	true

5.2.2.2.6. defaultSession

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	1

5.2.2.2.7. programmingSession

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	2

5.2.2.2.8. extendedDiagnosticSession

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	3

5.2.2.2.9. oem_BI_ProgrammingSession

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)



Parameters included

DcmDsdSubServiceId	64
------------------------------------	----

5.2.2.2.10. sys_BI_ProgrammingSession

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	65

5.2.2.2.11. EcuReset

Containers included

Container name	Container definition
hardReset	DcmDsdSubService
keyOffOnReset	DcmDsdSubService
softReset	DcmDsdSubService
enableRapidPowerShutDown	DcmDsdSubService
disableRapidPowerShutDown	DcmDsdSubService

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	17
DcmDsdSidTabSubfuncAvail	true

5.2.2.2.12. hardReset

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	1



5.2.2.2.13. keyOffOnReset

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	2

5.2.2.2.14. softReset

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	3

5.2.2.2.15. enableRapidPowerShutDown

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	4

5.2.2.2.16. disableRapidPowerShutDown

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	5

5.2.2.2.17. SecurityAccess

Parameters included	
Parameter name	Value
DcmAsyncServiceExecution	false



Parameters included

DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	39
DcmDsdSidTabSubfuncAvail	true

5.2.2.2.18. ReadDid

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	34
DcmDsdSidTabSubfuncAvail	false

5.2.2.2.19. WriteDid

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	46
DcmDsdSidTabSubfuncAvail	false

5.2.2.2.20. ReadDTCInformation

Containers included

Container name	Container definition
reportNumberOfDTCByStatusMask	DcmDsdSubService
reportDTCByStatusMask	DcmDsdSubService



Containers included

reportDTCSnapshotRecordIdentification	DcmDsdSubService
reportDTCSnapshotRecordByDTCNumber	DcmDsdSubService
reportDTCExtendedDataRecordByDTCNumber	DcmDsdSubService
reportSupportedDTCs	DcmDsdSubService

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	25
DcmDsdSidTabSubfuncAvail	true

5.2.2.2.21. reportNumberOfDTCByStatusMask

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	1

5.2.2.2.22. reportDTCByStatusMask

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	2

5.2.2.2.23. reportDTCSnapshotRecordIdentification

Parameters included

Parameter name	Value



Parameters included

DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	3

5.2.2.2.24. reportDTCSnapshotRecordByDTCNumber

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	4

5.2.2.2.25. reportDTCExtendedDataRecordByDTCNumber

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	6

5.2.2.2.26. reportSupportedDTCs

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	10

5.2.2.2.27. ClearDiagnosticInformation

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)



Parameters included

DcmDsdSidTabServiceId	20
DcmDsdSidTabSubfuncAvail	false

5.2.2.28. CommunicationControl

Containers included

Container name	Container definition
enableRxAndTx	DcmDsdSubService
enableRxAndDisableTx	DcmDsdSubService
disableRxAndEnableTx	DcmDsdSubService
disableRxAndTx	DcmDsdSubService

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	40
DcmDsdSidTabSubfuncAvail	true

5.2.2.29. enableRxAndTx

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	0

5.2.2.30. enableRxAndDisableTx

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)



Parameters included

DcmDsdSubServiceId	1
------------------------------------	---

5.2.2.2.31. disableRxAndEnableTx

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	2

5.2.2.2.32. disableRxAndTx

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	3

5.2.2.2.33. RoutineControl

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	49
DcmDsdSidTabSubfuncAvail	true

5.2.2.2.34. RequestDownload

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true



Parameters included

DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	52
DcmDsdSidTabSubfuncAvail	false

5.2.2.2.35. RequestUpload

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	53
DcmDsdSidTabSubfuncAvail	false

5.2.2.2.36. TransferData

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	54
DcmDsdSidTabSubfuncAvail	false

5.2.2.2.37. RequestTransferExit

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)



Parameters included

DcmDsdSidTabServiceId	55
DcmDsdSidTabSubfuncAvail	false

5.2.2.38. LinkControl

Containers included

Container name	Container definition
verifyBaudrateTransitionWithFixedBaudrate	DcmDsdSubService
verifyBaudrateTransitionWithSpecificBaudrate	DcmDsdSubService
verifyBaudrateTransitionWithTransitionBaudrate	DcmDsdSubService

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	135
DcmDsdSidTabSubfuncAvail	true

5.2.2.39. verifyBaudrateTransitionWithFixedBaudrate

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	1

5.2.2.40. verifyBaudrateTransitionWithSpecificBaudrate

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	2



5.2.2.2.41. verifyBaudrateTransitionWithTransitionBaudrate

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	3

5.2.2.2.42. ControlDTCSetting

Containers included	
Container name	Container definition
DcmDsdSubService_0	DcmDsdSubService
DcmDsdSubService_1	DcmDsdSubService

Parameters included

Parameter name	Value
DcmDsdServiceUsed	true
DcmDsdSidTabServiceId	133
DcmDsdSidTabSubfuncAvail	true
DcmAsyncServiceExecution	false
DcmDsdSidTabFnc	(DISABLED)

5.2.2.2.43. DcmDsdSubService_0

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	1

5.2.2.2.44. DcmDsdSubService_1

Parameters included	
Parameter name	Value



Parameters included

DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	2

5.2.2.2.45. ReadMemoryByAddress

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceld	35
DcmDsdSidTabSubfuncAvail	false

5.2.2.2.46. WriteMemoryByAddress

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceld	61
DcmDsdSidTabSubfuncAvail	false

5.2.2.2.47. InputOutputControlByIdentifier

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceld	47



Parameters included

DcmDsdSidTabSubfuncAvail	false
--	-------

5.2.2.2.48. ResponseOnEvent

Containers included

Container name	Container definition
stopResponseOnEvent	DcmDsdSubService
onDTCStatusChange	DcmDsdSubService
onChangeOfDataIdentifier	DcmDsdSubService
reportActivatedEvents	DcmDsdSubService
startResponseOnEvent	DcmDsdSubService
clearResponseOnEvent	DcmDsdSubService

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceld	134
DcmDsdSidTabSubfuncAvail	true

5.2.2.2.49. stopResponseOnEvent

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	0

5.2.2.2.50. onDTCStatusChange

Parameters included

Parameter name	Value
----------------	-------



Parameters included

DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	1

5.2.2.2.51. onChangeOfDataIdentifier

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	3

5.2.2.2.52. reportActivatedEvents

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	4

5.2.2.2.53. startResponseOnEvent

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	5

5.2.2.2.54. clearResponseOnEvent

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	6



5.2.2.2.55. DcmDsp

Containers included	
Container name	Container definition
DcmDspComControl	DcmDsp DcmDspComControl
DcmDspControlDTCSetting	DcmDsp DcmDspControlDTCSetting
DcmDspPeriodicDidTransmission	DcmDsp DcmDspPeriodicDidTransmission
DcmDspPeriodicTransmission	DcmDsp DcmDspPeriodicTransmission
DcmDspReadGenericInformation	DcmDsp DcmDspReadGenericInformation
DcmDspSession	DcmDspSession
DcmDspData_0x100	DcmDspData
DcmDspData_0xF186	DcmDspData
DcmDspDataInfo_0x100	DcmDspDataInfo
DcmDspDataInfo_0xF186	DcmDspDataInfo
DcmDspDid_0x100	DcmDspDid
DcmDspDid_0xF186	DcmDspDid
DcmDspDidInfo_0x100	DcmDspDidInfo
DcmDspDidInfo_0xF186	DcmDspDidInfo
DcmDspRoe	DcmDsp DcmDspRoe
DcmDspRoutine_0	DcmDspRoutine
DcmDspRoutineInfo_0	DcmDspRoutineInfo
DcmDspSecurity	DcmDsp DcmDspSecurity
DcmDspTestResultByObdmid	DcmDsp DcmDspTestResultByObdmid
DcmDspMemory	DcmDsp DcmDspMemory

Parameters included	
Parameter name	Value
DcmDspDDIDCheckPerSourceDID	false
DcmDspGenerateOnlyNeededIOControlOperations	false
DcmDspMaxPeriodicDidToRead	1
DcmDspReturnControlToEcuTimeout	1
DcmDspSessionControlTiming	false
DcmDspMaxDidToRead	1



Parameters included

DcmDspPowerDownTime	10
DcmDspRoutineVariableLengthInBytes	false

5.2.2.2.56. DcmDspComControl

Containers included

Container name	Container definition
DcmDspComControlSetting	DcmDspComControl DcmDspComControlSetting

Parameters included

Parameter name	Value

5.2.2.2.57. DcmDspComControlSetting

Parameters included

Parameter name	Value

5.2.2.2.58. DcmDspControlIDTCSetting

Parameters included

Parameter name	Value
DcmSupportDTCSettingControlOptionRecord	false

5.2.2.2.59. DcmDspPeriodicDidTransmission

Parameters included

Parameter name	Value
DcmDspMaxPeriodicDidScheduler	(DISABLED)
DcmDspPeriodicDidSamplingBufferSize	(DISABLED)
DcmDspPeriodicDidTransmissionRingEntries	(DISABLED)
DcmDspPeriodicDD DID Max Length	(DISABLED)



5.2.2.2.60. DcmDspPeriodicTransmission

Parameters included	
Parameter name	Value
DcmDspPeriodicTransmissionSlowRate	(DISABLED)
DcmDspPeriodicTransmissionMediumRate	(DISABLED)
DcmDspPeriodicTransmissionFastRate	(DISABLED)

5.2.2.2.61. DcmDspReadGenericInformation

Parameters included	
Parameter name	Value
DcmDspMaxDemGetDiagnosticDataConsecutive-Executions	(DISABLED)

5.2.2.2.62. DcmDspSession

Containers included	
Container name	Container definition
DCM_DEFAULT_SESSION	DcmDspSessionRow
DCM_EXTENDED_DIAGNOSTIC_SESSION	DcmDspSessionRow
DCM_PROGRAMMING_SESSION	DcmDspSessionRow
DCM_ALL_SESSION_LEVEL	DcmDspSessionRow
DcmDspSessionRow_0	DcmDspSessionRow
DcmDspSessionRow_1	DcmDspSessionRow

Parameters included

Parameter name	Value

5.2.2.2.63. DCM_DEFAULT_SESSION

Parameters included	
Parameter name	Value

**Parameters included**

DcmDspSessionLevel	1
DcmDspSessionForBoot	DCM_NO_BOOT
DcmDspSessionP2ServerMax	0.05
DcmDspSessionP2StarServerMax	5.0

5.2.2.64. DCM_EXTENDED_DIAGNOSTIC_SESSION**Parameters included**

Parameter name	Value
DcmDspSessionForBoot	DCM_NO_BOOT
DcmDspSessionLevel	3
DcmDspSessionP2ServerMax	0.1
DcmDspSessionP2StarServerMax	2.0

5.2.2.65. DCM_PROGRAMMING_SESSION**Parameters included**

Parameter name	Value
DcmDspSessionForBoot	DCM_NO_BOOT
DcmDspSessionLevel	2
DcmDspSessionP2ServerMax	0.5
DcmDspSessionP2StarServerMax	7.0

5.2.2.66. DCM_ALL_SESSION_LEVEL**Parameters included**

Parameter name	Value
DcmDspSessionForBoot	DCM_NO_BOOT
DcmDspSessionLevel	255
DcmDspSessionP2ServerMax	0.5
DcmDspSessionP2StarServerMax	5.0



5.2.2.2.67. DcmDspSessionRow_0

Parameters included	
Parameter name	Value
DcmDspSessionForBoot	DCM_OEM_BOOT
DcmDspSessionLevel	65
DcmDspSessionP2ServerMax	0.5
DcmDspSessionP2StarServerMax	5.0

5.2.2.2.68. DcmDspSessionRow_1

Parameters included	
Parameter name	Value
DcmDspSessionForBoot	DCM_SYS_BOOT
DcmDspSessionLevel	64
DcmDspSessionP2ServerMax	0.5
DcmDspSessionP2StarServerMax	5.0

5.2.2.2.69. DcmDspData_0x100

Parameters included	
Parameter name	Value
DcmDspDataConditionCheckReadFncUsed	true
DcmDspDataConditionCheckReadFnc	ReadDIDDataCondition0
DcmDspDataEcuSignal	(DISABLED)
DcmDspDataFreezeCurrentStateFnc	(DISABLED)
DcmDspDataGetScalingInfoFnc	(DISABLED)
DcmDspDataReadDataLengthFnc	ReadDIDDataLength0
DcmDspDataReadEcuSignal	(DISABLED)
DcmDspDataReadFnc	ReadDIDData0
DcmDspDataResetToDefaultFnc	(DISABLED)
DcmDspDataReturnControlToEcuFnc	(DISABLED)
DcmDspDataShortTermAdjustmentFnc	(DISABLED)



Parameters included

DcmDspDataSize	5
DcmDspDataType	UINT8
DcmDspDataUsePort	USE_DATA_ASYNCH_FNC
DcmDspDataWriteFnc	WriteDIDData0

5.2.2.2.70. DcmDspData_0xF186

Parameters included

Parameter name	Value
DcmDspDataConditionCheckReadFncUsed	false
DcmDspDataConditionCheckReadFnc	(DISABLED)
DcmDspDataEcuSignal	(DISABLED)
DcmDspDataFreezeCurrentStateFnc	(DISABLED)
DcmDspDataGetScalingInfoFnc	(DISABLED)
DcmDspDataReadDataLengthFnc	(DISABLED)
DcmDspDataReadEcuSignal	(DISABLED)
DcmDspDataReadFnc	Dcm_Internalhalf186Readout
DcmDspDataResetToDefaultFnc	(DISABLED)
DcmDspDataReturnControlToEcuFnc	(DISABLED)
DcmDspDataShortTermAdjustmentFnc	(DISABLED)
DcmDspDataSize	8
DcmDspDataType	UINT8
DcmDspDataUsePort	USE_DATA_SYNCH_FNC
DcmDspDataWriteFnc	(DISABLED)

5.2.2.2.71. DcmDspDataInfo_0x100

Parameters included

Parameter name	Value
DcmDspDataFixedLength	true
DcmDspDataScalingInfoSize	(DISABLED)



5.2.2.2.72. DcmDspDataInfo_0xF186

Parameters included	
Parameter name	Value
DcmDspDataFixedLength	true
DcmDspDataScalingInfoSize	(DISABLED)

5.2.2.2.73. DcmDspDid_0x100

Containers included	
Container name	Container definition
DcmDspDidSignal_0	DcmDspDidSignal

Parameters included

Parameter name	Value
DcmDspDidIdentifier	256
DcmDspDidUsed	true

5.2.2.2.74. DcmDspDidSignal_0

Parameters included	
Parameter name	Value
DcmDspDidDataPos	2

5.2.2.2.75. DcmDspDid_0xF186

Containers included	
Container name	Container definition
DcmDspDidSignal_0XF186	DcmDspDidSignal

Parameters included

Parameter name	Value
DcmDspDidIdentifier	61830
DcmDspDidUsed	true



5.2.2.2.76. DcmDspDidSignal_0XF186

Parameters included	
Parameter name	Value
DcmDspDidDataPos	0

5.2.2.2.77. DcmDspDidInfo_0x100

Containers included	
Container name	Container definition
DcmDspDidAccess	DcmDspDidAccess

Parameters included

Parameter name	Value
DcmDspDDIDMaxElements	1
DcmDspDidDynamicallyDefined	false

5.2.2.2.78. DcmDspDidAccess

Containers included	
Container name	Container definition
DcmDspDidControl	DcmDspDidAccess DcmDspDidControl
DcmDspDidRead	DcmDspDidAccess DcmDspDidRead
DcmDspDidWrite	DcmDspDidAccess DcmDspDidWrite

Parameters included

Parameter name	Value

5.2.2.2.79. DcmDspDidControl

Parameters included	
Parameter name	Value
DcmDspDidFreezeCurrentState	(DISABLED)
DcmDspDidResetToDefault	(DISABLED)



Parameters included

DcmDspDidReturnControlToEcu	(DISABLED)
DcmDspDidShortTermAdjustment	(DISABLED)

5.2.2.2.80. DcmDspDidRead

Parameters included

Parameter name	Value

5.2.2.2.81. DcmDspDidWrite

Parameters included

Parameter name	Value

5.2.2.2.82. DcmDspDidInfo_0xF186

Containers included

Container name	Container definition
DcmDspDidAccess	DcmDspDidAccess

Parameters included

Parameter name	Value
DcmDspDidDynamicallyDefined	false
DcmDspDD DIDMaxElements	1

5.2.2.2.83. DcmDspDidAccess

Containers included

Container name	Container definition
DcmDspDidControl	DcmDspDidAccess DcmDspDidControl
DcmDspDidRead	DcmDspDidAccess DcmDspDidRead
DcmDspDidWrite	DcmDspDidAccess DcmDspDidWrite

**Parameters included**

Parameter name	Value
----------------	-------

5.2.2.2.84. DcmDspDidControl**Parameters included**

Parameter name	Value
DcmDspDidFreezeCurrentState	(DISABLED)
DcmDspDidResetToDefault	(DISABLED)
DcmDspDidReturnControlToEcu	(DISABLED)
DcmDspDidShortTermAdjustment	(DISABLED)

5.2.2.2.85. DcmDspDidRead**Parameters included**

Parameter name	Value
----------------	-------

5.2.2.2.86. DcmDspDidWrite**Parameters included**

Parameter name	Value
----------------	-------

5.2.2.2.87. DcmDspRoe**Containers included**

Container name	Container definition
DcmDspRoeEvent_0	DcmDspRoeEvent
DcmDspRoeEventWindowTime_0	DcmDspRoeEventWindowTime

Parameters included

Parameter name	Value
DcmDspRoeInterMessageTime	0.0



5.2.2.2.88. DcmDspRoeEvent_0

Parameters included	
Parameter name	Value
DcmDspRoeEventId	0
DcmDspRoeInitialEventStatus	DCM_ROE_CLEARED

5.2.2.2.89. DcmDspRoeEventProperties

Containers included	
Container name	Container definition
DcmDspRoeOnChangeOfDataIdentifier	DcmDspRoeEventProperties DcmDspRoeOnChangeOfDataIdentifier
DcmDspRoeOnDTCStatusChange	DcmDspRoeEventProperties DcmDspRoeOnDTCStatusChange

Parameters included

Parameter name	Value

5.2.2.2.90. DcmDspRoeOnChangeOfDataIdentifier

Parameters included	
Parameter name	Value

5.2.2.2.91. DcmDspRoeOnDTCStatusChange

Parameters included	
Parameter name	Value
DcmDspRoeEventPreconfiguredDTCMask	0

5.2.2.2.92. DcmDspRoeEventWindowTime_0

Parameters included	
Parameter name	Value

**Parameters included**

DcmDspRoeEventWindowTime	DCM_ROE_EVENT_WINDOW_INFINITE
DcmDspRoeStorageState	false

5.2.2.2.93. DcmDspRoutine_0**Parameters included**

Parameter name	Value
DcmDspRequestResultsRoutineFnc	Routine0_RequestResult
DcmDspRoutineFixedLength	true
DcmDspRoutineIdentifier	65530
DcmDspRoutineUsePort	false
DcmDspRoutineUsed	true
DcmDspRoutineUsePortASRVersion	(DISABLED)
DcmDspStartRoutineFnc	Routine0_Start
DcmDspStopRoutineFnc	Routine0_Stop
DcmDspRequestResultsRoutineSupported	true
DcmDspStopRoutineSupported	true

5.2.2.2.94. DcmDspRoutineInfo_0**Containers included**

Container name	Container definition
DcmDspRoutineAuthorization	DcmDspRoutineAuthorization
DcmDspRoutineRequestResIn	DcmDspRoutineRequestResIn
DcmDspRoutineRequestResOut	DcmDspRoutineRequestResOut
DcmDspRoutineStopIn	DcmDspRoutineStopIn
DcmDspRoutineStopOut	DcmDspRoutineStopOut
DcmDspStartRoutineIn	DcmDspStartRoutineIn
DcmDspStartRoutineOut	DcmDspStartRoutineOut

Parameters included

Parameter name	Value



5.2.2.2.95. DcmDspRoutineAuthorization

Parameters included	
Parameter name	Value

5.2.2.2.96. DcmDspRoutineRequestResIn

Parameters included	
Parameter name	Value

5.2.2.2.97. DcmDspRoutineRequestResOut

Containers included	
Container name	Container definition
DcmDspRoutineRequestResOutSignal_0	DcmDspRoutineRequestResOutSignal
DcmDspRoutineRequestResOutSignal_1	DcmDspRoutineRequestResOutSignal

Parameters included	
Parameter name	Value

5.2.2.2.98. DcmDspRoutineRequestResOutSignal_0

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	8
DcmDspRoutineSignalPos	0
DcmDspRoutineSignalType	UINT8

5.2.2.2.99. DcmDspRoutineRequestResOutSignal_1

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	1
DcmDspRoutineSignalPos	8



Parameters included

DcmDspRoutineSignalType	BOOLEAN
---	---------

5.2.2.2.100. DcmDspRoutineStopIn

Containers included

Container name	Container definition
DcmDspRoutineStopInSignal_0	DcmDspRoutineStopInSignal
DcmDspRoutineStopInSignal_1	DcmDspRoutineStopInSignal

Parameters included

Parameter name	Value
----------------	-------

5.2.2.2.101. DcmDspRoutineStopInSignal_0

Parameters included

Parameter name	Value
DcmDspRoutineSignalLength	16
DcmDspRoutineSignalPos	0
DcmDspRoutineSignalType	UINT16

5.2.2.2.102. DcmDspRoutineStopInSignal_1

Parameters included

Parameter name	Value
DcmDspRoutineSignalLength	8
DcmDspRoutineSignalPos	16
DcmDspRoutineSignalType	UINT8

5.2.2.2.103. DcmDspRoutineStopOut

Containers included

Container name	Container definition
----------------	----------------------

**Containers included**

DcmDspRoutineStopOutSignal_0	DcmDspRoutineStopOutSignal
DcmDspRoutineStopOutSignal_1	DcmDspRoutineStopOutSignal

Parameters included

Parameter name	Value

5.2.2.2.104. DcmDspRoutineStopOutSignal_0**Parameters included**

Parameter name	Value
DcmDspRoutineSignalLength	1
DcmDspRoutineSignalPos	0
DcmDspRoutineSignalType	BOOLEAN

5.2.2.2.105. DcmDspRoutineStopOutSignal_1**Parameters included**

Parameter name	Value
DcmDspRoutineSignalLength	32
DcmDspRoutineSignalPos	1
DcmDspRoutineSignalType	UINT32

5.2.2.2.106. DcmDspStartRoutineIn**Containers included**

Container name	Container definition
DcmDspStartRoutineInSignal_0	DcmDspStartRoutineInSignal
DcmDspStartRoutineInSignal_1	DcmDspStartRoutineInSignal

Parameters included

Parameter name	Value



5.2.2.2.107. DcmDspStartRoutineInSignal_0

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	8
DcmDspRoutineSignalPos	0
DcmDspRoutineSignalType	SINT8

5.2.2.2.108. DcmDspStartRoutineInSignal_1

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	8
DcmDspRoutineSignalPos	8
DcmDspRoutineSignalType	UINT8

5.2.2.2.109. DcmDspStartRoutineOut

Containers included	
Container name	Container definition
DcmDspStartRoutineOutSignal_0	DcmDspStartRoutineOutSignal
DcmDspStartRoutineOutSignal_1	DcmDspStartRoutineOutSignal

Parameters included

Parameter name	Value
----------------	-------

5.2.2.2.110. DcmDspStartRoutineOutSignal_0

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	16
DcmDspRoutineSignalPos	0

**Parameters included**

DcmDspRoutineSignalType	SINT16
---	--------

5.2.2.2.111. DcmDspStartRoutineOutSignal_1**Parameters included**

Parameter name	Value
DcmDspRoutineSignalLength	32
DcmDspRoutineSignalPos	16
DcmDspRoutineSignalType	SINT32

5.2.2.2.112. DcmDspSecurity**Containers included**

Container name	Container definition
DCM_SEC_LEV_L1	DcmDspSecurityRow
DCM_SEC_LEV_L3	DcmDspSecurityRow

Parameters included

Parameter name	Value
----------------	-------

5.2.2.2.113. DCM_SEC_LEV_L1**Parameters included**

Parameter name	Value
DcmDspSecurityADRSize	1
DcmDspSecurityUsePort	USE_SECURITY_ASYNC_FNC
DcmDspSecurityGetSeedFnc	SecurityGetSeedADR0
DcmDspSecurityCompareKeyFnc	SecurityCompareKey0
DcmDspSecurityDelayTime	0.06
DcmDspSecurityDelayTimeOnBoot	0.0
DcmDspSecurityKeySize	2
DcmDspSecurityLevel	1

**Parameters included**

DcmDspSecurityNumAttDelay	1
DcmDspSecuritySeedSize	3

5.2.2.2.114. DCM_SEC_LEV_L3**Parameters included**

Parameter name	Value
DcmDspSecurityADRSIZE	1
DcmDspSecurityUsePort	USE_SECURITY_SYNCH_FNC
DcmDspSecurityGetSeedFnc	SecurityGetSeedADR0
DcmDspSecurityCompareKeyFnc	SecurityCompareKey0
DcmDspSecurityDelayTime	0.06
DcmDspSecurityDelayTimeOnBoot	0.0
DcmDspSecurityKeySize	4
DcmDspSecurityLevel	5
DcmDspSecurityNumAttDelay	2
DcmDspSecuritySeedSize	4

5.2.2.2.115. DcmDspTestResultByObdmid**Parameters included**

Parameter name	Value
DcmWriteMidBeforeTid	false

5.2.2.2.116. DcmDspMemory**Containers included**

Container name	Container definition
DcmDspAddressAndLengthFormatIdentifier	DcmDspMemory DcmDspAddressAndLengthFormatIdentifier
DcmDspMemoryIdInfo_0	DcmDspMemoryIdInfo



Parameters included

Parameter name	Value
DcmDspUseMemoryId	false

5.2.2.2.117. DcmDspAddressAndLengthFormatIdentifier

Parameters included

Parameter name	Value
----------------	-------

5.2.2.2.118. DcmDspMemoryIdInfo_0

Containers included

Container name	Container definition
DcmDspReadMemoryRangeInfo_0	DcmDspReadMemoryRangeInfo
DcmDspWriteMemoryRangeInfo_0	DcmDspWriteMemoryRangeInfo

Parameters included

Parameter name	Value
DcmDspMemoryIdValue	(DISABLED)

5.2.2.2.119. DcmDspReadMemoryRangeInfo_0

Parameters included

Parameter name	Value
DcmDspReadMemoryRangeHigh	0
DcmDspReadMemoryRangeLow	0

5.2.2.2.120. DcmDspWriteMemoryRangeInfo_0

Parameters included

Parameter name	Value
DcmDspWriteMemoryRangeHigh	0

**Parameters included**

DcmDspWriteMemoryRangeLow	0
---	---

5.2.2.2.121. DcmDsl**Containers included**

Container name	Container definition
DcmDslBuffer_Common	DcmDslBuffer
DcmDslBuffer_Rx	DcmDslBuffer
DcmDslBuffer_Tx	DcmDslBuffer
DcmDslDiagResp	DcmDsl DcmDslDiagResp
DcmDslProtocol	DcmDsl DcmDslProtocol

Parameters included

Parameter name	Value
----------------	-------

5.2.2.2.122. DcmDslBuffer_Common**Parameters included**

Parameter name	Value
DcmDslBufferSize	256

5.2.2.2.123. DcmDslBuffer_Rx**Parameters included**

Parameter name	Value
DcmDslBufferSize	8

5.2.2.2.124. DcmDslBuffer_Tx**Parameters included**

Parameter name	Value
DcmDslBufferSize	8



5.2.2.2.125. DcmDsIDiagResp

Parameters included	
Parameter name	Value
DcmDsIDiagRespMaxNumRespPend	5
DcmDsIDiagRespOnSecondDeclinedRequest	false

5.2.2.2.126. DcmDsIProtocol

Containers included	
Container name	Container definition
DcmDsIProtocolRow_0	DcmDsIProtocolRow
DcmDsIProtocolRow_1	DcmDsIProtocolRow

Parameters included

Parameter name	Value
----------------	-------

5.2.2.2.127. DcmDsIProtocolRow_0

Containers included	
Container name	Container definition
DcmDsIConnection_0	DcmDsIConnection
DcmDsIConnection_1	DcmDsIConnection
DcmDsIConnection_2	DcmDsIConnection

Parameters included

Parameter name	Value
DcmDsIProtocolPriority	0
DcmSendRespPendOnTransToBoot	true
DcmDsIProtocolEndiannessConvEnabled	false
DcmDsIProtocolID	DCM_UDS_ON_CAN
DcmDsIProtocolsParallelExecutab	false
DcmDsIProtocolPreemptTimeout	0.01
DcmDsIProtocolTransType	(DISABLED)



Parameters included

DcmTimStrP2ServerAdjust	0.0
DcmTimStrP2StarServerAdjust	0.0

5.2.2.128. DcmDsIConnection_0

Containers included

Container name	Container definition
DcmDsIMainConnection	DcmDsIMainConnection
DcmDsIPeriodicTransmission	DcmDsIPeriodicTransmission
DcmDsIResponseOnEvent	DcmDsIResponseOnEvent

Parameters included

Parameter name	Value

5.2.2.129. DcmDsIMainConnection

Containers included

Container name	Container definition
DcmDsIProtocolRx_0	DcmDsIProtocolRx
DcmDsIProtocolRx_1	DcmDsIProtocolRx
DcmDsIProtocolTx	DcmDsIMainConnection DcmDsIProtocolTx

Parameters included

Parameter name	Value
DcmDsIProtocolRxTesterSourceAddr	0

5.2.2.130. DcmDsIProtocolRx_0

Parameters included

Parameter name	Value
DcmDsIProtocolRxPduld	0
DcmDsIProtocolRxAddrType	DCM_PHYSICAL_TYPE
DcmDsIProtocolRxChannelId	0



5.2.2.2.131. DcmDsIProtocolRx_1

Parameters included	
Parameter name	Value
DcmDsIProtocolRxPduld	1
DcmDsIProtocolRxAddrType	DCM_FUNCTIONAL_TYPE
DcmDsIProtocolRxChannelId	0

5.2.2.2.132. DcmDsIProtocolTx

Parameters included	
Parameter name	Value
DcmDsITxConfirmationPduld	0

5.2.2.2.133. DcmDsIPeriodicTransmission

Parameters included	
Parameter name	Value

5.2.2.2.134. DcmDsIResponseOnEvent

Parameters included	
Parameter name	Value
DcmDsIRoeTxConfirmationPduld	(DISABLED)

5.2.2.2.135. DcmDsIConnection_1

Containers included	
Container name	Container definition
DcmDsIMainConnection	DcmDsIMainConnection
DcmDsIPeriodicTransmission	DcmDsIPeriodicTransmission
DcmDsIResponseOnEvent	DcmDsIResponseOnEvent



Parameters included

Parameter name	Value

5.2.2.2.136. DcmDsIMainConnection

Containers included

Container name	Container definition
DcmDsIProtocolTx	DcmDsIMainConnection DcmDsIProtocolTx

Parameters included

Parameter name	Value
DcmDsIProtocolRxTesterSourceAddr	0

5.2.2.2.137. DcmDsIProtocolTx

Parameters included

Parameter name	Value
DcmDsITxConfirmationPduld	0

5.2.2.2.138. DcmDsIPeriodicTransmission

Containers included

Container name	Container definition
DcmDsIPeriodicConnection	DcmDsIPeriodicConnection

Parameters included

Parameter name	Value

5.2.2.2.139. DcmDsIPeriodicConnection

Parameters included

Parameter name	Value



Parameters included

DcmDsIPeriodicTxConfirmationPduld	1
---	---

5.2.2.2.140. DcmDsIResponseOnEvent

Parameters included

Parameter name	Value
DcmDsIRoeTxConfirmationPduld	(DISABLED)

5.2.2.2.141. DcmDsIConnection_2

Containers included

Container name	Container definition
DcmDsIMainConnection	DcmDsIMainConnection
DcmDsIPeriodicTransmission	DcmDsIPeriodicTransmission
DcmDsIResponseOnEvent	DcmDsIResponseOnEvent

Parameters included

Parameter name	Value
----------------	-------

5.2.2.2.142. DcmDsIMainConnection

Containers included

Container name	Container definition
DcmDsIProtocolTx	DcmDsIMainConnection DcmDsIProtocolTx

Parameters included

Parameter name	Value
DcmDsIProtocolRxTesterSourceAddr	0

5.2.2.2.143. DcmDsIProtocolTx

Parameters included

Parameter name	Value
----------------	-------



Parameters included

DcmDslTxConfirmationPduld	0
---	---

5.2.2.2.144. DcmDslPeriodicTransmission

Parameters included

Parameter name	Value
----------------	-------

5.2.2.2.145. DcmDslResponseOnEvent

Parameters included

Parameter name	Value
DcmDslRoeTxConfirmationPduld	2

5.2.2.2.146. DcmDslProtocolRow_1

Containers included

Container name	Container definition
DcmDslConnection_3	DcmDslConnection
DcmDslConnection_4	DcmDslConnection
DcmDslConnection_5	DcmDslConnection

Parameters included

Parameter name	Value
DcmDslProtocolPriority	1
DcmSendRespPendOnTransToBoot	true
DcmDslProtocolEndiannessConvEnabled	false
DcmDslProtocolID	DCM_UDS_ON_FLEXRAY
DcmDslProtocolsParallelExecutab	false
DcmDslProtocolPreemptTimeout	0.01
DcmDslProtocolTransType	(DISABLED)
DcmTimStrP2ServerAdjust	0.0
DcmTimStrP2StarServerAdjust	0.0



5.2.2.2.147. DcmDslConnection_3

Containers included	
Container name	Container definition
DcmDslMainConnection	DcmDslMainConnection
DcmDslPeriodicTransmission	DcmDslPeriodicTransmission
DcmDslResponseOnEvent	DcmDslResponseOnEvent

Parameters included	
Parameter name	Value

5.2.2.2.148. DcmDslMainConnection

Containers included	
Container name	Container definition
DcmDslProtocolRx_0	DcmDslProtocolRx
DcmDslProtocolRx_1	DcmDslProtocolRx
DcmDslProtocolTx	DcmDslMainConnection DcmDslProtocolTx

Parameters included	
Parameter name	Value
DcmDslProtocolRxTesterSourceAddr	0

5.2.2.2.149. DcmDslProtocolRx_0

Parameters included	
Parameter name	Value
DcmDslProtocolRxPduld	2
DcmDslProtocolRxAddrType	DCM_PHYSICAL_TYPE
DcmDslProtocolRxChannelId	0

5.2.2.2.150. DcmDslProtocolRx_1

Parameters included	
Parameter name	Value

**Parameters included**

DcmDslProtocolRxPduld	3
DcmDslProtocolRxAddrType	DCM_FUNCTIONAL_TYPE
DcmDslProtocolRxChannelId	0

5.2.2.2.151. DcmDslProtocolTx**Parameters included**

Parameter name	Value
DcmDslTxConfirmationPduld	3

5.2.2.2.152. DcmDslPeriodicTransmission**Parameters included**

Parameter name	Value

5.2.2.2.153. DcmDslResponseOnEvent**Parameters included**

Parameter name	Value
DcmDslRoeTxConfirmationPduld	(DISABLED)

5.2.2.2.154. DcmDslConnection_4**Containers included**

Container name	Container definition
DcmDslMainConnection	DcmDslMainConnection
DcmDslPeriodicTransmission	DcmDslPeriodicTransmission
DcmDslResponseOnEvent	DcmDslResponseOnEvent

Parameters included

Parameter name	Value



5.2.2.2.155. DcmDsIMainConnection

Containers included	
Container name	Container definition
DcmDsIProtocolTx	DcmDsIMainConnection DcmDsIProtocolTx

Parameters included

Parameter name	Value
DcmDsIProtocolRxTesterSourceAddr	0

5.2.2.2.156. DcmDsIProtocolTx

Parameters included	
Parameter name	Value
DcmDsITxConfirmationPduld	0

5.2.2.2.157. DcmDsIPeriodicTransmission

Containers included	
Container name	Container definition
DcmDsIPeriodicConnection_0	DcmDsIPeriodicConnection
Parameters included	
Parameter name	Value

5.2.2.2.158. DcmDsIPeriodicConnection_0

Parameters included	
Parameter name	Value
DcmDsIPeriodicTxConfirmationPduld	4

5.2.2.2.159. DcmDsIResponseOnEvent

Parameters included	
Parameter name	Value



Parameters included

DcmDslRoeTxConfirmationPduld	(DISABLED)
--	------------

5.2.2.2.160. DcmDslConnection_5

Containers included

Container name	Container definition
DcmDslMainConnection	DcmDslMainConnection
DcmDslPeriodicTransmission	DcmDslPeriodicTransmission
DcmDslResponseOnEvent	DcmDslResponseOnEvent

Parameters included

Parameter name	Value
----------------	-------

5.2.2.2.161. DcmDslMainConnection

Containers included

Container name	Container definition
DcmDslProtocolTx	DcmDslMainConnection DcmDslProtocolTx

Parameters included

Parameter name	Value
DcmDslProtocolRxTesterSourceAddr	0

5.2.2.2.162. DcmDslProtocolTx

Parameters included

Parameter name	Value
DcmDslTxConfirmationPduld	0

5.2.2.2.163. DcmDslPeriodicTransmission

Parameters included

Parameter name	Value
----------------	-------



5.2.2.2.164. DcmDslResponseOnEvent

Parameters included	
Parameter name	Value
DcmDslRoeTxConfirmationPduld	5

5.2.2.2.165. DcmDefensiveProgramming

Parameters included	
Parameter name	Value
DcmDefProgEnabled	true
DcmPrecondAssertEnabled	true
DcmPostcondAssertEnabled	true
DcmStaticAssertEnabled	true
DcmUnreachAssertEnabled	true
DcmInvariantAssertEnabled	true

5.2.2.2.166. DcmGeneral

Containers included	
Container name	Container definition
DcmServiceAPI	DcmGeneral DcmServiceAPI

Parameters included	
Parameter name	Value
DcmDD DIDStorage	false
DcmDevErrorDetect	true
DcmDslEnableSetActiveDiagnosis	false
DcmDynamicallyDefinedDIDInitPolicy	(DISABLED)
DcmDynamicallyDefinedPeriodicDIDInitPolicy	(DISABLED)
DcmNRCPolicyISOVersion	ISO14229_2013
DcmModeDeclarationSupport	false
DcmRequestManufacturerNotificationEnabled	true
DcmRequestSupplierNotificationEnabled	true

**Parameters included**

DcmRespondAllRequest	true
DcmTaskTime	0.01
DcmVersionInfoApi	true
DcmRteUsage	false
DcmRequestSeedAfterFailedCompareKey	false

5.2.2.167. DcmServiceAPI**Parameters included**

Parameter name	Value
DcmEnableASR40ServiceAPI	false
DcmEnableASR42ServiceAPI	false
DcmDefaultASRServiceAPI	AUTOSAR_40

5.2.2.168. DcmPageBufferCfg**Parameters included**

Parameter name	Value
DcmPagedBufferEnabled	true
DcmPagedBufferTimeout	0.05

5.2.2.169. DcmProcessingConditions**Parameters included**

Parameter name	Value

5.2.2.3. DcmRecConfigurationWithRte**Containers included**

Container name	Container definition
DcmConfigSet_0	DcmConfigSet

**Parameters included**

Parameter name	Value
IMPLEMENTATION_CONFIG_VARIANT	VariantPreCompile

5.2.2.3.1. DcmConfigSet_0**Containers included**

Container name	Container definition
DcmDsd	DcmDsd
DcmDsp	DcmDsp
DcmDsl	DcmDsl
DcmDefensiveProgramming	DcmDefensiveProgramming
DcmGeneral	DcmGeneral
DcmPageBufferCfg	DcmPageBufferCfg
DcmProcessingConditions	DcmProcessingConditions

Parameters included

Parameter name	Value

5.2.2.3.2. DcmDsd**Containers included**

Container name	Container definition
DcmDsdServiceTable_0	DcmDsdServiceTable

Parameters included

Parameter name	Value

5.2.2.3.3. DcmDsdServiceTable_0**Containers included**

Container name	Container definition
TesterPresent	DcmDsdService

**Containers included**

DiagnosticSessionControl	DcmDsdService
EcuReset	DcmDsdService
SecurityAccess	DcmDsdService
ReadDid	DcmDsdService
WriteDid	DcmDsdService
ReadDTCInformation	DcmDsdService
ClearDiagnosticInformation	DcmDsdService
CommunicationControl	DcmDsdService
RoutineControl	DcmDsdService
RequestDownload	DcmDsdService
RequestUpload	DcmDsdService
TransferData	DcmDsdService
RequestTransferExit	DcmDsdService
LinkControl	DcmDsdService
ControlDTCSetting	DcmDsdService
ReadMemoryByAddress	DcmDsdService
WriteMemoryByAddress	DcmDsdService
InputOutputControlByIdentifier	DcmDsdService
ResponseOnEvent	DcmDsdService

Parameters included

Parameter name	Value
DcmDsdSidTabId	0

5.2.2.3.4. TesterPresent**Parameters included**

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	62



Parameters included

DcmDsdSidTabSubfuncAvail	true
--	------

5.2.2.3.5. DiagnosticSessionControl

Containers included

Container name	Container definition
defaultSession	DcmDsdSubService
programmingSession	DcmDsdSubService
extendedDiagnosticSession	DcmDsdSubService
oem_BI_ProgrammingSession	DcmDsdSubService
sys_BI_ProgrammingSession	DcmDsdSubService

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	16
DcmDsdSidTabSubfuncAvail	true

5.2.2.3.6. defaultSession

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	1

5.2.2.3.7. programmingSession

Parameters included

Parameter name	Value



Parameters included

DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	2

5.2.2.3.8. extendedDiagnosticSession

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	3

5.2.2.3.9. oem_BI_ProgrammingSession

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	64

5.2.2.3.10. sys_BI_ProgrammingSession

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	65

5.2.2.3.11. EcuReset

Containers included

Container name	Container definition
hardReset	DcmDsdSubService
keyOffOnReset	DcmDsdSubService



Containers included

softReset	DcmDsdSubService
enableRapidPowerShutDown	DcmDsdSubService
disableRapidPowerShutDown	DcmDsdSubService

Parameters included

Parameter name	Value
DcmDsdServiceUsed	true
DcmDsdSidTabServiceld	17
DcmDsdSidTabSubfuncAvail	true
DcmAsyncServiceExecution	false
DcmDsdSidTabFnc	(DISABLED)

5.2.2.3.12. hardReset

Parameters included

Parameter name	Value
DcmDsdSubServiceld	1
DcmDsdSubServiceFnc	(DISABLED)

5.2.2.3.13. keyOffOnReset

Parameters included

Parameter name	Value
DcmDsdSubServiceld	2
DcmDsdSubServiceFnc	(DISABLED)

5.2.2.3.14. softReset

Parameters included

Parameter name	Value
DcmDsdSubServiceld	3



Parameters included

DcmDsdSubServiceFnc	(DISABLED)
-------------------------------------	------------

5.2.2.3.15. enableRapidPowerShutDown

Parameters included

Parameter name	Value
DcmDsdSubServiceld	4
DcmDsdSubServiceFnc	(DISABLED)

5.2.2.3.16. disableRapidPowerShutDown

Parameters included

Parameter name	Value
DcmDsdSubServiceld	5
DcmDsdSubServiceFnc	(DISABLED)

5.2.2.3.17. SecurityAccess

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceld	39
DcmDsdSidTabSubfuncAvail	true

5.2.2.3.18. ReadDid

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true



Parameters included

DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	34
DcmDsdSidTabSubfuncAvail	false

5.2.2.3.19. WriteDid

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	46
DcmDsdSidTabSubfuncAvail	false

5.2.2.3.20. ReadDTCInformation

Containers included

Container name	Container definition
reportNumberOfDTCByStatusMask	DcmDsdSubService
reportDTCByStatusMask	DcmDsdSubService
reportDTCSnapshotRecordIdentification	DcmDsdSubService
reportDTCSnapshotRecordByDTCNumber	DcmDsdSubService
reportDTCExtendedDataRecordByDTCNumber	DcmDsdSubService
reportSupportedDTCs	DcmDsdSubService

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	25
DcmDsdSidTabSubfuncAvail	true



5.2.2.3.21. reportNumberOfDTCByStatusMask

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	1

5.2.2.3.22. reportDTCByStatusMask

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	2

5.2.2.3.23. reportDTCSnapshotRecordIdentification

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	3

5.2.2.3.24. reportDTCSnapshotRecordByDTCNumber

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	4

5.2.2.3.25. reportDTCExtendedDataRecordByDTCNumber

Parameters included	
Parameter name	Value



Parameters included

DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	6

5.2.2.3.26. reportSupportedDTCs

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	10

5.2.2.3.27. ClearDiagnosticInformation

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceld	20
DcmDsdSidTabSubfuncAvail	false

5.2.2.3.28. CommunicationControl

Containers included

Container name	Container definition
enableRxAndTx	DcmDsdSubService
enableRxAndDisableTx	DcmDsdSubService
disableRxAndEnableTx	DcmDsdSubService
disableRxAndTx	DcmDsdSubService

Parameters included

Parameter name	Value



Parameters included

DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	40
DcmDsdSidTabSubfuncAvail	true

5.2.2.3.29. enableRxAndTx

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	0

5.2.2.3.30. enableRxAndDisableTx

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	1

5.2.2.3.31. disableRxAndEnableTx

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	2

5.2.2.3.32. disableRxAndTx

Parameters included

Parameter name	Value



Parameters included

DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	3

5.2.2.3.33. RoutineControl

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceld	49
DcmDsdSidTabSubfuncAvail	true

5.2.2.3.34. RequestDownload

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceld	52
DcmDsdSidTabSubfuncAvail	false

5.2.2.3.35. RequestUpload

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceld	53



Parameters included

DcmDsdSidTabSubfuncAvail	false
--	-------

5.2.2.3.36. TransferData

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	54
DcmDsdSidTabSubfuncAvail	false

5.2.2.3.37. RequestTransferExit

Parameters included

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	55
DcmDsdSidTabSubfuncAvail	false

5.2.2.3.38. LinkControl

Containers included

Container name	Container definition
verifyBaudrateTransitionWithFixedBaudrate	DcmDsdSubService
verifyBaudrateTransitionWithSpecificBaudrate	DcmDsdSubService
verifyBaudrateTransitionWithTransitionBaudrate	DcmDsdSubService

Parameters included

Parameter name	Value
----------------	-------



Parameters included

DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	135
DcmDsdSidTabSubfuncAvail	true

5.2.2.3.39. verifyBaudrateTransitionWithFixedBaudrate

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	1

5.2.2.3.40. verifyBaudrateTransitionWithSpecificBaudrate

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	2

5.2.2.3.41. verifyBaudrateTransitionWithTransitionBaudrate

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	3

5.2.2.3.42. ControlDTCSetting

Containers included

Container name	Container definition
DcmDsdSubService_0	DcmDsdSubService

**Containers included**

DcmDsdSubService_1	DcmDsdSubService
------------------------------------	----------------------------------

Parameters included

Parameter name	Value
DcmDsdServiceUsed	true
DcmDsdSidTabServiceld	133
DcmDsdSidTabSubfuncAvail	true
DcmAsyncServiceExecution	false
DcmDsdSidTabFnc	(DISABLED)

5.2.2.3.43. DcmDsdSubService_0**Parameters included**

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	1

5.2.2.3.44. DcmDsdSubService_1**Parameters included**

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	2

5.2.2.3.45. ReadMemoryByAddress**Parameters included**

Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceld	35
DcmDsdSidTabSubfuncAvail	false



5.2.2.3.46. WriteMemoryByAddress

Parameters included	
Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	61
DcmDsdSidTabSubfuncAvail	false

5.2.2.3.47. InputOutputControlByIdentifier

Parameters included	
Parameter name	Value
DcmAsyncServiceExecution	false
DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	47
DcmDsdSidTabSubfuncAvail	false

5.2.2.3.48. ResponseOnEvent

Containers included	
Container name	Container definition
stopResponseOnEvent	DcmDsdSubService
onDTCStatusChange	DcmDsdSubService
onChangeOfDataIdentifier	DcmDsdSubService
reportActivatedEvents	DcmDsdSubService
startResponseOnEvent	DcmDsdSubService
clearResponseOnEvent	DcmDsdSubService

Parameters included	
Parameter name	Value
DcmAsyncServiceExecution	false



Parameters included

DcmDsdServiceUsed	true
DcmDsdSidTabFnc	(DISABLED)
DcmDsdSidTabServiceId	134
DcmDsdSidTabSubfuncAvail	true

5.2.2.3.49. stopResponseOnEvent

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	0

5.2.2.3.50. onDTCStatusChange

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	1

5.2.2.3.51. onChangeOfDataIdentifier

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	3

5.2.2.3.52. reportActivatedEvents

Parameters included

Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceId	4



5.2.2.3.53. startResponseOnEvent

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	5

5.2.2.3.54. clearResponseOnEvent

Parameters included	
Parameter name	Value
DcmDsdSubServiceFnc	(DISABLED)
DcmDsdSubServiceld	6

5.2.2.3.55. DcmDsp

Containers included	
Container name	Container definition
DcmDspComControl	DcmDsp DcmDspComControl
DcmDspControlDTCSetting	DcmDsp DcmDspControlDTCSetting
DcmDspPeriodicDidTransmission	DcmDsp DcmDspPeriodicDidTransmission
DcmDspPeriodicTransmission	DcmDsp DcmDspPeriodicTransmission
DcmDspReadGenericInformation	DcmDsp DcmDspReadGenericInformation
DcmDspSession	DcmDspSession
DcmDspData_0x100	DcmDspData
DcmDspData_0xF186	DcmDspData
DcmDspDataInfo_0x100	DcmDspDataInfo
DcmDspDataInfo_0xF186	DcmDspDataInfo
DcmDspDidRange	DcmDspDidRange
DcmDspDid_0x100	DcmDspDid
DcmDspDid_0xF186	DcmDspDid
DcmDspDidInfo_0x100	DcmDspDidInfo
DcmDspDidInfo_0xF186	DcmDspDidInfo

**Containers included**

DcmDspRoe	DcmDsp DcmDspRoe
DcmDspRoutine_0	DcmDspRoutine
DcmDspRoutineInfo_0	DcmDspRoutineInfo
DcmDspSecurity	DcmDsp DcmDspSecurity
DcmDspTestResultByObdmid	DcmDsp DcmDspTestResultByObdmid
DcmDspMemory	DcmDsp DcmDspMemory

Parameters included

Parameter name	Value
DcmDspDDIDCheckPerSourceDID	false
DcmDspGenerateOnlyNeededIOControlOperations	false
DcmDspMaxPeriodicDidToRead	1
DcmDspReturnControlToEcuTimeout	1
DcmDspSessionControlTiming	false
DcmDspMaxDidToRead	1
DcmDspPowerDownTime	10
DcmDspRoutineVariableLengthInBytes	false

5.2.2.3.56. DcmDspComControl**Containers included**

Container name	Container definition
DcmDspComControlSetting	DcmDspComControl DcmDspComControlSetting

Parameters included

Parameter name	Value

5.2.2.3.57. DcmDspComControlSetting**Parameters included**

Parameter name	Value



5.2.2.3.58. DcmDspControlDTCSetting

Parameters included	
Parameter name	Value
DcmSupportDTCSettingControlOptionRecord	false

5.2.2.3.59. DcmDspPeriodicDidTransmission

Parameters included	
Parameter name	Value
DcmDspMaxPeriodicDidScheduler	(DISABLED)
DcmDspPeriodicDidSamplingBufferSize	(DISABLED)
DcmDspPeriodicDidTransmissionRingEntries	(DISABLED)
DcmDspPeriodicDDIDMaxLength	(DISABLED)

5.2.2.3.60. DcmDspPeriodicTransmission

Parameters included	
Parameter name	Value
DcmDspPeriodicTransmissionSlowRate	(DISABLED)
DcmDspPeriodicTransmissionMediumRate	(DISABLED)
DcmDspPeriodicTransmissionFastRate	(DISABLED)

5.2.2.3.61. DcmDspReadGenericInformation

Parameters included	
Parameter name	Value
DcmDspMaxDemGetDiagnosticDataConsecutiveExecutions	(DISABLED)

5.2.2.3.62. DcmDspSession

Containers included	
Container name	Container definition



Containers included

DCM_DEFAULT_SESSION	DcmDspSessionRow
DCM_EXTENDED_DIAGNOSTIC_SESSION	DcmDspSessionRow
DCM_PROGRAMMING_SESSION	DcmDspSessionRow
DCM_ALL_SESSION_LEVEL	DcmDspSessionRow
DcmDspSessionRow_0	DcmDspSessionRow
DcmDspSessionRow_1	DcmDspSessionRow

Parameters included

Parameter name	Value

5.2.2.3.63. DCM_DEFAULT_SESSION

Parameters included

Parameter name	Value
DcmDspSessionLevel	1
DcmDspSessionForBoot	DCM_NO_BOOT
DcmDspSessionP2ServerMax	0.05
DcmDspSessionP2StarServerMax	5.0

5.2.2.3.64. DCM_EXTENDED_DIAGNOSTIC_SESSION

Parameters included

Parameter name	Value
DcmDspSessionForBoot	DCM_NO_BOOT
DcmDspSessionLevel	3
DcmDspSessionP2ServerMax	0.1
DcmDspSessionP2StarServerMax	2.0

5.2.2.3.65. DCM_PROGRAMMING_SESSION

Parameters included

Parameter name	Value

**Parameters included**

DcmDspSessionForBoot	DCM_NO_BOOT
DcmDspSessionLevel	2
DcmDspSessionP2ServerMax	0.5
DcmDspSessionP2StarServerMax	7.0

5.2.2.3.66. DCM_ALL_SESSION_LEVEL**Parameters included**

Parameter name	Value
DcmDspSessionForBoot	DCM_NO_BOOT
DcmDspSessionLevel	255
DcmDspSessionP2ServerMax	0.5
DcmDspSessionP2StarServerMax	5.0

5.2.2.3.67. DcmDspSessionRow_0**Parameters included**

Parameter name	Value
DcmDspSessionForBoot	DCM_OEM_BOOT
DcmDspSessionLevel	65
DcmDspSessionP2ServerMax	0.3
DcmDspSessionP2StarServerMax	3.0

5.2.2.3.68. DcmDspSessionRow_1**Parameters included**

Parameter name	Value
DcmDspSessionForBoot	DCM_SYS_BOOT
DcmDspSessionLevel	64
DcmDspSessionP2ServerMax	0.5
DcmDspSessionP2StarServerMax	5.0



5.2.2.3.69. DcmDspData_0x100

Parameters included	
Parameter name	Value
DcmDspDataConditionCheckReadFncUsed	true
DcmDspDataConditionCheckReadFnc	ReadDIDDataCondition0
DcmDspDataEcuSignal	(DISABLED)
DcmDspDataGetCurrentStateFnc	(DISABLED)
DcmDspDataGetScalingInfoFnc	(DISABLED)
DcmDspDataReadDataLengthFnc	ReadDIDDataLength0
DcmDspDataReadEcuSignal	(DISABLED)
DcmDspDataReadFnc	ReadDIDData0
DcmDspDataResetToDefaultFnc	(DISABLED)
DcmDspDataReturnControlToEcuFnc	(DISABLED)
DcmDspDataShortTermAdjustmentFnc	(DISABLED)
DcmDspDataSize	5
DcmDspDataType	UINT8
DcmDspDataUsePort	USE_DATA_ASYNC_FNC
DcmDspDataWriteFnc	WriteDIDData0

5.2.2.3.70. DcmDspData_0xF186

Parameters included	
Parameter name	Value
DcmDspDataConditionCheckReadFncUsed	false
DcmDspDataConditionCheckReadFnc	(DISABLED)
DcmDspDataEcuSignal	(DISABLED)
DcmDspDataGetCurrentStateFnc	(DISABLED)
DcmDspDataGetScalingInfoFnc	(DISABLED)
DcmDspDataReadDataLengthFnc	(DISABLED)
DcmDspDataReadEcuSignal	(DISABLED)
DcmDspDataReadFnc	Dcm_Internalf186Readout
DcmDspDataResetToDefaultFnc	(DISABLED)



Parameters included

DcmDspDataReturnControlToEcuFnc	(DISABLED)
DcmDspDataShortTermAdjustmentFnc	(DISABLED)
DcmDspDataSize	8
DcmDspDataType	UINT8
DcmDspDataUsePort	USE_DATA_SYNCH_FNC
DcmDspDataWriteFnc	(DISABLED)

5.2.2.3.71. DcmDspDataInfo_0x100

Parameters included

Parameter name	Value
DcmDspDataFixedLength	true
DcmDspDataScalingInfoSize	(DISABLED)

5.2.2.3.72. DcmDspDataInfo_0xF186

Parameters included

Parameter name	Value
DcmDspDataFixedLength	true
DcmDspDataScalingInfoSize	(DISABLED)

5.2.2.3.73. DcmDspDidRange

Parameters included

Parameter name	Value
DcmDspDidRangeHasGaps	false
DcmDspDidRangeldentifierLowerLimit	0
DcmDspDidRangeldentifierUpperLimit	0
DcmDspDidRangesDidAvailableFnc	(DISABLED)
DcmDspDidRangeMaxDataLength	1
DcmDspDidRangeReadDidFnc	(DISABLED)
DcmDspDidRangeUsePort	true



Parameters included

DcmDspDidRangeWriteDidFnc	(DISABLED)
---	------------

5.2.2.3.74. DcmDspDid_0x100

Containers included

Container name	Container definition
DcmDspDidSignal_0	DcmDspDidSignal

Parameters included

Parameter name	Value
DcmDspDidIdentifier	256
DcmDspDidUsed	true

5.2.2.3.75. DcmDspDidSignal_0

Parameters included

Parameter name	Value
DcmDspDidDataPos	2

5.2.2.3.76. DcmDspDid_0xF186

Containers included

Container name	Container definition
DcmDspDidSignal_0XF186	DcmDspDidSignal

Parameters included

Parameter name	Value
DcmDspDidIdentifier	61830
DcmDspDidUsed	true

5.2.2.3.77. DcmDspDidSignal_0XF186

Parameters included

Parameter name	Value



Parameters included

DcmDspDidDataPos	0
----------------------------------	---

5.2.2.3.78. DcmDspDidInfo_0x100

Containers included

Container name	Container definition
DcmDspDidAccess	DcmDspDidAccess

Parameters included

Parameter name	Value
DcmDspDDIDMaxElements	1
DcmDspDidDynamicallyDefined	false

5.2.2.3.79. DcmDspDidAccess

Containers included

Container name	Container definition
DcmDspDidControl	DcmDspDidAccess DcmDspDidControl
DcmDspDidRead	DcmDspDidAccess DcmDspDidRead
DcmDspDidWrite	DcmDspDidAccess DcmDspDidWrite

Parameters included

Parameter name	Value
----------------	-------

5.2.2.3.80. DcmDspDidControl

Parameters included

Parameter name	Value
DcmDspDidFreezeCurrentState	(DISABLED)
DcmDspDidResetToDefault	(DISABLED)
DcmDspDidReturnControlToEcu	(DISABLED)



Parameters included

DcmDspDidShortTermAdjustment	(DISABLED)
--	------------

5.2.2.3.81. DcmDspDidRead

Parameters included

Parameter name	Value
----------------	-------

5.2.2.3.82. DcmDspDidWrite

Parameters included

Parameter name	Value
----------------	-------

5.2.2.3.83. DcmDspDidInfo_0xF186

Containers included

Container name	Container definition
DcmDspDidAccess	DcmDspDidAccess

Parameters included

Parameter name	Value
DcmDspDidDynamicallyDefined	false
DcmDspDDIDMaxElements	1

5.2.2.3.84. DcmDspDidAccess

Containers included

Container name	Container definition
DcmDspDidControl	DcmDspDidAccess DcmDspDidControl
DcmDspDidRead	DcmDspDidAccess DcmDspDidRead
DcmDspDidWrite	DcmDspDidAccess DcmDspDidWrite

Parameters included

Parameter name	Value
----------------	-------



5.2.2.3.85. DcmDspDidControl

Parameters included	
Parameter name	Value
DcmDspDidFreezeCurrentState	(DISABLED)
DcmDspDidResetToDefault	(DISABLED)
DcmDspDidReturnControlToEcu	(DISABLED)
DcmDspDidShortTermAdjustment	(DISABLED)

5.2.2.3.86. DcmDspDidRead

Parameters included	
Parameter name	Value

5.2.2.3.87. DcmDspDidWrite

Parameters included	
Parameter name	Value

5.2.2.3.88. DcmDspRoe

Containers included	
Container name	Container definition
DcmDspRoeEvent_0	DcmDspRoeEvent
DcmDspRoeEventWindowTime_0	DcmDspRoeEventWindowTime

Parameters included

Parameter name	Value
DcmDspRoeInterMessageTime	0.0

5.2.2.3.89. DcmDspRoeEvent_0

Parameters included	
Parameter name	Value



Parameters included

DcmDspRoeEventId	0
DcmDspRoeInitialEventStatus	DCM_ROE_CLEARED

5.2.2.3.90. DcmDspRoeEventProperties

Containers included

Container name	Container definition
DcmDspRoeOnChangeOfDataIdentifier	DcmDspRoeEventProperties DcmDspRoeOnChangeOfDataIdentifier
DcmDspRoeOnDTCStatusChange	DcmDspRoeEventProperties DcmDspRoeOnDTCStatusChange

Parameters included

Parameter name	Value

5.2.2.3.91. DcmDspRoeOnChangeOfDataIdentifier

Parameters included

Parameter name	Value

5.2.2.3.92. DcmDspRoeOnDTCStatusChange

Parameters included

Parameter name	Value
DcmDspRoeEventPreconfiguredDTCMask	0

5.2.2.3.93. DcmDspRoeEventWindowTime_0

Parameters included

Parameter name	Value
DcmDspRoeEventWindowTime	DCM_ROE_EVENT_WINDOW_INFINITE
DcmDspRoeStorageState	false



5.2.2.3.94. DcmDspRoutine_0

Parameters included	
Parameter name	Value
DcmDspRequestResultsRoutineFnc	(DISABLED)
DcmDspRoutineFixedLength	true
DcmDspRoutineIdentifier	65530
DcmDspRoutineUsePort	true
DcmDspRoutineUsed	true
DcmDspRoutineUsePortASRVersion	(DISABLED)
DcmDspStartRoutineFnc	(DISABLED)
DcmDspStopRoutineFnc	(DISABLED)
DcmDspRequestResultsRoutineSupported	true
DcmDspStopRoutineSupported	true

5.2.2.3.95. DcmDspRoutineInfo_0

Containers included	
Container name	Container definition
DcmDspRoutineAuthorization	DcmDspRoutineAuthorization
DcmDspRoutineRequestResIn	DcmDspRoutineRequestResIn
DcmDspRoutineRequestResOut	DcmDspRoutineRequestResOut
DcmDspRoutineStopIn	DcmDspRoutineStopIn
DcmDspRoutineStopOut	DcmDspRoutineStopOut
DcmDspStartRoutineIn	DcmDspStartRoutineIn
DcmDspStartRoutineOut	DcmDspStartRoutineOut

Parameters included

Parameter name	Value

5.2.2.3.96. DcmDspRoutineAuthorization

Parameters included	
Parameter name	Value



5.2.2.3.97. DcmDspRoutineRequestResIn

Parameters included	
Parameter name	Value

5.2.2.3.98. DcmDspRoutineRequestResOut

Containers included	
Container name	Container definition
DcmDspRoutineRequestResOutSignal_0	DcmDspRoutineRequestResOutSignal
DcmDspRoutineRequestResOutSignal_1	DcmDspRoutineRequestResOutSignal

Parameters included	
Parameter name	Value

5.2.2.3.99. DcmDspRoutineRequestResOutSignal_0

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	8
DcmDspRoutineSignalPos	0
DcmDspRoutineSignalType	UINT8

5.2.2.3.100. DcmDspRoutineRequestResOutSignal_1

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	1
DcmDspRoutineSignalPos	8
DcmDspRoutineSignalType	BOOLEAN

5.2.2.3.101. DcmDspRoutineStopIn

Containers included	
Container name	Container definition

**Containers included**

DcmDspRoutineStopInSignal_0	DcmDspRoutineStopInSignal
DcmDspRoutineStopInSignal_1	DcmDspRoutineStopInSignal

Parameters included

Parameter name	Value

5.2.2.3.102. DcmDspRoutineStopInSignal_0**Parameters included**

Parameter name	Value
DcmDspRoutineSignalLength	16
DcmDspRoutineSignalPos	0
DcmDspRoutineSignalType	UINT16

5.2.2.3.103. DcmDspRoutineStopInSignal_1**Parameters included**

Parameter name	Value
DcmDspRoutineSignalLength	8
DcmDspRoutineSignalPos	16
DcmDspRoutineSignalType	UINT8

5.2.2.3.104. DcmDspRoutineStopOut**Containers included**

Container name	Container definition
DcmDspRoutineStopOutSignal_0	DcmDspRoutineStopOutSignal
DcmDspRoutineStopOutSignal_1	DcmDspRoutineStopOutSignal

Parameters included

Parameter name	Value



5.2.2.3.105. DcmDspRoutineStopOutSignal_0

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	1
DcmDspRoutineSignalPos	0
DcmDspRoutineSignalType	BOOLEAN

5.2.2.3.106. DcmDspRoutineStopOutSignal_1

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	32
DcmDspRoutineSignalPos	1
DcmDspRoutineSignalType	UINT32

5.2.2.3.107. DcmDspStartRoutineIn

Containers included	
Container name	Container definition
DcmDspStartRoutineInSignal_0	DcmDspStartRoutineInSignal
DcmDspStartRoutineInSignal_1	DcmDspStartRoutineInSignal

Parameters included

Parameter name	Value

5.2.2.3.108. DcmDspStartRoutineInSignal_0

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	8
DcmDspRoutineSignalPos	0
DcmDspRoutineSignalType	SINT8



5.2.2.3.109. DcmDspStartRoutineInSignal_1

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	8
DcmDspRoutineSignalPos	8
DcmDspRoutineSignalType	UINT8

5.2.2.3.110. DcmDspStartRoutineOut

Containers included	
Container name	Container definition
DcmDspStartRoutineOutSignal_0	DcmDspStartRoutineOutSignal
DcmDspStartRoutineOutSignal_1	DcmDspStartRoutineOutSignal

Parameters included

Parameter name	Value

5.2.2.3.111. DcmDspStartRoutineOutSignal_0

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	16
DcmDspRoutineSignalPos	0
DcmDspRoutineSignalType	SINT16

5.2.2.3.112. DcmDspStartRoutineOutSignal_1

Parameters included	
Parameter name	Value
DcmDspRoutineSignalLength	32
DcmDspRoutineSignalPos	16
DcmDspRoutineSignalType	SINT32



5.2.2.3.113. DcmDspSecurity

Containers included	
Container name	Container definition
DCM_SEC_LEV_L1	DcmDspSecurityRow
DCM_SEC_LEV_L3	DcmDspSecurityRow

Parameters included

Parameter name	Value
----------------	-------

5.2.2.3.114. DCM_SEC_LEV_L1

Parameters included	
Parameter name	Value
DcmDspSecurityADRSIZE	1
DcmDspSecurityCompareKeyFnc	(DISABLED)
DcmDspSecurityGetSeedFnc	(DISABLED)
DcmDspSecurityUsePort	USE_SECURITY_ASYNC_CLIENT_SERVER
DcmDspSecurityDelayTime	0.06
DcmDspSecurityDelayTimeOnBoot	0.0
DcmDspSecurityKeySize	2
DcmDspSecurityLevel	1
DcmDspSecurityNumAttDelay	1
DcmDspSecuritySeedSize	3

5.2.2.3.115. DCM_SEC_LEV_L3

Parameters included	
Parameter name	Value
DcmDspSecurityADRSIZE	1
DcmDspSecurityCompareKeyFnc	(DISABLED)
DcmDspSecurityGetSeedFnc	(DISABLED)
DcmDspSecurityUsePort	USE_SECURITY_ASYNC_CLIENT_SERVER
DcmDspSecurityDelayTime	0.06



Parameters included

DcmDspSecurityDelayTimeOnBoot	0.0
DcmDspSecurityKeySize	4
DcmDspSecurityLevel	5
DcmDspSecurityNumAttDelay	2
DcmDspSecuritySeedSize	4

5.2.2.3.116. DcmDspTestResultByObdmid

Parameters included

Parameter name	Value
DcmWriteMidBeforeTid	false

5.2.2.3.117. DcmDspMemory

Containers included

Container name	Container definition
DcmDspAddressAndLengthFormatIdentifier	DcmDspMemory DcmDspAddressAndLengthFormatIdentifier
DcmDspMemoryIdInfo_0	DcmDspMemoryIdInfo

Parameters included

Parameter name	Value
DcmDspUseMemoryId	false

5.2.2.3.118. DcmDspAddressAndLengthFormatIdentifier

Parameters included

Parameter name	Value

5.2.2.3.119. DcmDspMemoryIdInfo_0

Containers included

Container name	Container definition

**Containers included**

DcmDspReadMemoryRangeInfo_0	DcmDspReadMemoryRangeInfo
DcmDspWriteMemoryRangeInfo_0	DcmDspWriteMemoryRangeInfo

Parameters included

Parameter name	Value
DcmDspMemoryIdValue	(DISABLED)

5.2.2.3.120. DcmDspReadMemoryRangeInfo_0**Parameters included**

Parameter name	Value
DcmDspReadMemoryRangeHigh	0
DcmDspReadMemoryRangeLow	0

5.2.2.3.121. DcmDspWriteMemoryRangeInfo_0**Parameters included**

Parameter name	Value
DcmDspWriteMemoryRangeHigh	0
DcmDspWriteMemoryRangeLow	0

5.2.2.3.122. DcmDsl**Containers included**

Container name	Container definition
DcmDslBuffer_Common	DcmDslBuffer
DcmDslBuffer_Rx	DcmDslBuffer
DcmDslBuffer_Tx	DcmDslBuffer
DcmDslCallbackDCMRequestService_0	DcmDslCallbackDCMRequestService
DcmDslDiagResp	DcmDsl DcmDslDiagResp
DcmDslProtocol	DcmDsl DcmDslProtocol
DcmDslServiceRequestManufacturerNotification_0	DcmDslServiceRequestManufacturerNotification



Containers included

DcmDsIServiceRequestSupplierNotification_0	DcmDsIServiceRequestSupplierNotification
--	--

Parameters included

Parameter name	Value
----------------	-------

5.2.2.3.123. DcmDsIBuffer_Common

Parameters included

Parameter name	Value
DcmDsIBufferSize	256

5.2.2.3.124. DcmDsIBuffer_Rx

Parameters included

Parameter name	Value
DcmDsIBufferSize	8

5.2.2.3.125. DcmDsIBuffer_Tx

Parameters included

Parameter name	Value
DcmDsIBufferSize	8

5.2.2.3.126. DcmDsICallbackDCMRequestService_0

Parameters included

Parameter name	Value
----------------	-------

5.2.2.3.127. DcmDsIDiagResp

Parameters included

Parameter name	Value
----------------	-------



Parameters included

DcmDsIDiagRespMaxNumRespPend	5
DcmDsIDiagRespOnSecondDeclinedRequest	false

5.2.2.3.128. DcmDsIProtocol

Containers included

Container name	Container definition
DcmDsIProtocolRow_0	DcmDsIProtocolRow
DcmDsIProtocolRow_1	DcmDsIProtocolRow

Parameters included

Parameter name	Value

5.2.2.3.129. DcmDsIProtocolRow_0

Containers included

Container name	Container definition
DcmDsIConnection_0	DcmDsIConnection
DcmDsIConnection_1	DcmDsIConnection
DcmDsIConnection_2	DcmDsIConnection

Parameters included

Parameter name	Value
DcmDsIProtocolPriority	0
DcmSendRespPendOnTransToBoot	true
DcmDsIProtocolEndiannessConvEnabled	false
DcmDsProtocolID	DCM_UDS_ON_CAN
DcmDsProtocolsParallelExecutab	false
DcmDsProtocolPreemptTimeout	0.01
DcmDsProtocolTransType	(DISABLED)
DcmTimStrP2ServerAdjust	0.0

**Parameters included**

DcmTimStrP2StarServerAdjust	0.0
---	-----

5.2.2.3.130. DcmDsIConnection_0**Containers included**

Container name	Container definition
DcmDsIMainConnection	DcmDsIMainConnection
DcmDsIPeriodicTransmission	DcmDsIPeriodicTransmission
DcmDsIResponseOnEvent	DcmDsIResponseOnEvent

Parameters included

Parameter name	Value
----------------	-------

5.2.2.3.131. DcmDsIMainConnection**Containers included**

Container name	Container definition
DcmDsIProtocolRx_0	DcmDsIProtocolRx
DcmDsIProtocolRx_1	DcmDsIProtocolRx
DcmDsIProtocolTx	DcmDsIMainConnection DcmDsIProtocolTx

Parameters included

Parameter name	Value
DcmDsIProtocolRxTesterSourceAddr	0

5.2.2.3.132. DcmDsIProtocolRx_0**Parameters included**

Parameter name	Value
DcmDsIProtocolRxPduId	0
DcmDsIProtocolRxAddrType	DCM_PHYSICAL_TYPE
DcmDsIProtocolRxChannelId	0



5.2.2.3.133. DcmDsIProtocolRx_1

Parameters included	
Parameter name	Value
DcmDsIProtocolRxPduld	1
DcmDsIProtocolRxAddrType	DCM_FUNCTIONAL_TYPE
DcmDsIProtocolRxChannelId	0

5.2.2.3.134. DcmDsIProtocolTx

Parameters included	
Parameter name	Value
DcmDsITxConfirmationPduld	0

5.2.2.3.135. DcmDsIPeriodicTransmission

Parameters included	
Parameter name	Value

5.2.2.3.136. DcmDsIResponseOnEvent

Parameters included	
Parameter name	Value
DcmDsIRoeTxConfirmationPduld	(DISABLED)

5.2.2.3.137. DcmDsIConnection_1

Containers included	
Container name	Container definition
DcmDsIMainConnection	DcmDsIMainConnection
DcmDsIPeriodicTransmission	DcmDsIPeriodicTransmission
DcmDsIResponseOnEvent	DcmDsIResponseOnEvent



Parameters included

Parameter name	Value

5.2.2.3.138. DcmDsIMainConnection

Containers included

Container name	Container definition
DcmDsIProtocolTx	DcmDsIMainConnection DcmDsIProtocolTx

Parameters included

Parameter name	Value
DcmDsIProtocolRxTesterSourceAddr	0

5.2.2.3.139. DcmDsIProtocolTx

Parameters included

Parameter name	Value
DcmDsITxConfirmationPduld	0

5.2.2.3.140. DcmDsIPeriodicTransmission

Containers included

Container name	Container definition
DcmDsIPeriodicConnection	DcmDsIPeriodicConnection

Parameters included

Parameter name	Value

5.2.2.3.141. DcmDsIPeriodicConnection

Parameters included

Parameter name	Value



Parameters included

DcmDsIPeriodicTxConfirmationPduld	1
---	---

5.2.2.3.142. DcmDsIResponseOnEvent

Parameters included

Parameter name	Value
DcmDsIRoeTxConfirmationPduld	(DISABLED)

5.2.2.3.143. DcmDsIConnection_2

Containers included

Container name	Container definition
DcmDsIMainConnection	DcmDsIMainConnection
DcmDsIPeriodicTransmission	DcmDsIPeriodicTransmission
DcmDsIResponseOnEvent	DcmDsIResponseOnEvent

Parameters included

Parameter name	Value
----------------	-------

5.2.2.3.144. DcmDsIMainConnection

Containers included

Container name	Container definition
DcmDsIProtocolTx	DcmDsIMainConnection DcmDsIProtocolTx

Parameters included

Parameter name	Value
DcmDsIProtocolRxTesterSourceAddr	0

5.2.2.3.145. DcmDsIProtocolTx

Parameters included

Parameter name	Value
----------------	-------



Parameters included

DcmDslTxConfirmationPduld	0
---	---

5.2.2.3.146. DcmDslPeriodicTransmission

Parameters included

Parameter name	Value
----------------	-------

5.2.2.3.147. DcmDslResponseOnEvent

Parameters included

Parameter name	Value
DcmDslRoeTxConfirmationPduld	2

5.2.2.3.148. DcmDslProtocolRow_1

Containers included

Container name	Container definition
DcmDslConnection_3	DcmDslConnection
DcmDslConnection_4	DcmDslConnection
DcmDslConnection_5	DcmDslConnection

Parameters included

Parameter name	Value
DcmDslProtocolPriority	1
DcmSendRespPendOnTransToBoot	true
DcmDslProtocolEndiannessConvEnabled	false
DcmDslProtocolID	DCM_UDS_ON_FLEXRAY
DcmDslProtocolsParallelExecutab	false
DcmDslProtocolPreemptTimeout	0.01
DcmDslProtocolTransType	(DISABLED)
DcmTimStrP2ServerAdjust	0.0

**Parameters included**

DcmTimStrP2StarServerAdjust	0.0
---	-----

5.2.2.3.149. DcmDsIConnection_3**Containers included**

Container name	Container definition
DcmDsIMainConnection	DcmDsIMainConnection
DcmDsIPeriodicTransmission	DcmDsIPeriodicTransmission
DcmDsIResponseOnEvent	DcmDsIResponseOnEvent

Parameters included

Parameter name	Value
----------------	-------

5.2.2.3.150. DcmDsIMainConnection**Containers included**

Container name	Container definition
DcmDsIProtocolRx_0	DcmDsIProtocolRx
DcmDsIProtocolRx_1	DcmDsIProtocolRx
DcmDsIProtocolTx	DcmDsIMainConnection DcmDsIProtocolTx

Parameters included

Parameter name	Value
DcmDsIProtocolRxTesterSourceAddr	0

5.2.2.3.151. DcmDsIProtocolRx_0**Parameters included**

Parameter name	Value
DcmDsIProtocolRxPduId	2
DcmDsIProtocolRxAddrType	DCM_PHYSICAL_TYPE
DcmDsIProtocolRxChannelId	0



5.2.2.3.152. DcmDsIProtocolRx_1

Parameters included	
Parameter name	Value
DcmDsIProtocolRxPduld	3
DcmDsIProtocolRxAddrType	DCM_FUNCTIONAL_TYPE
DcmDsIProtocolRxChannelId	0

5.2.2.3.153. DcmDsIProtocolTx

Parameters included	
Parameter name	Value
DcmDsITxConfirmationPduld	3

5.2.2.3.154. DcmDsIPeriodicTransmission

Parameters included	
Parameter name	Value

5.2.2.3.155. DcmDsIResponseOnEvent

Parameters included	
Parameter name	Value
DcmDsIRoeTxConfirmationPduld	(DISABLED)

5.2.2.3.156. DcmDsIConnection_4

Containers included	
Container name	Container definition
DcmDsIMainConnection	DcmDsIMainConnection
DcmDsIPeriodicTransmission	DcmDsIPeriodicTransmission
DcmDsIResponseOnEvent	DcmDsIResponseOnEvent



Parameters included

Parameter name	Value

5.2.2.3.157. DcmDsIMainConnection

Containers included

Container name	Container definition
DcmDsIProtocolTx	DcmDsIMainConnection DcmDsIProtocolTx

Parameters included

Parameter name	Value
DcmDsIProtocolRxTesterSourceAddr	0

5.2.2.3.158. DcmDsIProtocolTx

Parameters included

Parameter name	Value
DcmDsITxConfirmationPduld	0

5.2.2.3.159. DcmDsIPeriodicTransmission

Containers included

Container name	Container definition
DcmDsIPeriodicConnection_0	DcmDsIPeriodicConnection

Parameters included

Parameter name	Value

5.2.2.3.160. DcmDsIPeriodicConnection_0

Parameters included

Parameter name	Value



Parameters included

DcmDsIPeriodicTxConfirmationPduld	4
---	---

5.2.2.3.161. DcmDsIResponseOnEvent

Parameters included

Parameter name	Value
DcmDsIRoeTxConfirmationPduld	(DISABLED)

5.2.2.3.162. DcmDsIConnection_5

Containers included

Container name	Container definition
DcmDsIMainConnection	DcmDsIMainConnection
DcmDsIPeriodicTransmission	DcmDsIPeriodicTransmission
DcmDsIResponseOnEvent	DcmDsIResponseOnEvent

Parameters included

Parameter name	Value
----------------	-------

5.2.2.3.163. DcmDsIMainConnection

Containers included

Container name	Container definition
DcmDsIProtocolTx	DcmDsIMainConnection DcmDsIProtocolTx

Parameters included

Parameter name	Value
DcmDsIProtocolRxTesterSourceAddr	0

5.2.2.3.164. DcmDsIProtocolTx

Parameters included

Parameter name	Value
----------------	-------



Parameters included

DcmDsITxConfirmationPduld	0
---	---

5.2.2.3.165. DcmDsIPeriodicTransmission

Parameters included

Parameter name	Value
----------------	-------

5.2.2.3.166. DcmDsIResponseOnEvent

Parameters included

Parameter name	Value
DcmDsIRoeTxConfirmationPduld	(DISABLED)

5.2.2.3.167. DcmDsIServiceRequestManufacturerNotification_0

Parameters included

Parameter name	Value
----------------	-------

5.2.2.3.168. DcmDsIServiceRequestSupplierNotification_0

Parameters included

Parameter name	Value
----------------	-------

5.2.2.3.169. DcmDefensiveProgramming

Parameters included

Parameter name	Value
DcmDefProgEnabled	false
DcmPrecondAssertEnabled	false
DcmPostcondAssertEnabled	false
DcmStaticAssertEnabled	false
DcmUnreachAssertEnabled	false



Parameters included

DcmInvariantAssertEnabled	false
---	-------

5.2.2.3.170. DcmGeneral

Containers included

Container name	Container definition
DcmServiceAPI	DcmGeneral DcmServiceAPI

Parameters included

Parameter name	Value
DcmDD DIDStorage	false
DcmDevErrorDetect	true
DcmDSIEnableSetActiveDiagnosis	false
DcmDynamicallyDefinedDIDInitPolicy	(DISABLED)
DcmDynamicallyDefinedPeriodicDIDInitPolicy	(DISABLED)
DcmNRCPolicyISOVersion	ISO14229_2013
DcmModeDeclarationSupport	false
DcmRequestManufacturerNotificationEnabled	true
DcmRequestSupplierNotificationEnabled	true
DcmRespondAllRequest	true
DcmTaskTime	0.01
DcmVersionInfoApi	true
DcmRteUsage	true
DcmRequestSeedAfterFailedCompareKey	false

5.2.2.3.171. DcmServiceAPI

Parameters included

Parameter name	Value
DcmEnableASR40ServiceAPI	false
DcmEnableASR42ServiceAPI	false
DcmDefaultASRServiceAPI	AUTOSAR_40



5.2.2.3.172. DcmPageBufferCfg

Parameters included	
Parameter name	Value
DcmPagedBufferEnabled	true
DcmPagedBufferTimeout	0.05

5.2.2.3.173. DcmProcessingConditions

Containers included	
Container name	Container definition
DcmModeCondition_0	DcmModeCondition
DcmModeCondition_1	DcmModeCondition
DcmModeRule_0	DcmModeRule
DcmModeRule_1	DcmModeRule

Parameters included

Parameter name	Value
DcmConditionType	DCM_EQUALS

5.2.2.3.174. DcmModeCondition_0

Containers included	
Container name	Container definition
DcmBswModeRef	DcmBswModeRef
DcmSwcModeRef	DcmSwcModeRef

Parameters included

Parameter name	Value
DcmConditionType	DCM_EQUALS

5.2.2.3.175. DcmBswModeRef

Parameters included	
Parameter name	Value
DcmConditionType	DCM_EQUALS



5.2.2.3.176. DcmSwcModeRef

Parameters included	
Parameter name	Value

5.2.2.3.177. DcmModeCondition_1

Containers included	
Container name	Container definition
DcmBswModeRef	DcmBswModeRef
DcmSwcModeRef	DcmSwcModeRef

Parameters included	
Parameter name	Value
DcmConditionType	DCM_EQUALS_NOT

5.2.2.3.178. DcmBswModeRef

Parameters included	
Parameter name	Value

5.2.2.3.179. DcmSwcModeRef

Parameters included	
Parameter name	Value

5.2.2.3.180. DcmModeRule_0

Parameters included	
Parameter name	Value
DcmLogicalOperator	DCM_OR
DcmModeRuleNrcValue	10



5.2.2.3.181. DcmModeRule_1

Parameters included	
Parameter name	Value
DcmLogicalOperator	DCM_AND
DcmModeRuleNrcValue	22

5.2.3. Application programming interface (API)

5.2.3.1. Type definitions

5.2.3.1.1. Dcm_CommunicationModeType

Purpose	Type of Requested Communication Mode for UDS Service 'CommunicationControl'.
Type	uint8

5.2.3.1.2. Dcm_ConfirmationStatusType

Purpose	Type used for status informations in confirmations sent to Central Diagnostic Software Component.
Type	uint8

5.2.3.1.3. Dcm_DIDIDType

Purpose	Type of an unconfigured Did.
Type	uint16

5.2.3.1.4. Dcm_DTRStatusType

Purpose	Type used for providing the status of DTR's.
Type	uint8



5.2.3.1.5. Dcm_DidSupportedType

Purpose	Type used call-back functions for DidRanges, indicates if Dids are supported.
Type	uint8

5.2.3.1.6. Dcm_EcuStartModeType

Purpose	Allows the DCM to know if a diagnostic response shall be sent in the case of a jump from bootloader.
Type	uint8

5.2.3.1.7. Dcm_IdContextType

Purpose	Type for context identifier As a part of the message context the context identifier may be used to determine the relation between a diagnostic request the confirmation after sending the diagnostic response.
Type	uint8

5.2.3.1.8. Dcm_LinkControlRequestType

Purpose	Type to indicate which kind of LinkControl shall be realized by Callout.
Type	uint8

5.2.3.1.9. Dcm_MsgAddInfoType

Purpose	Additional information on message request.	
Type	struct	
Members	unsigned int reqType	Addressing type of the request(0=physical request, 1=functional request).
	unsigned int suppressPosResponse	Suppress positive response (0 = no (do not supress), 1 = yes (no positive response)).
	unsigned int requestOrigin	Origin of the request ▶ 0 = Invalid origin



	<ul style="list-style-type: none"> ▶ 1 = External request ▶ 2 = ROE-generated request ▶ 3 = Periodic transmission request ▶ 4 = PostBoot response request with no answer ▶ 5 = PostBoot response request with answer.
--	--

5.2.3.1.10. Dcm_MsgContextPtrType

Purpose	This data structure contains all information which is necessary to process a diagnostic message from request to response and response confirmation.
Type	Dcm_MsgContextType *
Description	Pointer to message related information for one diagnostic request.

5.2.3.1.11. Dcm_MsgContextType

Purpose	This data structure contains all information which is necessary to process a diagnostic message from request to response and response confirmation.	
Type	struct	
Members	Dcm_MsgType reqData	Request data, starting directly after service identifier (which is not part of this data).
	Dcm_MsgLenType reqDataLen	Request data length (excluding service identifier).
	Dcm_MsgType resData	Positive response data (excluding service identifier).
	Dcm_MsgLenType resDataLen	Positive response data length (excluding service identifier).
	Dcm_MsgAddInfoType msgAddInfo	Additional information about service request and response.
	Dcm_MsgLenType resMaxDataLen	The maximal length of a response is restricted by the size of the buffer.
	Dcm_IdContextType idContext	context identifier can be used to determine the relation between request and response confirmation



	PduIdType dcmRxPduId	Pdu identifier on which the request was received.
--	----------------------	---

5.2.3.1.12. Dcm_MsgItemType

Purpose	Base type for diagnostic message item.
Type	uint8

5.2.3.1.13. Dcm_MsgLenType

Purpose	Length of diagnostic message.
Type	uint32

5.2.3.1.14. Dcm_MsgType

Purpose	Base type for diagnostic message (request, positive or negative response).
Type	Dcm_MsgItemType *

5.2.3.1.15. Dcm_NegativeResponseCodeType

Purpose	Type used for negative response codes.
Type	uint8

5.2.3.1.16. Dcm_OpStatusType

Purpose	Type used for providing the status of service being processed, to the application.
Type	uint8

5.2.3.1.17. Dcm_PduInfoPtrType

Purpose	Helper type definition to be used to prevent nested compiler abstraction macros which may lead to problems for some tool chains.
---------	--



Type	PduInfoType *
------	---------------

5.2.3.1.18. Dcm_ProgConditionsType

Purpose	Type Used in Dcm_SetProgConditions() to allow the integrator to store relevant information prior to jumping to bootloader.	
Type	struct	
Members	uint16 TesterSourceAddr	Tester source address configured per protocol.
	uint8 ProtocolId	Id of the protocol on which the request has been received.
	uint8 Sid	Service identifier of the received request.
	uint8 SubFncId	Identifier of the received subfunction.
	boolean ReprogrammingRequest	Set/Clear the ReprogrammingRequest flag on ECU reset.
	boolean ApplUpdated	Set/Clear the ApplUpdated flag on ECU reset.
	boolean ResponseRequired	Set/Clear the ResponseRequired flag on ECU reset.

5.2.3.1.19. Dcm_ProtocolType

Purpose	Type of diagnostic protocol.
Type	uint8

5.2.3.1.20. Dcm_ResetModeType

Purpose	Type to indicate which kind of reset shall be realized by BswM.
Type	uint8

5.2.3.1.21. Dcm_ResponseConfirmationTypeType

Purpose	Encodes the type of the response which is transmitted.
---------	--



Type	uint8
------	-------

5.2.3.1.22. Dcm_ReturnReadMemoryType

Purpose	Return values of callout 'Dcm_ReadMemory'.
Type	uint8

5.2.3.1.23. Dcm_ReturnWriteMemoryType

Purpose	Return type of callout 'Dcm_WriteMemory'.
Type	uint8

5.2.3.1.24. Dcm_RoeStateType

Purpose	Roe State types.
Type	uint8

5.2.3.1.25. Dcm_SecLevelType

Purpose	Type of diagnostic security level.
Type	uint8

5.2.3.1.26. Dcm_ServiceIDType

Purpose	Type to indicate ID of a diagnostic service.
Type	uint8

5.2.3.1.27. Dcm_SesCtrlType

Purpose	Type of diagnostic session.
Type	uint8



5.2.3.1.28. Dcm_StatusType

Purpose	This type is used as return type for API function provided by Dcm or the Central Diagnostic software component.
Type	uint8

5.2.3.2. Macro constants

5.2.3.2.1. DCM_ALL_SESSION_LEVEL

Purpose	Value for ALL session level.
Value	0xFFU

5.2.3.2.2. DCM_API_CALLER_GENERIC_SVCH_IMPLEMENTATION_ID

Purpose	External API service ID for Dcm_APICaller_GenericSvcHImplementation.
Value	0x7FU
Description	Definition of DCM_API_CALLER_GENERIC_SVCH_IMPLEMENTATION_ID.

5.2.3.2.3. DCM_AR_RELEASE_MAJOR_VERSION

Purpose	AUTOSAR release major version.
Value	4U

5.2.3.2.4. DCM_AR_RELEASE_MINOR_VERSION

Purpose	AUTOSAR release minor version.
Value	0U

5.2.3.2.5. DCM_AR_RELEASE_REVISION_VERSION

Purpose	AUTOSAR release revision version.
---------	-----------------------------------



Value	3U
--------------	----

5.2.3.2.6. DCM_ASYNC_PROCESSING_DONE

Purpose	Signals the end of asynchronous processing.
Value	0x08U

5.2.3.2.7. DCM_ASYNC_PROCESSING_WAIT

Purpose	Signals a wait state for asynchronous processing.
Value	0x07U

5.2.3.2.8. DCM_BOOTLOADER_RESET

Purpose	Jump to Bootloader.
Value	6U

5.2.3.2.9. DCM_CANCEL

Purpose	Indicates request to cancel a pending operation.
Value	0x02U

5.2.3.2.10. DCM_COLD_START

Purpose	The ECU starts normally.
Value	0U

5.2.3.2.11. DCM_CONFIRMED_NOK

Purpose	Confirms failure in transmission of the response.
Value	0x06U



5.2.3.2.12. DCM_CONFIRMED_OK

Purpose	Confirms successful transmission of the response.
Value	0x05U

5.2.3.2.13. DCM_DEFAULT_SESSION

Purpose	Value for Default session level.
Value	0x01U

5.2.3.2.14. DCM_DID_NOT_SUPPORTED

Purpose	Value of Dcm_DidSupportedType: Did is not supported.
Value	((uint8)1U)

5.2.3.2.15. DCM_DID_SUPPORTED

Purpose	Value of Dcm_DidSupportedType: Did is supported.
Value	((uint8)0U)

5.2.3.2.16. DCM_DISABLE_RAPID_POWER_SHUTDOWN_RESET

Purpose	Reset mode type is 'Disable the Rapid power shutdown'.
Value	5U

5.2.3.2.17. DCM_DISABLE_RX_ENABLE_TX_NM

Purpose	Disable the Rx and enable the Tx for network management communication.
Value	6U

5.2.3.2.18. DCM_DISABLE_RX_ENABLE_TX_NORM

Purpose	Disable the Rx and enable the Tx for normal communication.
----------------	--



Value	2U
--------------	----

5.2.3.2.19. DCM_DISABLE_RX_ENABLE_TX_NORM_NM

Purpose	Enable the Rx and disable the Tx for normal and network management communication.
Value	0x0AU

5.2.3.2.20. DCM_DISABLE_RX_TX_NM

Purpose	Diable Rx and Tx for network management communication.
Value	7U

5.2.3.2.21. DCM_DISABLE_RX_TX_NORM

Purpose	Disable Rx and Tx for normal communication.
Value	3U

5.2.3.2.22. DCM_DISABLE_RX_TX_NORM_NM

Purpose	Disable Rx and Tx for normal and network.
Value	0x0BU

5.2.3.2.23. DCM_DTRSTATUS_INVISIBLE

Purpose	Test Id is not available for execution.
Value	1U

5.2.3.2.24. DCM_DTRSTATUS_VISIBLE

Purpose	Test Id is available for execution.
----------------	-------------------------------------



Value	0U
--------------	----

5.2.3.2.25. DCM_ENABLE_RAPID_POWER_SHUTDOWN_RESET

Purpose	Reset mode type is 'Enable the Rapid power shutdown'.
Value	4U

5.2.3.2.26. DCM_ENABLE_RX_DISABLE_TX_NM

Purpose	Enable Rx and disable the Tx for network management communication.
Value	5U

5.2.3.2.27. DCM_ENABLE_RX_DISABLE_TX_NORM

Purpose	Enable the Rx and disable the Tx for normal communication.
Value	1U

5.2.3.2.28. DCM_ENABLE_RX_DISABLE_TX_NORM_NM

Purpose	Enable the Rx and disable the Tx for normal and network management communication.
Value	9U

5.2.3.2.29. DCM_ENABLE_RX_TX_NM

Purpose	Enable the Rx and Tx for network management communication.
Value	4U

5.2.3.2.30. DCM_ENABLE_RX_TX_NORM

Purpose	Enable the Rx and Tx for normal communication.
----------------	--



Value	0U
--------------	----

5.2.3.2.31. DCM_ENABLE_RX_TX_NORM_NM

Purpose	Enable Rx and Tx for normal and network management communication.
Value	8U

5.2.3.2.32. DCM_EXTENDED_DIAGNOSTIC_SESSION

Purpose	Value for Extended session level.
Value	0x03U

5.2.3.2.33. DCM_E_ABORT

Purpose	Operation aborted.
Value	0x1EU

5.2.3.2.34. DCM_E_BRAKESWITCH_NOTCLOSED

Purpose	For safety reasons, this is required for certain tests before it begins, and must be maintained for the entire duration of the test.
Value	0x8fU

5.2.3.2.35. DCM_E_BUSYREPEATREQUEST

Purpose	This return value indicates that the application is temporarily too busy to perform the requested operation. In this circumstance the client shall perform repetition of the "identical request message" or "another request message". The repetition of the request shall be delayed by a time specified in the respective implementation documents. Example: In a multi-client environment the diagnostic request of one client might be blocked temporarily by a NRC 0x21 while a different client finishes a diagnostic task. Note: If the application is able to perform the diagnostic task but needs additional time to finish the task and prepare the response, the NRC 0x78 shall be used instead of NRC 0x21. This return value is in general supported by each diagnos-
----------------	---



	tic service, as not otherwise stated in the data link specific implementation document, therefore it is not listed in the list of applicable return values of the diagnostic services.
Value	0x21U

5.2.3.2.36. DCM_E_COMPARE_KEY_FAILED

Purpose	ECU compares by tester requested key with own calculated key When comparison fails this definition is used. (used by API: Xxx_CompareKey() within processing of security access service).
Value	0x0BU

5.2.3.2.37. DCM_E_CONDITIONSNOTCORRECT

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite conditions are not met.
Value	0x22U

5.2.3.2.38. DCM_E_DONE

Purpose	Value indicates Operation Completed for the execution status of types of Dcm Service function.
Value	0x1AU

5.2.3.2.39. DCM_E_ENGINENOTRUNNING

Purpose	This is required for those actuator tests which cannot be actuated unless the Engine is running. This is different from RPM too low negative response, and needs to be allowed.
Value	0x84U

5.2.3.2.40. DCM_E_ENGINERUNNING

Purpose	This is required for those actuator tests which cannot be actuated while the Engine is running. This is different from RPM too high negative response, and needs to be allowed.
Value	0x85U



Value	0x83U
--------------	-------

5.2.3.2.41. DCM_E_ENGINERUNTIMETOLOW

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite condition for engine run time is not met (current engine run time is below a pre-programmed limit).
Value	0x85U

5.2.3.2.42. DCM_E_EXCEEDNUMBEROFATTEMPTS

Purpose	This return value indicates that the requested action will not be taken because the client has unsuccessfully attempted to gain security access more times than the application's security strategy will allow.
Value	0x36U

5.2.3.2.43. DCM_E_FAILUREPREVENTSEXECUTIONOFREQUESTEDACTION

Purpose	This response code indicates that the requested action will not be taken because a failure condition, identified by a DTC (with at least one DTC status bit for TestFailed, Pending, Confirmed or TestFailedSinceLastClear set to 1), has occurred and that this failure condition prevents the server from performing the requested action.
Value	0x26U

5.2.3.2.44. DCM_E_FORCE_RCRP

Purpose	Application request the transmission of a response Response Pending (NRC 0x78) (used by API: XXX_DcmGetSesChgPermission()).
Value	0x0CU

5.2.3.2.45. DCM_E_GENERALPROGRAMMINGFAILURE

Purpose	This return value indicates that the application detected an error when erasing or programming a memory location in the permanent memory device (e.g. Flash Memory).
Value	0x72U



5.2.3.2.46. DCM_E_GENERALREJECT

Purpose	This return value indicates that the requested action has been rejected by the application. The generalReject return value shall only be implemented in the application if none of the negative return values defined in this document meet the needs of the implementation. At no means shall this return value be a general replacement for the return values defined in this document.
Value	0x10U

5.2.3.2.47. DCM_E_INCORRECTMESSAGELENGTHORINVALIDFORMAT

Purpose	This return value indicates that the requested action will not be taken because the length of the received request message does not match the prescribed length for the specified service or the format of the parameters do not match the prescribed format for the specified service.
Value	0x13U

5.2.3.2.48. DCM_E_INVALIDKEY

Purpose	This return value indicates that the application has not given security access because the key sent by the client did not match with the key in the application's memory. This counts as an attempt to gain security. The application shall remain locked and increment is internal securityAccessFailed counter.
Value	0x35U

5.2.3.2.49. DCM_E_NORESPONSEFROMSUBNETCOMPONENT

Purpose	This response code indicates that the server has received the request but the requested action could not be performed by the server, as a subnet component which is necessary to supply the requested information did not respond within the specified time.
Value	0x25U

5.2.3.2.50. DCM_E_NOT_OK

Purpose	This value is representing a unsuccessful operation.
----------------	--



Value	0x01U
--------------	-------

5.2.3.2.51. DCM_E_NO_DATA

Purpose	RTE interface call returned no data.(This is not error).
Value	0x83U

5.2.3.2.52. DCM_E_OK

Purpose	This value is representing a successful operation.
Value	0x00U

5.2.3.2.53. DCM_E_PENDING

Purpose	Session change is allowed but the DCM is not allowed to send a positive reponse. Instead a response pending must be send back and the Bootloader must finish the service.
Value	0x0AU

5.2.3.2.54. DCM_E_PERIODICID_NOT_ACCEPTED

Purpose	Periodic transmission request is not accepted by DCM (e.g. old Periodic transmission is not finished) (used by API: DsllInternal_ResponseOnOneDataByPeriodicId()).
Value	0x07U

5.2.3.2.55. DCM_E_POSITIVERESPONSE

Purpose	This response code shall not be used in a negative response message. This positiveResponse parameter value is reserved for server-internal implementation.
Value	0x00U

5.2.3.2.56. DCM_E_PROCESS_PAGE

Purpose	The DCM shall send an "response pending" immediately.
----------------	---



Value	0x1DU
--------------	-------

5.2.3.2.57. DCM_E_PROTOCOL_NOT_ALLOWED

Purpose	Application does not allow start of requested protocol (used by API: XXX_DcmStartProtocol()).
Value	0x05U

5.2.3.2.58. DCM_E_REQUESTCORRECTLYRECEIVEDRESPONSEPENDING

Purpose	This return value indicates that the request message was received correctly, and that all parameters in the request message were valid, but the action to be performed is not yet completed and the application is not yet ready to receive another request. As soon as the requested service has been completed, the application shall send a positive response message or negative response message with a return value different from this. The negative response message with this return value may be repeated by the application until the requested service is completed and the final response message is sent. This return value might impact the application layer timing parameter values. The detailed specification shall be included in the data link specific implementation document. This return value shall only be used in a negative response message if the application will not be able to receive further request messages from the client while completing the requested diagnostic service. When this return value is used, the application shall always send a final response (positive or negative) independent of the suppressPosRspMsgIndicationBit value. A typical example where this return value may be used is when the client has sent a request message, which includes data to be programmed or erased in flash memory of the application. If the programming_erasing routine (usually executed out of RAM) is not able to support serial communication while writing to the flash memory the application shall send a negative response message with this return value. This return value is in general supported by each diagnostic service, as not otherwise stated in the data link specific implementation document, therefore it is not listed in the list of applicable return values of the diagnostic services.
Value	0x78U

5.2.3.2.59. DCM_E_REQUESTOUTOFRANGE

Purpose	This return value indicates that the requested action will not be taken because the application has detected that the request message contains a parameter which attempts
----------------	---



	to substitute a value beyond its range of authority (e.g. attempting to substitute a data byte of 111 when the data is only defined to 100), or which attempts to access a dataIdentifier_routineIdentifier that is not supported or not supported in active session. This return value shall be implemented for all services, which allow the client to read data, write data or adjust functions by data in the application.
Value	0x31U

5.2.3.2.60. DCM_E_REQUESTSEQUENCEERROR

Purpose	This return value indicates that the requested action will not be taken because the application expects a different sequence of request messages or message as sent by the client. This may occur when sequence sensitive requests are issued in the wrong order. EXAMPLE A successful SecurityAccess service specifies a sequence of requestSeed and sendKey as subfunctions in the request messages. If the sequence is sent different by the client the application shall send a negative response message with the negative return value 0x24- . requestSequenceError.
Value	0x24U

5.2.3.2.61. DCM_E_REQUEST_ENV_NOK

Purpose	Diagnostic request is not allowed by application because of not fitting environmental conditions -> (used by API: XXX_DcmIndication()) Base item type to transport status information.
Value	0x09U

5.2.3.2.62. DCM_E_REQUEST_NOT_ACCEPTED

Purpose	Application rejects diagnostic request -> (used by API: XXX_DcmIndication()).
Value	0x08U

5.2.3.2.63. DCM_E_REQUIREDTIMEDELAYNOTEXPIRED

Purpose	This return value indicates that the requested action will not be taken because the client's latest attempt to gain security access was initiated before the application's required timeout period had elapsed.
Value	0x37U



5.2.3.2.64. DCM_E_RESPONSETOOLONG

Purpose	This return value indicates that the buffer of the lower layer is not large enough to transmit all data of the response.
Value	0x14U

5.2.3.2.65. DCM_E_ROE_NOT_ACCEPTED

Purpose	ResponseOnOneEvent request is not accepted by DCM (e.g. old ResponseOnOneEvent is not finished) (used by API: Dcm_DsInternal_ResponseOnOneEvent()).
Value	0x06U

5.2.3.2.66. DCM_E_RPMTOOHIGH

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite condition for RPM is not met (current RPM is above a pre-programmed maximum threshold).
Value	0x81U

5.2.3.2.67. DCM_E_RPMTOOLOW

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite condition for RPM is not met (current RPM is below a pre-programmed minimum threshold).
Value	0x82U

5.2.3.2.68. DCM_E_SECURITYACCESSDENIED

Purpose	This return value indicates that the requested action will not be taken because the application's security strategy has not been satisfied by the client. The application shall send this return value if one of the following cases occur: - the test conditions of the application are not met, - the required message sequence e.g. DiagnosticSessionControl, securityAccess is not met, - the client has sent a request message which requires an unlocked application.
----------------	---



Value	0x33U
Description	Beside the mandatory use of this negative return value as specified in the applicable services within this standard, this negative return value can also be used for any case where security is required and is not yet granted to perform the required service.

5.2.3.2.69. DCM_E_SERVICENOTSUPPORTED

Purpose	This return value indicates that the requested action will not be taken because the application does not support the requested service. The application shall send this return value in case the client has sent a request message with a service identifier, which is either unknown or not supported by the application. Therefore this negative return value is not shown in the list of negative return values to be supported for a diagnostic service, because this negative return value is not applicable for supported services.
Value	0x11U

5.2.3.2.70. DCM_E_SERVICENOTSUPPORTEDINACTIVESESSION

Purpose	This return value indicates that the requested action will not be taken because the application does not support the requested service in the session currently active. This return value shall only be used when the requested service is known to be supported in another session, otherwise return value SNS (serviceNotSupported) shall be used. This return value is in general supported by each diagnostic service, as not otherwise stated in the data link specific implementation document, therefore it is not listed in the list of applicable return values of the diagnostic services.
Value	0x7FU

5.2.3.2.71. DCM_E_SESSION_NOT_ALLOWED

Purpose	Application does not allow start of requested session (used by API: XXX_-DcmGetSesChgPermission()).
Value	0x04U

5.2.3.2.72. DCM_E_SHIFTERLEVERNOTINPARK

Purpose	For safety reasons, this is required for certain tests before it begins, and must be maintained for the entire duration of the test.
----------------	--



Value	0x90U
--------------	-------

5.2.3.2.73. DCM_E_SUBFUNCTIONNOTSUPPORTED

Purpose	This return value indicates that the requested action will not be taken because the application does not support the service specific parameters of the request message. The application shall send this return value in case the client has sent a request message with a known and supported service identifier but with "sub function" which is either unknown or not supported.
Value	0x12U

5.2.3.2.74. DCM_E_SUBFUNCTIONNOTSUPPORTEDINACTIVESESSION

Purpose	This return value indicates that the requested action will not be taken because the application does not support the requested sub-function in the session currently active. This return value shall only be used when the requested sub-function is known to be supported in another session, otherwise return value SFNS (subFunctionNotSupported) shall be used. This return value shall be supported by each diagnostic service with a sub-function parameter, if not otherwise stated in the data link specific implementation document, therefore it is not listed in the list of applicable return values of the diagnostic services.
Value	0x7EU

5.2.3.2.75. DCM_E_TEMPERATURETOOHIGH

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite condition for temperature is not met (current temperature is above a pre-programmed maximum threshold).
Value	0x86U

5.2.3.2.76. DCM_E_TEMPERATURETOOLOW

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite condition for temperature is not met (current temperature is below a pre-programmed minimum threshold).
Value	0x87U



5.2.3.2.77. DCM_E_THROTTLE_PEDALTOOHIGH

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite condition for throttle/pedal position is not met (current TP/APP is above a pre-programmed maximum threshold).
Value	0x8aU

5.2.3.2.78. DCM_E_THROTTLE_PEDALTOOLOW

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite condition for throttle/pedal position is not met (current TP/APP is below a pre-programmed minimum threshold).
Value	0x8bU

5.2.3.2.79. DCM_E_TI_PREPARE_INCONSTENT

Purpose	New timing parameter are not ok, since requested values are not consistent (e.g. P2min not smaller than P2max) (used by API: Dcm_PreparesTiming-Values ()).
Value	0x03U

5.2.3.2.80. DCM_E_TI_PREPARE_LIMITS

Purpose	New timing parameter are not ok, since requested values are not within the defined limits (used by API: Dcm_PreparesTiming-Values()).
Value	0x02U

5.2.3.2.81. DCM_E_TORQUECONVERTERCLUTCHLOCKED

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite condition for torque converter clutch is not met (current TCC status above a pre-programmed limit or locked).
Value	0x91U

5.2.3.2.82. DCM_E_TRANSFERDATASUSPENDED

Purpose	This return value indicates that a data transfer operation was halted due to some fault.
----------------	--



Value	0x71U
--------------	-------

5.2.3.2.83. DCM_E_TRANSMISSIONRANGENOTINGEAR

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite condition for being in gear is not met (current transmission range is not in gear).
Value	0x8dU

5.2.3.2.84. DCM_E_TRANSMISSIONRANGENOTINNEUTRAL

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite condition for being in neutral is not met (current transmission range is not in neutral).
Value	0x8cU

5.2.3.2.85. DCM_E_UPLOADDOWNLOADNOTACCEPTED

Purpose	This return value indicates that an attempt to download to a application's memory cannot be accomplished due to some fault conditions.
Value	0x70U

5.2.3.2.86. DCM_E_VEHICLESPEEDTOOHIGH

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite condition for vehicle speed is not met (current VS is above a pre-programmed maximum threshold).
Value	0x88U

5.2.3.2.87. DCM_E_VEHICLESPEEDTOOLOW

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite condition for vehicle speed is not met (current VS is below a pre-programmed minimum threshold).
Value	0x89U



5.2.3.2.88. DCM_E_VOLTAGETOHIGH

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite condition for voltage at the primary pin of the application (ECU) is not met (current voltage is above a pre-programmed maximum threshold).
Value	0x92U

5.2.3.2.89. DCM_E_VOLTAGETOLOW

Purpose	This return value indicates that the requested action will not be taken because the application prerequisite condition for voltage at the primary pin of the application (ECU) is not met (current voltage is below a pre-programmed minimum threshold).
Value	0x93U

5.2.3.2.90. DCM_E_WRONGBLOCKSEQUENCECOUNTER

Purpose	This return value indicates that the application detected an error in the sequence of blockSequenceCounter values. Note that the repetition of a TransferData request message with a blockSequenceCounter equal to the one included in the previous TransferData request message shall be accepted by the application.
Value	0x73U

5.2.3.2.91. DCM_E_WRONG_RECORD_NUM

Purpose	Wrong record number is passed for getting FReeze frame or Extended Data Record.
Value	0x11U

5.2.3.2.92. DCM_FINALRESPONSE

Purpose	
Value	0X00U

5.2.3.2.93. DCM_FORCED_RCRRP

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



Value	0X01U
--------------	-------

5.2.3.2.94. DCM_FORCE_RCRRP_OK

Purpose	Confirms a ResponsePending transmission.
Value	0x03U

5.2.3.2.95. DCM_HARD_RESET

Purpose	Reset mode type is 'Hard Reset'.
Value	1U

5.2.3.2.96. DCM_INITIAL

Purpose	Indicates the initial call to the operation.
Value	0x00U

5.2.3.2.97. DCM_KEY_ON_OFF_RESET

Purpose	Reset mode type is 'Key ON/OFF reset'.
Value	2U

5.2.3.2.98. DCM_MODULE_ID

Purpose	AUTOSAR module identification.
Value	53U

5.2.3.2.99. DCM_NO_ACTIVE_PROTOCOL

Purpose	No protocol has been started.
Value	0x0CU



5.2.3.2.100. DCM_NO_RESET

Purpose	Reset mode type is 'No Reset'.
Value	0U

5.2.3.2.101. DCM_OBD_ON_CAN

Purpose	OBD protocol on CAN (ISO15765-4; ISO15031-5).
Value	0x00U

5.2.3.2.102. DCM_OBD_ON_FLEXRAY

Purpose	OBD protocol on Flexray (Manufacturer specific; ISO15031-5).
Value	0x01U

5.2.3.2.103. DCM_OBD_ON_IP

Purpose	OBD on Internet Protocol (Manufacturer specific; ISO15031-5).
Value	0x02U

5.2.3.2.104. DCM_OPSTATUS_INVALID

Purpose	Generic invalid opStatus.
Value	0xFFU

5.2.3.2.105. DCM_PENDING

Purpose	Indicates that a pending has been returned from previous call of the operation.
Value	0x01U

5.2.3.2.106. DCM_PERIODICTRANS_ON_CAN

Purpose	Periodic Transmission on CAN.
----------------	-------------------------------



Value	0x09U
--------------	-------

5.2.3.2.107. DCM_PERIODICTRANS_ON_FLEXRAY

Purpose	Periodic Transmission on FlexRay.
Value	0x0AU

5.2.3.2.108. DCM_PERIODICTRANS_ON_IP

Purpose	Periodic Transmission on Internet Protocol.
Value	0x0BU

5.2.3.2.109. DCM_PROGRAMMING_SESSION

Purpose	Value for Programming session level.
Value	0x02U

5.2.3.2.110. DCM_READ_FAILED

Purpose	Read failed for the service 0x36.
Value	0x02U

5.2.3.2.111. DCM_READ_FORCE_RCRRP

Purpose	Application request the transmission of a Response Pending (NRC 0x78) (used by API: Dcm_WriteMemory()).
Value	0x03U

5.2.3.2.112. DCM_READ_OK

Purpose	Read is done for the service 0x36.
Value	0x00U



5.2.3.2.113. DCM_READ_PENDING

Purpose	Read is pending for the service 0x36.
Value	0x01U

5.2.3.2.114. DCM_RESET_EXECUTION

Purpose	Request the execution of the reset.
Value	8U

5.2.3.2.115. DCM_RES_NEG_NOT_OK

Purpose	Status passed via Xxx_DcmConfirmation if a negative response was sent and status of Dcm_TpTxConfirmation was NOT OK.
Value	3U

5.2.3.2.116. DCM_RES_NEG_OK

Purpose	Status passed via Xxx_DcmConfirmation if a negative response was sent and status of Dcm_TpTxConfirmation was OK.
Value	2U

5.2.3.2.117. DCM_RES_POS_NOT_OK

Purpose	Status passed via Xxx_DcmConfirmation if a positive response was sent and status of Dcm_TpTxConfirmation was NOT OK.
Value	1U

5.2.3.2.118. DCM_RES_POS_OK

Purpose	Status passed via Xxx_DcmConfirmation if a positive response was sent and status of Dcm_TpTxConfirmation was OK.
Value	0U



5.2.3.2.119. DCM_ROE_ACTIVE

Purpose	Activate the event.
Value	0U

5.2.3.2.120. DCM_ROE_ON_CAN

Purpose	Response On Event on CAN.
Value	0x06U

5.2.3.2.121. DCM_ROE_ON_FLEXRAY

Purpose	Response On Event on FlexRay.
Value	0x07U

5.2.3.2.122. DCM_ROE_ON_IP

Purpose	Response on Event on Internet Protocol.
Value	0x08U

5.2.3.2.123. DCM_ROE_UNACTIVE

Purpose	Deactivate the event.
Value	1U

5.2.3.2.124. DCM_SAFETY_SYSTEM_DIAGNOSTIC_SESSION

Purpose	Value for Safety System session level.
Value	0x04U

5.2.3.2.125. DCM_SEC_LEV_LOCKED

Purpose	Value for locked security state.
Value	0x00U



5.2.3.2.126. DCM_SERVID_APPLICATIONTRANSMISSIONCONFIRMATION

Purpose	EB API service ID.
Value	0x41U
Description	Definition of DCM_SERVID_APPLICATIONTRANSMISSIONCONFIRMATION.

5.2.3.2.127. DCM_SERVID_COMMFLLCOMMODEENTERED

Purpose	AUTOSAR API service ID.
Value	0x23
Description	Definition of DCM_SERVID_COMMFLLCOMMODEENTERED.

5.2.3.2.128. DCM_SERVID_COMMFULLCOMMODEENTERED

Purpose	AUTOSAR API service ID.
Value	0x23U
Description	Definition of DCM_SERVID_COMMFULLCOMMODEENTERED.

5.2.3.2.129. DCM_SERVID_COMMNOCOMMODEENTERED

Purpose	AUTOSAR API service ID.
Value	0x21
Description	Definition of DCM_SERVID_COMMNOCOMMODEENTERED.

5.2.3.2.130. DCM_SERVID_COMMON_COMM_MODE_ENTRY

Purpose	AUTOSAR API service ID.
Value	0x2C
Description	Definition of DCM_SERVID_COMMON_COMM_MODE_ENTRY.

5.2.3.2.131. DCM_SERVID_COMMSENLCOMMODEENTERED

Purpose	AUTOSAR API service ID.
----------------	-------------------------



Value	0x22
Description	Definition of DCM_SERVID_COMMODEENTERED.

5.2.3.2.132. DCM_SERVID_DEFAULTDIDFREEZECURRENTSTATE_DEFAULTINTERFACE

Purpose	Internal API service ID for Dcm_DefaultDidFreezeCurrentState.
Value	0x43U
Description	Definition of DCM_SERVID_DEFAULTDIDFREEZECURRENTSTATE_DEFAULTINTERFACE.

5.2.3.2.133. DCM_SERVID_DEFAULTDIDREAD_DEFAULTINTERFACE

Purpose	Internal API service ID for Dcm_DefaultDidRead.
Value	0x48U
Description	Definition of DCM_SERVID_DEFAULTDIDREAD_DEFAULTINTERFACE.

5.2.3.2.134. DCM_SERVID_DEFAULTDIDRESETTODEFAULT_DEFAULTINTERFACE

Purpose	Internal API service ID for Dcm_DefaultDidResetToDefault.
Value	0x46U
Description	Definition of DCM_SERVID_DEFAULTDIDRESETTODEFAULT_DEFAULTINTERFACE.

5.2.3.2.135. DCM_SERVID_DEFAULTDIDRETURNCONTROLTOECUFORALL_DEFAULTINTERFACE

Purpose	Internal API service ID for Dcm_DefaultDidReturnControlToECUForAll.
Value	0x44U
Description	Definition of DCM_SERVID_DEFAULTDIDRETURNCONTROLTOECUFORALL_DEFAULTINTERFACE.

5.2.3.2.136. DCM_SERVID_DEFAULTDIDRETURNCONTROLTOECU_DEFAULTINTERFACE

Purpose	Internal API service ID for Dcm_DefaultDidReturnControlToECU.
----------------	---



Value	0x45U
Description	Definition of DCM_SERVID_DEFAULTDIDRETURNCONTROLTOECU_DEFAULTINTERFACE.

5.2.3.2.137. DCM_SERVID_DEFAULTDIDSHORTTERMADJUSTMENT_DEFAULTINTERFACE

Purpose	Internal API service ID for Dcm_DefaultDIDShortTermAdjustment.
Value	0x42U
Description	Definition of DCM_SERVID_DEFAULTDIDSHORTTERMADJUSTMENT_DEFAULTINTERFACE.

5.2.3.2.138. DCM_SERVID_DEFAULTDIDWRITE_DEFAULTINTERFACE

Purpose	Internal API service ID for Dcm_DefaultDIDWrite.
Value	0x47U
Description	Definition of DCM_SERVID_DEFAULTDIDWRITE_DEFAULTINTERFACE.

5.2.3.2.139. DCM_SERVID_DEMTRIGGERONDTC

Purpose	AUTOSAR API service ID.
Value	0x2B
Description	Definition of DCM_SERVID_DEMTRIGGERONDTC.

5.2.3.2.140. DCM_SERVID_GETACTIVEPROTOCOL

Purpose	AUTOSAR API service ID.
Value	0x0F
Description	Definition of DCM_SERVID_GETACTIVEPROTOCOL.

5.2.3.2.141. DCM_SERVID_GETSECURITYLEVEL

Purpose	AUTOSAR API service ID.
----------------	-------------------------



Value	0x0D
Description	Definition of DCM_SERVID_GETSECURITYLEVEL.

5.2.3.2.142. DCM_SERVID_GETSESCTRLTYPE

Purpose	AUTOSAR API service ID.
Value	0x06
Description	Definition of DCM_SERVID_GETSESCTRLTYPE.

5.2.3.2.143. DCM_SERVID_GETVERSIONINFO

Purpose	AUTOSAR API service ID.
Value	0x24
Description	Definition of DCM_SERVID_GETVERSIONINFO.

5.2.3.2.144. DCM_SERVID_INIT

Purpose	AUTOSAR API service ID.
Value	0x01
Description	Definition of DCM_SERVID_INIT.

5.2.3.2.145. DCM_SERVID_MAINFNCTION

Purpose	AUTOSAR API service ID.
Value	0x25
Description	Definition of DCM_SERVID_MAINFNCTION.

5.2.3.2.146. DCM_SERVID_MAINFUNCTION

Purpose	AUTOSAR API service ID.
Value	0x25
Description	Definition of DCM_SERVID_MAINFUNCTION.



5.2.3.2.147. DCM_SERVID_PROCESSINGDONE

Purpose	AUTOSAR API service ID.
Value	0x43
Description	Definition of DCM_SERVID_PROCESSINGDONE.

5.2.3.2.148. DCM_SERVID_RESETTODEFAULTSESSION

Purpose	AUTOSAR API service ID.
Value	0x2A
Description	Definition of DCM_SERVID_RESETTODEFAULTSESSION.

5.2.3.2.149. DCM_SERVID_SETACTIVEDIAGNOSTIC

Purpose	AUTOSAR API service ID.
Value	0x56U
Description	Definition of DCM_SERVID_SETACTIVEDIAGNOSTIC.

5.2.3.2.150. DCM_SERVID_SETNEGRESPONSE

Purpose	AUTOSAR API service ID.
Value	0x44
Description	Definition of DCM_SERVID_SETNEGRESPONSE.

5.2.3.2.151. DCM_SERVID_TRIGGERONEVENT

Purpose	AUTOSAR API service ID.
Value	0x2D
Description	Definition of DCM_SERVID_TRIGGERONEVENT.

5.2.3.2.152. DCM_SOFT_RESET

Purpose	Reset mode type is 'Software reset'.
Value	3U



5.2.3.2.153. DCM_SS_BOOTLOADER_RESET

Purpose	Jump to System Supplier Bootloader.
Value	7U

5.2.3.2.154. DCM_SUPPLIER_1

Purpose	SW Suppler Specific Protocol 1.
Value	0xF0U

5.2.3.2.155. DCM_SUPPLIER_10

Purpose	SW Suppler Specific Protocol 10.
Value	0xF9U

5.2.3.2.156. DCM_SUPPLIER_11

Purpose	SW Suppler Specific Protocol 11.
Value	0xFAU

5.2.3.2.157. DCM_SUPPLIER_12

Purpose	SW Suppler Specific Protocol 12.
Value	0xFBÜ

5.2.3.2.158. DCM_SUPPLIER_13

Purpose	SW Suppler Specific Protocol 13.
Value	0xFCU

5.2.3.2.159. DCM_SUPPLIER_14

Purpose	SW Suppler Specific Protocol 14.
Value	0xFDÜ



5.2.3.2.160. DCM_SUPPLIER_15

Purpose	SW Suppler Specific Protocol 15.
Value	0xFEU

5.2.3.2.161. DCM_SUPPLIER_2

Purpose	SW Suppler Specific Protocol 2.
Value	0xF1U

5.2.3.2.162. DCM_SUPPLIER_3

Purpose	SW Suppler Specific Protocol 3.
Value	0xF2U

5.2.3.2.163. DCM_SUPPLIER_4

Purpose	SW Suppler Specific Protocol 4.
Value	0xF3U

5.2.3.2.164. DCM_SUPPLIER_5

Purpose	SW Suppler Specific Protocol 5.
Value	0xF4U

5.2.3.2.165. DCM_SUPPLIER_6

Purpose	SW Suppler Specific Protocol 6.
Value	0xF5U

5.2.3.2.166. DCM_SUPPLIER_7

Purpose	SW Suppler Specific Protocol 7.
----------------	---------------------------------



Value	0xF6U
--------------	-------

5.2.3.2.167. DCM_SUPPLIER_8

Purpose	SW Suppler Specific Protocol 8.
Value	0xF7U

5.2.3.2.168. DCM_SUPPLIER_9

Purpose	SW Suppler Specific Protocol 9.
Value	0xF8U

5.2.3.2.169. DCM_SVCID_COPYRXDATA

Purpose	AUTOSAR API service ID.
Value	0x02
Description	Definition of DCM_SVCID_COPYRXDATA.

5.2.3.2.170. DCM_SVCID_COPYTXDATA

Purpose	AUTOSAR API service ID.
Value	0x04
Description	Definition of DCM_SVCID_COPYTXDATA.

5.2.3.2.171. DCM_SVCID_GETPROGCONDITIONS

Purpose	AUTOSAR API service ID.
Value	0x4B
Description	Definition of DCM_SVCID_GETPROGCONDITIONS.

5.2.3.2.172. DCM_SVCID_READMEMORY

Purpose	AUTOSAR API service ID.
----------------	-------------------------



Value	0x26
Description	Definition of DCM_SVCID_READMEMORY.

5.2.3.2.173. DCM_SVCID_REQUESTDOWNLOAD

Purpose	AUTOSAR API service ID.
Value	0x30
Description	Definition of DCM_SVCID_REQUESTDOWNLOAD.

5.2.3.2.174. DCM_SVCID_REQUESTUPLOAD

Purpose	AUTOSAR API service ID.
Value	0x31
Description	Definition of DCM_SVCID_REQUESTUPLOAD.

5.2.3.2.175. DCM_SVCID_REQUESTTRANSFEREXIT

Purpose	AUTOSAR API service ID.
Value	0x32
Description	Definition of DCM_SVCID_REQUESTTRANSFEREXIT.

5.2.3.2.176. DCM_SVCID_REQUESTDOWNLOAD

Purpose	AUTOSAR API service ID.
Value	0x30
Description	Definition of DCM_SVCID_REQUESTDOWNLOAD.

5.2.3.2.177. DCM_SVCID_REQUESTTRANSFEREXIT

Purpose	AUTOSAR API service ID.
Value	0x32
Description	Definition of DCM_SVCID_REQUESTTRANSFEREXIT.



5.2.3.2.178. DCM_SVCID_REQUESTUPLOAD

Purpose	AUTOSAR API service ID.
Value	0x31
Description	Definition of DCM_SVCID_REQUESTUPLOAD.

5.2.3.2.179. DCM_SVCID_SETPROGCONDITIONS

Purpose	AUTOSAR API service ID.
Value	0x4A
Description	Definition of DCM_SVCID_SETPROGCONDITIONS.

5.2.3.2.180. DCM_SVCID_STARTOFRECEPTION

Purpose	AUTOSAR API service ID.
Value	0x00
Description	Definition of DCM_SVCID_STARTOFRECEPTION.

5.2.3.2.181. DCM_SVCID_TPRXINDICATION

Purpose	AUTOSAR API service ID.
Value	0x03
Description	Definition of DCM_SVCID_TPRXINDICATION.

5.2.3.2.182. DCM_SVCID_TPTXCONFIRMATION

Purpose	AUTOSAR API service ID.
Value	0x05
Description	Definition of DCM_SVCID_TPTXCONFIRMATION.

5.2.3.2.183. DCM_SVCID_TXCONFIRMATION

Purpose	AUTOSAR API service ID.
----------------	-------------------------



Value	0x40U
Description	Definition of DCM_SVCID_TXCONFIRMATION.

5.2.3.2.184. DCM_SVCID_WRITEMEMORY

Purpose	AUTOSAR API service ID.
Value	0x27
Description	Definition of DCM_SVCID_WRITEMEMORY.

5.2.3.2.185. DCM_SW_MAJOR_VERSION

Purpose	AUTOSAR module major version.
Value	5U

5.2.3.2.186. DCM_SW_MINOR_VERSION

Purpose	AUTOSAR module minor version.
Value	0U

5.2.3.2.187. DCM_SW_PATCH_VERSION

Purpose	AUTOSAR module patch version.
Value	10U

5.2.3.2.188. DCM_TIMEOUT_RCRRP

Purpose	
Value	0X02U

5.2.3.2.189. DCM_UDS_ON_CAN

Purpose	UDS Protocol on CAN (ISO15765-3; ISO14229-1).
----------------	---



Value	0x03U
--------------	-------

5.2.3.2.190. DCM_UDS_ON_FLEXRAY

Purpose	UDS Protocol on FlexRay (Manufacturer specific; ISO14229-1).
Value	0x04U

5.2.3.2.191. DCM_UDS_ON_IP

Purpose	UDS on Internet Protocol (Manufacturer specific; ISO14229-1).
Value	0x05U

5.2.3.2.192. DCM_UDS_RDBPI_TRANSMISSIONMODE_FASTRATE

Purpose	This value specifies that the server shall transmit the requested dataRecord information at a fast rate in response to the request message.
Value	(0x03U)

5.2.3.2.193. DCM_UDS_RDBPI_TRANSMISSIONMODE_MEDIUMRATE

Purpose	This value specifies that the server shall transmit the requested dataRecord information at a medium rate in response to the request message.
Value	(0x02U)

5.2.3.2.194. DCM_UDS_RDBPI_TRANSMISSIONMODE_SLOWRATE

Purpose	This value specifies that the server shall transmit the requested dataRecord information at a slow rate in response to the request message.
Value	(0x01U)

5.2.3.2.195. DCM_UDS_RDBPI_TRANSMISSIONMODE_STOPSENDING

Purpose	The server stops transmitting positive response messages send periodically/repeatedly.
----------------	--



Value	(0x04U)
--------------	---------

5.2.3.2.196. DCM_UPDATE_PAGE

Purpose	Requests to update the next page in paged buffer processing.
Value	0x04U

5.2.3.2.197. DCM_VENDOR_ID

Purpose	AUTOSAR vendor identification: Elektrobit Automotive GmbH.
Value	1U

5.2.3.2.198. DCM_WARM_START

Purpose	The ECU starts from a bootloader jump.
Value	1U

5.2.3.2.199. DCM_WRITE_FAILED

Purpose	Write failed for the service 0x36.
Value	0x02U

5.2.3.2.200. DCM_WRITE_FORCE_RCRRP

Purpose	Application request the transmission of a Response Pending (NRC 0x78) (used by API: Dcm_WriteMemory()).
Value	0x03U

5.2.3.2.201. DCM_WRITE_OK

Purpose	Write is done for the service 0x36.
Value	0x00U



5.2.3.2.202. DCM_WRITE_PENDING

Purpose	Write is pending for the service 0x36.
Value	0x01U

5.2.3.3. Functions

5.2.3.3.1. Dcm_APICaller_GenericSvchImplementation

Purpose	Caller of internally implemented SvcH.	
Synopsis	<code>Std_ReturnType Dcm_APICaller_GenericSvchImplementation (Dcm_-OpStatusType OpStatus , Dcm_MsgContextType * pMsgContext);</code>	
Service ID	DCM_API_CALLER_GENERIC_SVCH_IMPLEMENTATION_ID	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	OpStatus	Operation state passed by External Service Handler to internal SvcH
Parameters (in,out)	pMsgContext	Contextual information message passed to a diagnostic protocol
Return Value	code of API	
	return_code_of_internally_implemented_SvcH	if internal SvcH exists
	DCM_E_SERVICENOTSUPPORTED	if internal SvcH does not exists
Description	This API provides a correct context for the internally implemented SvcH to process the request.	

5.2.3.3.2. Dcm_ApplicationTransmissionConfirmation

Purpose	Confirm (un)successful transmission.
Synopsis	<code>void Dcm_ApplicationTransmissionConfirmation (Dcm_ServiceIDType ServiceID , Dcm_MsgContextPtrType MsgContext , Dcm_ResponseConfirmationType ResponseConfirmationType , boolean ResponseSuppressed , boolean TransmissionSuccessful);</code>
Service ID	DCM_SERVID_APPLICATIONTRANSMISSIONCONFIRMATION
Sync/Async	Synchronous



Reentrancy	Non reentrant	
Parameters (in)	ServiceID	Service ID of diagnostic request which triggered the response transmission.
	MsgContext	Pointer to the MessageContext of the service handler which triggered the response transmission.
	ResponseConfirmationType	The type of response which was confirmed.
	ResponseSuppressed	TRUE if the response was suppressed (either as a result of the suppressPositiveResponse bit being set or otherwise). FALSE otherwise.
	TransmissionSuccessful	TRUE if the confirmation of transmission was positive. FALSE otherwise.
Description	This function is provided by the upper layer (application).	

5.2.3.3.3. Dcm_BootloaderResetModeEntry

Purpose	Notification of Reset Mode entered.
Synopsis	Std_ReturnType Dcm_BootloaderResetModeEntry (void) ;
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'BootloaderReset' request received from tester

5.2.3.3.4. Dcm_CopyRxData

Purpose	Called once upon reception of each segment. Within this call, the received data is copied from the receive TP buffer to the DCM receive buffer.
Synopsis	BufReq_ReturnType Dcm_CopyRxData (PduIdType DcmRxPduId , const PduInfoType * PduInfoPtr , PduLengthType * RxBufferSizePtr) ;
Service ID	DCM_SVCID_COPYRXDATA
Sync/Async	Synchronous



Reentrancy	Reentrant for different Pdulds. Non reentrant for the same Pduld.	
Parameters (in)	DcmRxPduId	Identifies the DCM data to be received. This information is used within the DCM to distinguish two or more receptions at the same time.
	PduInfoPtr	Pointer to a PduInfoType which indicates the number of bytes to be copied (SduLength) and the location of the source data (SduDataPtr).
Parameters (out)	RxBufferSizePtr	pointer to Remaining free place in receive buffer after completion of this call.
Return Value	Request data copy status	
	BUFREQ_OK	Data has been copied to the receive buffer completely as requested.
	BUFREQ_E_NOT_OK	Data has not been copied. Request failed.

5.2.3.3.5. Dcm_CopyTxData

Purpose	Request transmission buffer.	
Synopsis	<pre>BufReq_ReturnType Dcm_CopyTxData (PduIdType DcmTxPduId , PduInfoType * PduInfoPtr , RetryInfoType * RetryInfoPtr , Pdu- LengthType * TxDataCntPtr);</pre>	
Service ID	DCM_SVCID_COPYTXDATA	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Pdulds. Non reentrant for the same Pduld.	
Parameters (in)	DcmTxPduId	Identifies the DCM data to be sent. This information is used to derive the PCI information within the transport protocol. The value has to be same as in the according service call PduR_DcmTransmit().
	PduInfoPtr	Pointer to a PduInfoType, which indicates the number of bytes to be copied (SduLength) and the location where the data have to be copied to (SduDataPtr). An SduLength of 0 is possible in order to poll the available transmit data count. In this case no data are to be copied and SduDataPtr might be invalid.



	RetryInfoPtr	If the transmitted TP I-PDU does not support the retry feature a NULL_PTR can be provided. This indicates that the copied transmit data can be removed from the buffer after it has been copied.
Parameters (out)	TxDataCntPtr	Remaining Tx data after completion of this call
Return Value	Response data copy status	
	BUFREQ_OK	Data has been copied to the transmit buffer completely as requested.
	BUFREQ_E_NOT_OK	Data has not been copied. Request failed, in case the corresponding I-PDU was stopped.
	BUFREQ_E_BUSY	Request could not be fulfilled, because the required amount of Tx data is not available. The lower layer module may retry this call later on. No data has been copied.
Description	At invocation of Dcm_CopyTxData the DCM module copies the requested transmit data with ID Pduld from its internal transmit buffer to the location specified by the PduInfoPtr. The function Dcm_CopyTxData also calculates and sets the TxDataCntPtr to the amount of remaining bytes for the transmission of this data. If RetryInfoPtr is NULL_PTR or if TpDataState is not equal TP_DATARETRY, the Dcm shall always copy the next fragment of data to the SduDataPtr. If TpDataState equals TP_DATARETRY, the Dcm shall copy previously copied data again, beginning from the offset position given in RetryInfoPtr->TxTpDataCnt from the current position.	

5.2.3.3.6. Dcm_DefaultDIDFreezeCurrentState

Purpose	Function called on reception of UDS Service InputOutputControlByIdentifier (0x2F) with InputOutputControlParameter equal to freezeCurrentState, when DcmDspRoutineEnableDefaultInterfaces configuration parameter is set to TRUE.
Synopsis	Std_ReturnType Dcm_DefaultDIDFreezeCurrentState (Dcm_OpStatusType OpStatus , Dcm_DIDIDType DID , Dcm_NegativeResponseCodeType * Nrc);
Service ID	DCM_SERVID_DEFAULTDIDFREEZECURRENTSTATE_DEFAULTINTERFACE
Sync/Async	Asynchronous
Reentrancy	Non-Reentrant



Parameters (in)	OpStatus	Operation Status: DCM_INITIAL - Initial call. DCM_PENDING - Subsequent call after a previous DCM_E_PENDING return. DCM_CANCEL - Final call for cancellation of operation. The Dcm ignores the return value for this call.
	DID	2-byte DID identifier.
Parameters (out)	Nrc	Pointer to which the application shall write the NRC to be returned or whether the DID is supported or not. Valid only if the return value is E_NOT_OK.
Return Value	Permission of application <ul style="list-style-type: none"> ▶ E_OK: Request was successful. ▶ E_NOT_OK: Request was not successful. ▶ DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish. 	
Description	<p>Default operation for freezing unconfigured DIDs to current control state. Unconfigured DIDs shall be considered as being composed of a single asynchronously-handled signal. The Application may perform the operation for the requested DataIdentifier (DID) or return an adequate Negative Response Code.</p> <p>This callout allows the user to define the functionality of the freezeCurrentState sub-service.</p>	

5.2.3.3.7. [Dcm_DefaultDIDRead](#)

Purpose	Function called on reception of UDS Service ReadDataByIdentifier (0x22) for an unsupported DID when DcmDspRoutineEnableDefaultInterfaces configuration parameter is set to TRUE.	
Synopsis	<pre>Std_ReturnType Dcm_DefaultDIDRead (Dcm_OpStatusType OpStatus , Dcm_DIDIDType DID , uint8 * DestinationBuffer , uint32 * Length , Dcm_NegativeResponseCodeType * Nrc);</pre>	
Service ID	DCM_SERVID_DEFAULTDIDREAD_DEFAULTINTERFACE	
Sync/Async	Asynchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	OpStatus	Operation Status: DCM_INITIAL - Initial call. DCM_PENDING - Subsequent call after a previous DCM_E_PENDING re-



		turn. DCM_CANCEL - Final call for cancellation of operation. The Dcm ignores the return value for this call.
	DID	2-byte DID identifier.
Parameters (out)	DestinationBuffer	Pointer to the location in the output buffer where DID data is written.
	Length	pointer to which the application shall write the amount of data, in bytes, which was written to the DestinationBuffer pointer.
	Nrc	pointer to which the application shall write the NRC to be returned or whether the DID is supported or not. Valid only if the return value is E_NOT_OK.
Return Value	Permission of application <ul style="list-style-type: none"> ▶ E_OK: Request was successful. ▶ E_NOT_OK: Request was not successful. ▶ DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish. 	
Description	Default operation for reading unconfigured DIDs. Unconfigured DIDs shall be considered as being composed of a single asynchronously-handled signal. The Application may perform the operation for the requested DID or return an adequate NegativeResponseCode. For ReadDataByIdentifier (0x22) requests comprising more than one DID, returning the RequestOutOfRange (0x31) NRC shall advance reading to the next DID if existing, as with regular DIDs. This callout allows the user to define the functionality of the ReadDataByIdentifier (0x22)	

5.2.3.3.8. [Dcm_DefaultDIDResetToDefault](#)

Purpose	Function called on reception of UDS Service InputOutputControlByIdentifier (0x2F) with InputOutputControlParameter equal to resetToDefault, when DcmDspRoutineEnableDefaultInterfaces configuration parameter is set to TRUE.
Synopsis	Std_ReturnType Dcm_DefaultDIDResetToDefault (Dcm_OpStatusType OpStatus , Dcm_DIDIDType DID , Dcm_NegativeResponseCodeType * Nrc);
Service ID	DCM_SERVID_DEFAULTDIDRESETTODEFAULT_DEFAULTINTERFACE
Sync/Async	Asynchronous



Reentrancy	Non-Reentrant	
Parameters (in)	OpStatus	Operation Status: DCM_INITIAL - Initial call. DCM_PENDING - Subsequent call after a previous DCM_E_PENDING return. DCM_CANCEL - Final call for cancellation of operation. The Dcm ignores the return value for this call.
	DID	2-byte DID identifier.
Parameters (out)	Nrc	Pointer to which the application shall write the NRC to be returned or whether the DID is supported or not. Valid only if the return value is E_NOT_OK.
Return Value	Permission of application <ul style="list-style-type: none"> ▶ E_OK: Request was successful. ▶ E_NOT_OK: Request was not successful. ▶ DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish. 	
Description	Default operation for resetting unconfigured DIDs to default control. Unconfigured DIDs shall be considered as being composed of a single asynchronously-handled signal. The Application may perform the operation for the requested DID or return an adequate NegativeResponseCode. This callout allows the user to define the functionality of the resetToDefault sub-service.	

5.2.3.3.9. [Dcm_DefaultDIDReturnControlToECU](#)

Purpose	Function called on reception of UDS Service InputOutputControlByIdentifier (0x2F) with InputOutputControlParameter equal to returnControlToECU, when DcmDspRoutineEnableDefaultInterfaces configuration parameter is set to TRUE.	
Synopsis	<pre>Std_ReturnType Dcm_DefaultDIDReturnControlToECU (Dcm_OpStatusType OpStatus , Dcm_DIDIDType DID , Dcm_NegativeResponseCodeType * Nrc);</pre>	
Service ID	DCM_SERVID_DEFAULTDIDRETURNCONTROLTOECU_DEFAULTINTERFACE	
Sync/Async	Asynchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	OpStatus	Operation Status: DCM_INITIAL - Initial call. DCM_PENDING - Subsequent call



		after a previous DCM_E_PENDING return. DCM_CANCEL - Final call for cancellation of operation. The Dcm ignores the return value for this call.
	DID	2-byte DID identifier.
Parameters (out)	Nrc	Pointer to which the application shall write the NRC to be returned or whether the DID is supported or not. Valid only if the return value is E_NOT_OK.
Return Value	Permission of application <ul style="list-style-type: none"> ▶ E_OK: Request was successful. ▶ E_NOT_OK: Request was not successful. ▶ DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish. 	
Description	<p>This callout allows the user to define the functionality of the returnControlToECU sub-service.</p> <p>Default operation for returning unconfigured DIDs to ECU control. Unconfigured DIDs shall be considered as being composed of a single asynchronously-handled signal. The Application may perform the operation for the requested DID or return an adequate NegativeResponseCode.</p>	

5.2.3.3.10. Dcm_DefaultDIDReturnControlToECUForAll

Purpose	Return Control to ECU for All Input Output Controls.	
Synopsis	<pre>Std_ReturnType Dcm_DefaultDIDReturnControlToECUForAll (Dcm_- OpStatusType OpStatus);</pre>	
Service ID	DCM_SERVID_DEFAULTDIDRETURNCONTROLTOECUFORALL_DEFAULTINTER- FACE	
Sync/Async	Asynchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	OpStatus	Operation Status: DCM_INITIAL - Initial call. DCM_PENDING - Subsequent call after a previous DCM_E_PENDING return. DCM_CANCEL - Final call for cancellation of operation. The Dcm ignores the return value for this call.
Return Value	Permission of application	



	<ul style="list-style-type: none"> ▶ E_OK: Request was successful. ▶ E_NOT_OK: Request was not successful. ▶ <u>DCM_E_PENDING</u>: Request is not yet finished. Further call(s) required to finish.
Description	<p>This callout allows the user to define the functionality of the ReturnControlToECU-ForAll functionality.</p> <p>Default operation for returning all unconfigured DIDs to ECU control. To be used for the ReturnControlToECU operation at Session switch to DefaultSession since the Dcm cannot decide which unconfigured DIDs are under control.</p>

5.2.3.3.11. Dcm_DefaultDIDShortTermAdjustment

Purpose	Function called on reception of UDS Service InputOutputControlByIdentifier (0x2F) with InputOutputControlParameter equal to shortTermAdjustment, when DcmDspRoutineEnableDefaultInterfaces configuration parameter is set to TRUE.	
Synopsis	<pre>Std_ReturnType Dcm_DefaultDIDShortTermAdjustment (Dcm_OpStatus OpStatus , Dcm_DIDIDType DID , uint8 * controlState , uint32 Length , Dcm_NegativeResponseCodeType * Nrc);</pre>	
Service ID	<u>DCM_SERVID_DEFAULTDIDSHORTTERMADJUSTMENT_DEFAULTINTERFACE</u>	
Sync/Async	Asynchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	OpStatus DID Length	Operation Status: DCM_INITIAL - Initial call. DCM_PENDING - Subsequent call after a previous DCM_E_PENDING return. DCM_CANCEL - Final call for cancellation of operation. The Dcm ignores the return value for this call. 2-byte DID identifier. - the valid length, in bytes, of the controlState. Represents the length, in bytes, of the received controlOptionRecord and controlMask from the InputOutputControlByIdentifier (0x2F) service request.
Parameters (out)	controlState	Pointer to the location in the input buffer where the controlState (containing the controlOptionRecord and controlMask) is located.



	Nrc	Pointer to which the application shall write the NRC to be returned or whether the DID is supported or not. Valid only if the return value is E_NOT_OK.
Return Value	<p>Permission of application</p> <ul style="list-style-type: none"> ➤ E_OK: Request was successful. ➤ E_NOT_OK: Request was not successful. ➤ DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish. 	
Description	<p>This callout allows the user to define the functionality of the shortTermAdjustment sub-service.</p> <p>Default operation for short term adjustment of control state of unconfigured DIDs. Unconfigured DIDs shall be considered as being composed of a single asynchronous-handled signal. The Application may perform the operation for the requested DID or return an adequate NegativeResponseCode.</p>	

5.2.3.3.12. **Dcm_DefaultDIDWrite**

Purpose	Function called on reception of UDS Service WriteDataByIdentifier (0x2E) for an unsupported DID when DcmDspRoutineEnableDefaultInterfaces configuration parameter is set to TRUE.	
Synopsis	<pre>Std_ReturnType Dcm_DefaultDIDWrite (Dcm_OpStatusType OpStatus , Dcm_DIDIDType DID , uint8 * SourceBuffer , uint32 Length , Dcm_NegativeResponseCodeType * Nrc);</pre>	
Service ID	DCM_SERVID_DEFAULTDIDWRITE_DEFAULTINTERFACE	
Sync/Async	Asynchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	OpStatus	Operation Status: DCM_INITIAL - Initial call. DCM_PENDING - Subsequent call after a previous DCM_E_PENDING return. DCM_CANCEL - Final call for cancellation of operation. The Dcm ignores the return value for this call.
	DID	2-byte DID identifier.
	Length	The valid length, in bytes, of the Source-Buffer. Represents the length, in bytes, of



		the received dataRecord from the WriteDataByIdentifier (0x2E) service request.
Parameters (out)	SourceBuffer	Pointer to the location in the input buffer where DID data is read from.
	Nrc	Pointer to which the application shall write the NRC to be returned or whether the DID is supported or not. Valid only if the return value is E_NOT_OK.
Return Value	Permission of application <ul style="list-style-type: none"> ▶ E_OK: Request was successful. ▶ E_NOT_OK: Request was not successful. ▶ DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish. 	
Description	Default operation for writing unconfigured DIDs. Unconfigured DIDs shall be considered as being composed of a single asynchronously-handled signal. The Application may perform the operation for the requested DID or return an adequate NegativeResponseCode. This callout allows the user to define the functionality of the WriteDataByIdentifier (0x2E)	

5.2.3.3.13. Dcm_DemTriggerOnDTCStatus

Purpose	Allows Dem to notify Dcm on status change of a DTC (for service RoE).	
Synopsis	<pre>Std_ReturnType Dcm_DemTriggerOnDTCStatus (uint32 DTC , uint8 DTCStatusOld , uint8 DTCStatusNew);</pre>	
Service ID	DCM_SERVID_DEMTRIGGERONDTC	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	DTC	Identifier of the DTC
	DTCStatusOld	Status of the DTC before the change
	DTCStatusNew	Status of the DTC after the change
Return Value	Always return E_OK.	
Description	When this function is called, Dcm verifies the DTC against the status mask in the diagnostic request and if the verification is positive, executes the associated service (serviceToRespondTo).	



5.2.3.3.14. Dcm_DisableRapidPowerShutdownResetModeEntry

Purpose	Notification of Reset Mode entered.
Synopsis	Std_ReturnType Dcm_DisableRapidPowerShutdownResetModeEntry (void);
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'DisableRapidPowerShutdownReset' request received from tester

5.2.3.3.15. Dcm_DisableRxEnableTxNmModeEntry

Purpose	Notification of Communication Mode entered.
Synopsis	Std_ReturnType Dcm_DisableRxEnableTxNmModeEntry (void);
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'DisableRxEnableTx' request for networkManagementCommunicationMessages received from tester

5.2.3.3.16. Dcm_DisableRxEnableTxNormModeEntry

Purpose	Notification of Communication Mode entered.
Synopsis	Std_ReturnType Dcm_DisableRxEnableTxNormModeEntry (void);
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'DisableRxEnableTx' request for normalCommunicationMessages received from tester



5.2.3.3.17. Dcm_DisableRxEnableTxNormNmModeEntry

Purpose	Notification of Communication Mode entered.
Synopsis	Std_ReturnType Dcm_DisableRxEnableTxNormNmModeEntry (void);
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'DisableRxEnableTx' request for networkManagementCommunicationMessages and normalCommunicationMessages received from tester

5.2.3.3.18. Dcm_DisableRxTxNmModeEntry

Purpose	Notification of Communication Mode entered.
Synopsis	Std_ReturnType Dcm_DisableRxTxNmModeEntry (void);
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'DisableRxTx' request for networkManagement-CommunicationMessages received from tester

5.2.3.3.19. Dcm_DisableRxTxNormModeEntry

Purpose	Notification of Communication Mode entered.
Synopsis	Std_ReturnType Dcm_DisableRxTxNormModeEntry (void);
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'DisableRxTx' request for normalCommunication-Messages received from tester



5.2.3.3.20. Dcm_DisableRxTxNormNmModeEntry

Purpose	Notification of Communication Mode entered.
Synopsis	Std_ReturnType Dcm_DisableRxTxNormNmModeEntry (void);
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'DisableRxTx' request for networkManagement-CommunicationMessages and normalCommunicationMessages received from tester

5.2.3.3.21. Dcm_EnableRapidPowerShutdownResetModeEntry

Purpose	Notification of Reset Mode entered.
Synopsis	Std_ReturnType Dcm_EnableRapidPowerShutdownResetModeEntry (void);
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'EnableRapidPowerShutdownReset' request received from tester

5.2.3.3.22. Dcm_EnableRxDisableTxNmModeEntry

Purpose	Notification of Communication Mode entered.
Synopsis	Std_ReturnType Dcm_EnableRxDisableTxNmModeEntry (void);
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'EnableRxDisableTx' request for networkManagementCommunicationMessages received from tester



5.2.3.3.23. Dcm_EnableRxDisableTxNormModeEntry

Purpose	Notification of Communication Mode entered.
Synopsis	Std_ReturnType Dcm_EnableRxDisableTxNormModeEntry (void);
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for an 'EnableRxDisableTx' request for normalCommunicationMessages received from tester

5.2.3.3.24. Dcm_EnableRxDisableTxNormNmModeEntry

Purpose	Notification of Communication Mode entered.
Synopsis	Std_ReturnType Dcm_EnableRxDisableTxNormNmModeEntry (void);
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'EnableRxDisableTx' request for networkManagementCommunicationMessages and normalCommunicationMessages received from tester

5.2.3.3.25. Dcm_EnableRxTxNmModeEntry

Purpose	Notification of Communication Mode entered.
Synopsis	Std_ReturnType Dcm_EnableRxTxNmModeEntry (void);
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'EnableRxTx' request for networkManagementCommunicationMessages received from tester



5.2.3.3.26. Dcm_EnableRxTxNormModeEntry

Purpose	Notification of Communication Mode entered.
Synopsis	Std_ReturnType Dcm_EnableRxTxNormModeEntry (void);
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for an 'EnableRxTx' request for normalCommunication-Messages received from tester

5.2.3.3.27. Dcm_EnableRxTxNormNmModeEntry

Purpose	Notification of Communication Mode entered.
Synopsis	Std_ReturnType Dcm_EnableRxTxNormNmModeEntry (void);
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'EnableRxTx' request for networkManagementCommunicationMessages and normalCommunicationMessages received from tester

5.2.3.3.28. Dcm_ExternalProcessingDone

Purpose	Send a response.	
Synopsis	void Dcm_ExternalProcessingDone (Dcm_MsgContextType * pMsgContext);	
Service ID	DCM_SERVID_PROCESSINGDONE	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	pMsgContext	Message-related information for one one diagnostic protocol identifier.
Description	When this function is called, a response will be sent based on the data contained in pMsgContext. If Dcm_ExternalSetNegResponse() was called before, a negative re-	



sponse is sent, otherwise a positive response will be sent. The application does not have to care about any timing requirement to process a request.

For diagnostic experts: Between the arrival of a request (XXX_-Dcm<DiagnosticService>) and finishing the corresponding response (Dcm_Processing_Done), busy-acknowledges (negative response with response code 0x78) are sent automatically.

5.2.3.3.29. Dcm_ExternalSetNegResponse

Purpose	Send a negative response instead of a positive response. The negative response code is taken from the first call - duplicate calls are allowed but the errorCode transmitted is ignored. The transmission of the negative response will be performed after emitting PROC_DONE.	
Synopsis	<pre>void Dcm_ExternalSetNegResponse (Dcm_MsgContextType * pMsgContext , Dcm_NegativeResponseCodeType errorCode);</pre>	
Service ID	DCM_SERVID_SETNEGRESPONSE	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	pMsgContext	Message-related information for one diagnostic protocol identifier.
	errorCode	Error code to be set. This error code will be taken as negative response code (NRC) (see description).(DCM_E_GENERALREJECT , DCM_E_SERVICENOTSUPPORTED , DCM_E_SUBFUNCTIONNOTSUPPORTED , DCM_E_REQUESTCORRECTLYRECEIVEDRESPONSEPENDING , ...)

5.2.3.3.30. Dcm_GetActiveProtocol

Purpose	Provides active protocol.
Synopsis	<pre>Std_ReturnType Dcm_GetActiveProtocol (Dcm_ProtocolType * ActiveProtocol);</pre>
Service ID	DCM_SERVID_GETACTIVEPROTOCOL
Sync/Async	Synchronous



Reentrancy	Reentrant	
Parameters (out)	ActiveProtocol	Active diagnostic protocol (DCM_UDS_ON_CAN , DCM_OBD_ON_CAN , ...)
Return Value	<ul style="list-style-type: none"> ▶ E_OK - valid call of this function ▶ E_NOT_OK - if Det is enabled and a Det check fails. 	
Description	This Dcm function returns the active protocol.	

5.2.3.3.31. Dcm_GetProgConditions

Purpose	Callout for retrieving the stored program conditions.	
Synopsis	<code>Dcm_EcuStartModeType Dcm_GetProgConditions (Dcm_ProgConditionsType * ProgConditions);</code>	
Service ID	DCM_SVCID_GETPROGCONDITIONS	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (out)	ProgConditions	Pointer to the structure which holds the program conditions
Return Value	DCM_COLD_START or DCM_WARM_START.	
Description	This callout allows the user to retrieve the stored programming environment from the non-volatile memory after jumping from bootloader.	

5.2.3.3.32. Dcm_GetSecurityLevel

Purpose	Provides the active security level value.	
Synopsis	<code>Std_ReturnType Dcm_GetSecurityLevel (Dcm_SecLevelType * SecLevel);</code>	
Service ID	DCM_SERVID_GETSECURITYLEVEL	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	SecLevel	Active Security Level value Conversion formula to calculate SecurityLevel out of tester requested SecurityAccessType parameter: SecurityLevel = (SecurityAccessType + 1) / 2 Content of SecurityAccessType is according to "securityAc-



	cessType" parameter of SecurityAccess request
Return Value	<ul style="list-style-type: none"> ▶ E_OK - valid security level available ▶ E_NOT_OK - if Det is enabled and a Det check fails
Description	This function provides the active security level value.

5.2.3.33. Dcm_GetSesCtrlType

Purpose	Provides the active session control type value.	
Synopsis	Std_ReturnType Dcm_GetSesCtrlType (Dcm_SesCtrlType * SesCtrlType);	
Service ID	DCM_SERVID_GETSESCRTLTYPE	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	SesCtrlType	Active session control type value content is according to "diagnosticSessionType" parameter of DiagnosticSessionControl request
Return Value	<ul style="list-style-type: none"> ▶ E_OK - valid call of this function ▶ E_NOT_OK - if Det is enabled and a Det check fails 	
Description	This function provides the active session control type value.	

5.2.3.34. Dcm_GetSesCtrlTypeExt

Purpose	Provides the active session control type value.	
Synopsis	Std_ReturnType Dcm_GetSesCtrlTypeExt (Dcm_SesCtrlType * SesCtrlType);	
Service ID	DCM_SERVID_GETSESCRTLTYPE	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	SesCtrlType	Active session control type value content is according to "diagnosticSessionType" parameter of DiagnosticSessionControl request



Return Value	<ul style="list-style-type: none"> ▶ E_OK - valid call of this function ▶ E_NOT_OK - if Det is enabled and a Det check fails
Description	This function provides the active session control type value and it is visible all the time.

5.2.3.35. Dcm_GetVersionInfo

Purpose	This function returns the version information of this module.	
Synopsis	<code>void Dcm_GetVersionInfo (Std_VersionInfoType * versioninfo);</code>	
Service ID	DCM_SERVID_GETVERSIONINFO	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (out)	versioninfo	Pointer to location where version info is written to

5.2.3.36. Dcm_HardResetModeEntry

Purpose	Notification of Reset Mode entered.	
Synopsis	<code>Std_ReturnType Dcm_HardResetModeEntry (void);</code>	
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Return Value	Always return E_OK.	
Description	This provides a confirmation for a 'HardReset' request received from tester	

5.2.3.37. Dcm_Init

Purpose	Initialize Dcm Service for basic initialization of Dcm module. The initialization function will initialize all Dcm global variables with the values of the configuration.	
Synopsis	<code>void Dcm_Init (const Dcm_ConfigType * ConfigPtr);</code>	
Service ID	DCM_SERVID_INIT	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	



Parameters (in)	ConfigPtr	Pointer to configuration set (unused).
Description	The Dcm supports only one configuration variant. Therefore the ConfigPtr is unused.	

5.2.3.3.38. Dcm.InjectDiagnosticRequest

Purpose	Injector Interface function to CDD.	
Synopsis	<pre>BufReq_ReturnType Dcm.InjectDiagnosticRequest (PduIdType RxPduID , const uint8 * RequestData , PduLengthType RequestLength) ;</pre>	
Parameters (in)	RxPduID	Reception Pdu Identifier of the PDU where the request is intended for processing.
	RequestData	Pointer to data array where the request intended for processing is stored.
	RequestLength	Length of the request intended for processing.
Return Value	Result of the injection attempt.	
	BUFREQ_OK	The injection has been accepted and will be processed.
	BUFREQ_E_BUSY	A request for injection is already ongoing.
	BUFREQ_E_OVFL	The request is too long and cannot be processed.
	BUFREQ_E_NOT_OK	The provided RxPduID is invalid.
Description	This API allows the caller to directly inject a Service request for DCM processing.	

5.2.3.3.39. Dcm.KeyOnOffResetModeEntry

Purpose	Notification of Reset Mode entered.
Synopsis	<pre>Std_ReturnType Dcm.KeyOnOffResetModeEntry (void) ;</pre>
Service ID	DCM_SERVID_COMMON_COMM_MODE_ENTRY
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'KeyOnOffReset' request received from tester



5.2.3.3.40. Dcm_MainFunction

Purpose	Main loop within the DSD part of the DCM module.
Synopsis	<code>void Dcm_MainFunction (void);</code>
Service ID	DCM_SERVID_MAINFUNCTION
Sync/Async	Asynchronous
Reentrancy	Non-Reentrant
Description	This function is used for processing the tasks of the main loop.

5.2.3.3.41. Dcm_ProcessRequestDownload

Purpose	Callout for RequestDownload.	
Synopsis	<code>Std_ReturnType Dcm_ProcessRequestDownload (Dcm_OpStatusType OpStatus , uint8 DataFormatIdentifier , uint32 MemoryAddress , uint32 MemorySize , uint32 BlockLength , Dcm_NegativeResponseCodeType * ErrorCode);</code>	
Service ID	DCM_SVCID_REQUESTDOWNLOAD	
Sync/Async	Asynchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	OpStatus	Operation status
	DataFormatIdentifier	Format identifier from the request
	MemoryAddress	Memory address to download
	MemorySize	Size of memory to be downloaded
	BlockLength	Maximum number of bytes for a write operation
Parameters (out)	ErrorCode	NRC if the request for download fails.
Return Value	<code>E_OK, E_NOT_OK or DCM_E_PENDING.</code>	
Description	This callout allows the user to implement the negotiation of a data transfer from the tester to the ECU.	

5.2.3.3.42. Dcm_ProcessRequestTransferExit

Purpose	Callout for termination of an upload or download process.
----------------	---



Synopsis	<code>Std_ReturnType Dcm_ProcessRequestTransferExit (Dcm_OpStatusType OpStatus , uint8 * ParameterRecord , uint32 ParameterRecordSize , Dcm_NegativeResponseCodeType * ErrorCode);</code>	
Service ID	DCM_SVCID_REQUESTTRANSFEREXIT	
Sync/Async	Asynchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	OpStatus	Operation status
	ParameterRecord	Pointer to vehicle/manufacturer specific data (optional)
	ParameterRecordSize	Length of ParameterRecord in bytes (optional)
Parameters (out)	ErrorCode	NRC if the request for termination fails.
Return Value	E_OK, E_NOT_OK or DCM_E_PENDING.	
Description	This callout allows the user to terminate the data transfer between the ECU and the tester.	

5.2.3.3.43. Dcm_ProcessRequestUpload

Purpose	Callout for RequestUpload.	
Synopsis	<code>Std_ReturnType Dcm_ProcessRequestUpload (Dcm_OpStatusType OpStatus , uint8 DataFormatIdentifier , uint32 MemoryAddress , uint32 MemorySize , Dcm_NegativeResponseCodeType * ErrorCode);</code>	
Service ID	DCM_SVCID_REQUESTUPLOAD	
Sync/Async	Asynchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	OpStatus	Operation status
	DataFormatIdentifier	Format identifier from the request
	MemoryAddress	Memory address to upload
	MemorySize	Size of memory to be uploaded
Parameters (out)	ErrorCode	NRC if the request for upload fails.
Return Value	E_OK, E_NOT_OK or DCM_E_PENDING.	
Description	This callout allows the user to implement the negotiation of a data transfer from the ECU to the tester.	



5.2.3.3.44. Dcm_ReadMemory

Purpose	Callout for upload of data (used for service TransferData).	
Synopsis	<code>Dcm_ReturnReadMemoryType Dcm_ReadMemory (Dcm_OpStatusType OpStatus , uint8 MemoryIdentifier , uint32 MemoryAddress , uint32 MemorySize , uint8 * MemoryData);</code>	
Service ID	DCM_SVCID_READMEMORY	
Sync/Async	Asynchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	OpStatus	Operation status
	MemoryIdentifier	Identifier of the memory device (optional)
	MemoryAddress	Start address of the memory from which data is to be retrieved
	MemorySize	Number of bytes to be read
Parameters (out)	MemoryData	Pointer to the buffer in Dcm
Return Value	DCM_READ_OK, DCM_READ_PENDING or DCM_READ_FAILED.	
Description	This callout allows the user to transfer data from the ECU to the tester, after successful execution of a RequestUpload service.	

5.2.3.3.45. Dcm_ResetToDefaultSession

Purpose	Provides the ResetToDefaultSession interface for user.
Synopsis	<code>Std_ReturnType Dcm_ResetToDefaultSession (void);</code>
Service ID	DCM_SERVID_RESETTODEFAULTSESSION
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Return Value	<ul style="list-style-type: none"> ▶ E_OK - valid call of this function ▶ E_NOT_OK - if Det is enabled and a Det check fails
Description	When this function is called current session will be switched to default session. If current session is default session, E_OK is returned without any session change. BswM is not informed of the transition to default session.

5.2.3.3.46. Dcm_RxIndication

Purpose	This function is only a stub to provide the function symbol.
----------------	--



Synopsis	<code>void Dcm_RxIndication (PduIdType DcmRxPduId , PduInfoType * PduInfoPtr);</code>
-----------------	---

5.2.3.3.47. Dcm_SetProgConditions

Purpose	Callout for storing the program conditions.	
Synopsis	<code>Std_ReturnType Dcm_SetProgConditions (Dcm_ProgConditionsType * ProgConditions);</code>	
Service ID	<u>DCM_SVCID_SETPROGCONDITIONS</u>	
Sync/Async	Asynchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	ProgConditions	Pointer to the structure which holds the program conditions
Return Value	E_OK, E_NOT_OK or DCM_E_PENDING.	
Description	This callout allows the user to store the programming environment in a non-volatile memory before jumping to bootloader.	

5.2.3.3.48. Dcm_SoftResetModeEntry

Purpose	Notification of Reset Mode entered.	
Synopsis	<code>Std_ReturnType Dcm_SoftResetModeEntry (void);</code>	
Service ID	<u>DCM_SERVID_COMMON_COMM_MODE_ENTRY</u>	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Return Value	Always return E_OK.	
Description	This provides a confirmation for a 'SoftReset' request received from tester	

5.2.3.3.49. Dcm_SsBootloaderResetModeEntry

Purpose	Notification of Reset Mode entered.	
Synopsis	<code>Std_ReturnType Dcm_SsBootloaderResetModeEntry (void);</code>	
Service ID	<u>DCM_SERVID_COMMON_COMM_MODE_ENTRY</u>	
Sync/Async	Synchronous	



Reentrancy	Non-Reentrant
Return Value	Always return E_OK.
Description	This provides a confirmation for a 'SsBootloaderReset' request received from tester

5.2.3.3.50. Dcm_StartOfReception

Purpose	Called once to initialize the reception of a diagnostic request.	
Synopsis	BufReq_ReturnType Dcm_StartOfReception (PduIdType DcmRxPduId , PduLengthType TpSduLength , PduLengthType * RxBufferSizePtr);	
Service ID	DCM_SVCID_STARTOFRECEPTION	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	DcmRxPduId	Identifies the DCM data to be received. This information is used within the DCM to distinguish two or more receptions at the same time.
	TpSduLength	This length identifies the overall number of bytes to be received.
Parameters (out)	RxBufferSizePtr	pointer to PduLengthType containing length of the available buffer.
Return Value	Buffer request status	
	BUFREQ_OK	Buffer request accomplished successful.
	BUFREQ_E_NOT_OK	Buffer request not successful. Buffer cannot be accessed by TP.
	BUFREQ_E_OVFL	DCM is not capable to receive the number of TpSduLength Bytes.

5.2.3.3.51. Dcm_TpRxIndication

Purpose	Indicate that reception of data is finished.
Synopsis	void Dcm_TpRxIndication (PduIdType DcmRxPduId , NotifResultType Result);
Service ID	DCM_SVCID_TPRXINDICATION
Sync/Async	Synchronous
Reentrancy	Reentrant



Parameters (in)	DcmRxPduId	ID of DCM I-PDU that has been received.
	Result	Result of the N-PDU reception
Description	<p>This function is called by the lower layer (in general the PDU Router):</p> <ul style="list-style-type: none"> ▶ with Result = NTFSRSLT_OK after the complete DCM I-PDU has successfully been received, i.e. at the very end of the segmented TP receive cycle or after receiving an unsegmented N-PDU. ▶ with Result = NTFSRSLT_NOT_OK it is indicated that an error (e.g. timeout) has occurred during the reception of the DCM I-PDU. This passes the receive buffer back to DCM and allows error handling. It is undefined which part of the buffer contains valid data in this case, so the DCM shall not evaluate that buffer. By calling this service only the DCM is allowed to access the buffer. 	

5.2.3.3.52. Dcm_TpTxConfirmation

Purpose	Comfirm (un)successfull transmission.	
Synopsis	<pre>void Dcm_TpTxConfirmation (PduIdType DcmTxPduId , NotifResult- Type Result);</pre>	
Service ID	DCM_SVCID_TPTXCONFIRMATION	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	DcmTxPduId	ID of DCM I-PDU that has been transmitted.
	Result	Result of the N-PDU transmission.
Description	<p>This function is called by the lower layer (in general the PDU Router):</p> <ul style="list-style-type: none"> ▶ with Result = NTFSRSLT_OK after the complete DCM I-PDU has successfully been transmitted, i.e. at the very end of the segmented TP transmit cycle. Within this function, the DCM shall unlock the transmit buffer. ▶ with Result = NTFSRSLT_NOT_OK if an error (e.g. timeout) has occurred during the transmission of the DCM I-PDU. This enables unlocking of the transmit buffer and error handling. 	

5.2.3.3.53. Dcm_TriggerOnEvent

Purpose	Allows to notify Dcm that a event has been triggered (for service RoE).
----------------	---



Synopsis	<code>Std_ReturnType Dcm_TriggerOnEvent (uint8 RoeEventId);</code>	
Service ID	DCM_SERVID_TRIGGERONEVENT	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	RoeEventId	Identifier of the event that is triggered
Return Value	Validity of the event identifier RoeEventId E_OK - valid call of this function E_NOT_OK - if any other checks fail	
Description	When this function is called, Dcm executes the service requested in subfunction serviceToRespondTo.	

5.2.3.3.54. Dcm_TxConfirmation

Purpose	Comfirm (un)successfull transmission.	
Synopsis	<code>void Dcm_TxConfirmation (PduIdType DcmTxPduId);</code>	
Service ID	DCM_SVCID_TXCONFIRMATION	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different Pdulds. Non reentrant for the same Pduld.	
Parameters (in)	DcmTxPduId	ID of DCM I-PDU that has been transmitted.
Description	This function is called by the lower layer (in general the PDU Router). This enables unlocking of the transmit buffer and error handling.	

5.2.3.3.55. Dcm_WriteMemory

Purpose	Callout for download of data (used for service TransferData).	
Synopsis	<code>Dcm_ReturnWriteMemoryType Dcm_WriteMemory (Dcm_OpStatusType OpStatus , uint8 MemoryIdentifier , uint32 MemoryAddress , uint32 MemorySize , uint8 * MemoryData);</code>	
Service ID	DCM_SVCID_WRITEMEMORY	
Sync/Async	Asynchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	OpStatus	Operation status



	MemoryIdentifier	Identifier of the memory device (optional)
	MemoryAddress	Start address of the memory from which data is to be retrieved
	MemorySize	Number of bytes to be read
	MemoryData	Pointer to the buffer in Dcm
Return Value	DCM_WRITE_OK, DCM_WRITE_PENDING, DCM_WRITE_FAILED or DCM_WRITE_FORCE_RCRRP.	
Description	This callout allows the user to transfer data from the tester to the ECU, after successful execution of a RequestDownload service.	

5.2.4. Integration notes

5.2.4.1. Exclusive areas

This section describes the exclusive areas used by the `Dcm` module.

5.2.4.1.1. SCHM_DCM_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.

5.2.4.2. Production errors

Production errors are not reported by the `Dcm` module.

5.2.4.3. 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

5.2.4.4. 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.

5.2.4.4.1. dev.Dcm.External.ServiceAndSubService

Description	Diagnostic requests and subfunctions which are configured to be externally handled behave as follows: <ul style="list-style-type: none"> ➤ Service and subfunction request message validations are not performed as part of the Dcm. Request message validations for externally handled services and subfunctions are part of an integration activity. ➤ As the name of the callout is defined within the parameters <i>DcmDsdSidTabFnc</i> and <i>DcmDsdSubServiceFnc</i>, the service and subfunction handler prototype given by the SWS cannot be followed. ➤ A const qualifier is removed from the parameter <i>Dcm_MsgContextType* pMsgContext</i> because structure contents at the address <i>pMsgContext</i> get modified. This occurs, for example, if writing the positive response size in the <i>resDataLen</i> parameter of this structure while servicing the request. ➤ <i>DCM_PENDING</i> is used as a value for the <i>OpStatus</i> in the call. This is to inform the service or subfunction handler whether the previous execution is complete or not and needs to be resumed. Also see the enhancement related to 'Usage of <i>DCM_FORCE_RCRRP_OK</i> in external service handlers'.
Rationale	See the individual list items.

5.2.4.4.2. intgr.Dcm.ReadDTCInformation.ResponseBuffer

Description	The integrator should provide enough buffer for accommodating the total response if paged buffering is disabled. Also, buffer size should be at least enough to hold one DTC record if paged buffering is enabled.
Rationale	Response buffer size should suffice the total response if paging is disabled and it should have the minimum length if paging is enabled.



5.2.4.4.3. intgr.Dcm.DiagnosticSessionControl.ExtSubService

Description	The user cannot configure external subfunction handlers for service <i>DiagnosticSessionControl</i> (0x10).
Rationale	Subfunctions configured for service <i>DiagnosticSessionControl</i> (0x10) do not refer to different actions/functions to execute for different session levels requested. However, these subfunctions are configurable in order to allow security/session level and mode rule checks if transitioning between different session levels. Therefore they can all be handled in the same manner and no external subfunction handlers are required. If customized actions need to be executed for transitions to certain sessions, the <i>ServiceRequestNotification</i> interface may be used.

5.2.4.4.4. intgr.Dcm.ModeDeclaration.RangeLimit

Description	The range for <i>ModeDeclaration</i> within <i>ModeDeclarationGroup</i> is limited. The current implementation supports <i>ImplementationDataType</i> uint16 for <i>ModeDeclaration</i> inside <i>ModeDeclarationGroup</i> for modes provided by SWC or any other BSW mode manager (including Dcm) and required by the Dcm. This implies that a reference of <i>ModeDeclarationGroupPrototype</i> of <i>ModeDeclarationGroup</i> contains a maximum of 65535 elements of <i>ModeDeclaration</i> only.
Rationale	The Dcm does not know the total number of <i>ModeDeclaration</i> defined by an SWC for its <i>ModeDeclarationGroup</i> . The Dcm cannot infer this because the mapping of the data type to the <i>ModeDeclarationGroup</i> cannot be obtained from the reference to the <i>ModeDeclaration</i> which belongs to the <i>ModeDeclarationGroupPrototype</i> , see configuration parameter <i>DcmSwcModeRef</i> . Thus, the Dcm sets the data type for <i>ModeDeclaration</i> to uint16 in order to ensure compatibility with elements of <i>ModeDeclarationGroup</i> that contain a maximum of 65535 <i>ModeDeclaration</i> elements. Also see constr_1166 in Software Component Template, V4.2.0, AUTOSAR 4.0.3.

5.2.4.4.5. intgr.Dcm.Nvm.Timing

Description	The timing of operations issued by other users of the same NvM blocks that the Dcm uses must allow successful polling of block operation status by the Dcm. The integrator must ensure that other users of NvM blocks which the Dcm also uses shall execute operations on these NvM blocks only on startup for read operations or shutdown for write operations. The reason is to interfere as little as possible with the operations that the Dcm might execute using these blocks.
Rationale	The Dcm polls the state of NvM operations <i>NvM_ReadBlock()</i> and <i>NvM_WriteBlock()</i> by calling the <i>NvM_GetErrorStatus()</i> API. This is done cyclically in each <i>Dcm_Main</i> -



Function() cycle. If another user of the same NvM block triggers an operation between two of these polls, the user is enqueued. Thus, the next poll of the *NvM_GetErrorStatus()* API returns the result that an operation is still pending. Although the operation requested by the Dcm was completed, the Dcm is kept in a waiting state by the pending status of another user's operation. This means that the Dcm may time out while waiting for an operation on which its functionality does not depend to succeed. Requires clarification from Bugzilla http://www.autosar.org/bugzilla/show_bug.cgi?id=60936.

5.2.4.4.6. intgr.Dcm.DynamicallyDefineDataIdentifier.ExtSubService

Description	The user cannot configure external subfunction handlers for service <i>DynamicallyDefineDataIdentifier</i> (0x2C).
Rationale	Subfunctions configured for service <i>DynamicallyDefineDataIdentifier</i> (0x2C) do not refer to different actions/functions to execute for different session levels requested. However, these subfunctions are configurable in order to allow security/session level and mode rule checks if transitioning between different session levels. Therefore they can all be handled in the same manner and no external subfunction handlers are required. If customized actions need to be executed for transitions to certain sessions, the <i>ServiceRequestNotification</i> interface may be used.

5.2.4.4.7. intgr.Dcm.ReadDTCInformation.ReportDTCFaultDetectionCounter

Description	If the <i>ReadDTCInformation</i> (0x19) service is configured to also offer the EB-supplied <i>reportDTCFaultDetectionCounter</i> (0x14) subfunction, the following limitations apply: <ul style="list-style-type: none"> ▶ If the service <i>ReadDTC</i> (0x19) is configured to be processed synchronously (i.e.- the parameter <i>DcmDspProcessServiceAsync</i> is set to 'false'), the <i>TimingEvent_MainFunction</i> TimingEvent of the Dcm must be mapped either to: <ul style="list-style-type: none"> ▶ a task of a priority that is lower than the priority of any task which contains a direct call to the <i>Dcm_SetDTCFilter()</i> API or ▶ the same task that the <i>Dem_MainFunction()</i> is mapped to. ▶ If the service <i>ReadDTC</i> (0x19) is configured to be processed asynchronously (i.e. the parameter <i>DcmDspProcessServiceAsync</i> is set to 'true'), the <i>BswInternalTriggerOccurredEvent_Async</i> TriggerOccurredEvent of the Dcm must be mapped to a task of a priority which is lower than the priority of any task which contains a direct call to the <i>Dcm_SetDTCFilter()</i> API.
Rationale	This ensures that even though the Rte cannot correctly determine the context of calls to operations originating from the <i>Dcm_SetDTCFilter()</i> API call, corruption of data as a result of preemptions does not occur.



5.2.4.4.8. intgr.Dcm.NvmBlocks.ServiceNeedsWizard

Description	<p>When using NvM blocks for persistence in Dcm, the integrator must check the block sizes that are calculated by the Calculate Service Needs wizard against the size of the data that the block needs to contain. The following NvM blocks are affected:</p> <ul style="list-style-type: none"> ▶ BlockId: NVM_BLOCK_DCM_DDDID <ul style="list-style-type: none"> ▶ Variable: Dcm_DDDidSrcTable ▶ in case of UDS service 0x2C (DynamicallyDefineDataIdentifier) ▶ BlockId: NVM_BLOCK_DCM_ROE <ul style="list-style-type: none"> ▶ Variable: Dcm_Dsl_RoeServices_Persistent_Data ▶ in case of UDS service 0x86 (ResponseOnEvent)
Rationale	<p>The real size could be different from the calculated size for different platforms/compiler options.</p>

5.2.4.4.9. intgr.Dcm.SecurityAccess.DelayTimersMigration

Description	<p>When migrating to the ACG-7.7.2.3 or the ACG-8 Dcm, the mechanism for handling DelayTimers for the SecurityAccess (0x27) service changed. The following interfaces are no longer used:</p> <ul style="list-style-type: none"> ▶ <i>Rte_DcmSecSetNumAtt(uint8 SecurityLevel)</i> ▶ <i>Rte_DcmSecGetNumAtt(uint8 SecurityLevel, uint8 NumberOfAttempts)</i> <p>In order to migrate the behavior to the new interfaces, use the following generated or configured AUTOSAR 4.3-compliant:</p> <ul style="list-style-type: none"> ▶ <i>Xxx_GetSecurityAttemptCounter((optional Dcm_OpStatusType OpStatus), uint8* AttemptCounter)</i> ▶ <i>Xxx_SetSecurityAttemptCounter((optional Dcm_OpStatusType OpStatus), uint8 AttemptCounter)</i> <p>as wrappers for the previous interfaces. In order to do this, the <i>Xxx_Get/SetSecurityAttemptCounter</i> interface shall pass the SecurityLevel identifier corresponding to the SecurityLevel in question. These interfaces are configured/generated per SecurityLevel, therefore the SecurityLevel identifier is known within them.</p> <p>Alternatively, if the legacy behavior is required, perform the following actions:</p>
--------------------	---



	<ul style="list-style-type: none"> ▶ For every <i>DcmDspSecurityRow</i>, configure the <i>DcmDspSecurityAttemptCounter-Enabled</i> parameter to <i>FALSE</i>. ▶ Configure the <i>DcmDspSecurityLegacyAttemptCountersHandling</i> parameter to <i>TRUE</i>.
Rationale	The new implementation is based on AUTOSAR 4.3 and ISO 14229-1:2013. The old implementation was not completely and correctly specified within the AUTOSAR 4.0.-3 Dcm specification.

5.3. Dem

5.3.1. Configuration parameters

Containers included		
Container name	Multiplicity	Description
CommonPublishedInformation	1..1	<p>Label: Common Published Information</p> <p>Common container, aggregated by all modules. It contains published information about vendor and versions.</p>
DemConfigSet	1..1	<p>This container contains the configuration parameters and sub containers of the Dem module supporting multiple configuration sets.</p> <p>This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.</p>
DemGeneral	1..1	<p>Label: General Configuration</p> <p>This container contains the configuration (parameters) of the BSW Dem.</p>
DemDefensiveProgramming	1..1	<p>Label: Defensive Programming Options</p> <p>Parameters for defensive programming</p>
PublishedInformation	1..1	<p>Label: EB Published Information</p> <p>Additional published parameters not covered by CommonPublishedInformation container.</p>

Parameters included

Parameter name	Multiplicity



Parameters included

IMPLEMENTATION_CONFIG_VARIANT	1..1
---	------

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT	
Label	Configuration Variant	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	VariantPreCompile	
Range	VariantPreCompile	
Configuration class	VariantPreCompile:	VariantPreCompile

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
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	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	6
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	3
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	54
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. DemConfigSet

Containers included		
Container name	Multiplicity	Description
DemDTCClass	0..65535	This container contains the configuration (parameters) for DTCClass.
DemEventParameter	1..8190	<p>This container contains the configuration (parameters) for events.</p> <p>Note that this container definition does not explicitly define a symbolic name parameter. Instead, the short name of the container will be used in the Ecu configuration description to specify the symbolic name of the diagnostic event.</p>
DemPidClass	0..255	<p>This container contains the different PIDs for the global OBD-relevant freeze frame class. It is assembled out of one or several data elements.</p> <p>An OBD relevant event is an event with an OBD DTC either configured or calibrated. In case no OBD DTC is configured, then no data is captured for such event.</p> <p>The selection of DemFreezeFrameCapture defines the behavior of OBD freeze frame collection and reporting mechanism.</p> <p>For selection DEM_TRIGGER_ON_FDC_THRESHOLD, a single OBD freeze frame is available for storing for all OBD-relevant events. An OBD-relevant event is an event with an OBD DTC either configured or calibrated. The OBD freeze frame data behavior when DemFreezeFrameCapture equals DEM_TRIGGER_ON_FDC_THRESHOLD:</p>



Containers included

		<ul style="list-style-type: none"> ▶ store is done on first error recognition, meaning when UDS DTC status bit 2 (PendingDTC) transitions from 0 to 1. ▶ clear is done when UDS DTC status bit 2 (PendingDTC) transitions from 1 to 0 while UDS DTC status bit 3 (ConfirmedDTC) is not set. e.g. during aging, clearing of emission data ▶ update data, if another OBD-relevant event higher in priority reports its first error, the previous OBD event is replaced by the higher priority event and the data is overwritten with data of the higher priority event ▶ report, if data is available, with services: <ul style="list-style-type: none"> ▶ Service \$02 and the UDS Service \$19 05; Dem-Dcm API behavior via Dem_DcmGetDTCOfOBD-FreezeFrame() / Dem_DcmReadDataOfOBD-FreezeFrame() ▶ Service \$19 04; Dem-Dcm API behavior via Dem_GetSizeOfFreezeFrameSelection() / Dem_GetNextFreezeFrameData() <p>For a selection different than DEM_TRIGGER_ON_FDC_THRESHOLD, OBD freeze frame is available per entry of OBD relevant events. The OBD freeze frame data behavior when DemFreezeFrameCapture is different than DEM_TRIGGER_ON_FDC_THRESHOLD:</p> <ul style="list-style-type: none"> ▶ store is done on first error recognition, meaning when UDS DTC status bit 2 (PendingDTC) transitions from 0 to 1 on the event memory entry creation. ▶ clear is done along with UDS DTC status bit 2 (PendingDTC) and UDS DTC status bit 3 (ConfirmedDTC), both transitioning from 0 to 1 (e.g. during aging, clearing of emission data, on event memory entry displacement) ▶ update is not possible on OBD error reoccurrence, the event memory entry is not updated with new data ▶ report if data is available with the same services in the same format. The selection of the OBD freeze frame data has the following criteria applied:
--	--	---



Containers included

		<ul style="list-style-type: none"> ▶ priority-based selection and occurrence-based selection meaning between the events which caused the OBD freeze frame capturing the higher priority one which has the oldest entry creation time and is currently confirmed will be selected for reporting via services \$02 and \$19. <p>For service \$19 \$04 the event which stores the most relevant OBD freeze frame will react with RecordNumer 0x00.</p>
<u>DemAdvDisplacementPassiveTableLine</u>	0..n	<p>Describes a line in the passive table used in the advanced displacement algorithm.</p> <p>This container is available only if the following is configured:</p> <ul style="list-style-type: none"> ▶ DemEventDisplacementSupport = 'true' ▶ DemEventDisplacementStrategy == DEM_DISPLACEMENT_FULL ▶ DemExtendedDataCapture == DEM_TRIGGER_ON_FDC_THRESHOLD (because Si30 is available only in this mode)

5.3.1.3. DemDTCClass

Containers included

Container name	Multiplicity	Description
<u>DemCallbackInitMForF</u>	0..n	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>The presence of this container indicates that the Dem has access to an "InitMonitorForFunction" callback, which the Dem will call to initialize a monitor.</p> <p>If the container has a <code>DemCallbackInitMForFFnc</code>, this parameter defines the name of the function that the Dem will call.</p> <p>If there is no <code>DemCallbackInitMForFFnc</code>, the Dem will have an R-Port requiring the interface <code>CallbackInitMonitorForFunction</code>, whose name is generated by using the unique callback-prefix followed by the event name.</p>



Parameters included

Parameter name	Multiplicity
DemObdDTC	0..1
DemUdsDTC	0..1
DemJ1939DTCValue	0..1
DemJ1939NodeRef	1..1
DemDTCFunctionalUnit	1..1
DemDTCSeverity	0..1
DemImmediateNvStorage	1..1

Parameter Name	DemObdDTC			
Label	OBD DTC Value			
Description	<p>Diagnostic Trouble Code value for OBD.</p> <p>More than one DemDTCClass can be configured with identical OBD DTC value.</p> <p>Identical OBD DTC value usage between different DTC allows different events to typically use the same DTC value to report to Dcm.</p> <p>Important Note:</p> <p>Configuring identical OBD DTC value for multiple DTCs have impact on the following features</p> <ul style="list-style-type: none"> ▶ for services \$03/\$07/\$0A, same DTc will be reported multiple times. ▶ for PID01, same DTC can be counted multiple times due to several confirmed events. ▶ for PID01 and PID41, same DTC can belong to different readiness groups which may result in inconsistent readiness completion. ▶ hidden displacement of OBD freeze frame by identical OBD DTC events (if different priorities). 			
Multiplicity				
Type				
Configuration class	PreCompile:	VariantPreCompile		
Origin	AUTOSAR_ECUC			

Parameter Name	DemUdsDTC	
Label	UDS DTC Value	
Description	Unique Diagnostic Trouble Code value for UDS.	



	(Range: 0x000000 and 0xFFFF are reserved for DTC groups by ISO 14229-1)
	Events without DTC value will be cleared by ClearDTC for DEM_DTC_GROUP_ALL_DTCS
Multiplicity	0..1
Type	INTEGER
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemJ1939DTCValue
Label	J1939 DTC Value
Description	<p>Diagnostic Trouble Code value for J1939.</p> <p>Unique Diagnostic Trouble Code value for J1939 (consisting of SPN and FMI).</p> <p>The SPN is 19 bits and FMI is 5 bits.</p>
Multiplicity	0..1
Type	INTEGER
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC R4.2.2

Parameter Name	DemJ1939NodeRef
Label	J1939 NM node reference
Description	Reference to the J1939 NM node.
Multiplicity	1..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC R4.2.2

Parameter Name	DemDTCFunctionalUnit
Label	DTC Functional Unit
Description	DTCFunctionalUnit is a 1-byte value that identifies the corresponding basic vehicle / system function that reports the DTC. This parameter is necessary for the report of severity information.
Multiplicity	1..1
Type	INTEGER



Default value	0
Range	<=255 >=0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DemDTCSeverity
Label	DTC Severity
Description	<p>This parameter depends on the automotive manufacturer and is optional.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_DTC_SEV_NO_SEVERITY: No severity information available. ▶ DEM_DTC_SEV_MAINTENANCE_ONLY: Maintenance required. ▶ DEM_DTC_SEV_CHECK_AT_NEXT_HALT: Check at next halt. ▶ DEM_DTC_SEV_IMMEDIATELY: Check immediately.
Multiplicity	0..1
Type	ENUMERATION
Default value	DEM_DTC_SEV_NO_SEVERITY
Range	DEM_DTC_SEV_NO_SEVERITY DEM_DTC_SEV_MAINTENANCE_ONLY DEM_DTC_SEV_CHECK_AT_NEXT_HALT DEM_DTC_SEV_IMMEDIATELY
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DemImmediateNvStorage
Label	Immediate Nv Storage
Description	<p>Switch to enable immediate storage triggering of a respective event memory entry persistently to NVRAM.</p> <p>For an event with DemImmediateNvStorage ON, a respective event memory entry is stored immediately.</p> <p>For an event with DemImmediateNvStorage OFF, a respective event memory entry is stored persistently with <code>NvM_WriteAll()</code> during shut-down.</p>



	<ul style="list-style-type: none"> ▶ true: immediate non-volatile storage triggering enabled ▶ false: immediate non-volatile storage triggering disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

5.3.1.4. DemCallbackInitMForF

Parameters included	
Parameter name	Multiplicity
DemCallbackInitMForFFnc	0..1

Parameter Name	DemCallbackInitMForFFnc
Label	InitMonitorForFunction Callback C-Function
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Function name of prototype "InitMonitorForFunction".</p>
Multiplicity	0..1
Type	FUNCTION-NAME
Default value	Module_DemCallbackInitMForFunction
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

5.3.1.5. DemEventParameter

Containers included		
Container name	Multiplicity	Description
DemCallbackClearEventAllowed	0..1	The presence of this container indicates that the Dem has access to a "ClearEventAllowed" callback, which the Dem will call to realize the suppression handling of the DTC-deletion in Dem_ClearDTC.



Containers included

		<p>In case there is a <code>DemCallbackClearEventAllowedFnc</code>, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no <code>DemCallbackClearEventAllowedFnc</code>, the Dem will have an R-Port requiring the interface <code>CallbackClearEventAllowed</code> whose name is generated by using the unique callback-prefix followed by the event name.</p> <p>In the case that this is configured for OBD events that reference the MIL and <code>DemOBDCentralizedPID21Handling</code> is disabled, the information related to PID \$21 will be reset, and the Distance Traveled while MIL is activated will start counting from value 0.</p> <p>Note: If the event is used for event combination, this parameter is editable only if this event is configured as Combined DTC Master Configuration Event (<code>DemEvtCmbCommonParamMaster = TRUE</code>). Otherwise this parameter is not editable and derives its configuration value from the associated event with <code>DemEvtCmbCommonParamMaster</code> enabled.</p>
<u>DemCallbackEventDataChanged</u>	0..1	<p>The presence of this container indicates that the Dem has access to an "EventDataChanged" callback, which the Dem will call if an event is stored in its event memory and if an existing event memory entry is updated.</p> <p>In case there is a <code>DemCallbackEventDataChangedFnc</code>, this parameter defines the name of the function that the Dem will call if the event memory entry corresponding to this specific event is updated.</p> <p>In case there is no <code>DemCallbackEventDataChangedFnc</code>, the Dem will have an R-Port requiring the interface <code>CallbackEventDataChanged</code> whose name is generated by using the unique callback-prefix followed by the event name. This port will be triggered whenever the event memory entry corresponding to this specific event is updated.</p> <p>Dem will also have one global R-port <code>GeneralCBDataEvt</code> requiring the interface <code>GeneralCallbackEventDataChanged</code>, if it has at least one R-port requiring the interface <code>CallbackEventDataChanged</code>. This port will be trig-</p>



Containers included		
		<p>gered whenever the event memory entry corresponding to any event is updated.</p> <p>If one of these R-ports is not needed, it could be connected to a dummy P-port, without affecting the functionality.</p>
DemCallbackEventStatusChanged	0..n	<p>The presence of this container indicates, that the Dem has access to an "EventStatusChanged" callback, which the Dem will call to notify other components about the change in the status of an event. See also DemCallbackEventStatusChangedOpCycStartSupport.</p> <p>In case there is a DemCallbackEvenStatusChangedFnc, this parameter defines the name of the function that the Dem will call to notify about the change in the status of this specific event.</p> <p>In case there is no DemCallbackEvenStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackEventStatusChanged, whose name is generated by using the unique callback-prefix followed by the event name. This port will be triggered whenever the status of this specific event is changed.</p> <p>Dem will also have one global R-port GeneralCBStatusEvt requiring the interface GeneralCallbackEventsStatusChange, if it has at least one R-port requiring the interface CallbackEventStatusChanged. This port will be triggered whenever the status of any event is changed.</p> <p>If one of these R-ports is not needed, it could be connected to a dummy P-port, without affecting the functionality.</p>
DemCallbackInitMForE	0..1	<p>Label: InitMonitorForEvent Callback Function</p> <p>The presence of this container indicates, that the Dem has access to an "InitMonitorForEvent" callback, which the Dem will call to initialize a monitor.</p> <p>In case the container has a DemCallbackInitMForEFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackInitMForEFnc, the Dem will have an R-Port requiring the interface CallbackInit-</p>



Containers included

		<p><code>MonitorForEvent</code>, whose name is generated by using the unique callback-prefix followed by the event name.</p> <p>In case <code>DemCallbackInitMForEReenabledSupport</code> parameter is set to "true", the callback function <code>InitMonitorForEvent</code> will be triggered with <code>InitMonitorReason DEM_INIT_MONITOR_REENABLED</code> when one of the following conditions is satisfied:</p> <ul style="list-style-type: none"> ▶ Enable condition group status of the event is changed to fulfilled. ▶ ControlDTCSetting of the event is changed to enabled. ▶ Event availability is changed to available. <p>Note: If the ControlDTCSetting status is changed to enabled, then the callback function is triggered from the context of <code>Dem_MainFunction()</code>. Thus, run-time of the callback function will have a direct impact on the processing time inside <code>Dem_MainFunction()</code>.</p>
DemEventClass	1..1	<p>Label: Event Class</p> <p>This container contains the configuration (parameters) for <code>EventClass</code>.</p>

Parameters included

Parameter name	Multiplicity
DemEventId	1..1
DemEventKind	1..1
DemMaxNumberFreezeFrameRecords	1..1
DemEvtCmbCommonParamMaster	1..1
DemStatusBitStorageTestFailedPerEvent	0..1
DemDTCClassRef	0..1
DemExtendedDataClassRef	0..1
DemFreezeFrameClassRef	0..1
DemJ1939FreezeFrameClassRef	0..1
DemFreezeFrameRecNumClassRef	1..1
DemDevFreezeFrameAClassRef	0..1
DemDevFreezeFrameARecNum	1..1

**Parameters included**

DemDevFreezeFrameBClassRef	0..1
DemDevFreezeFrameBRecNum	1..1

Parameter Name	DemEventId
Label	Event Identifier
Description	<p>Unique identifier of a diagnostic event. This parameter should not be changeable by the user, because the ID should be generated by Dem itself to prevent gaps and multiple use of an ID.</p> <p>The event IDs shall be sequentially ordered beginning with 1 and no gaps in between.</p> <p>The Handle-ID wizard can be used to set this value automatically.</p>
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemEventKind
Label	Event Kind
Description	<p>This parameter is used to distinguish between SW-C and BSW events.</p> <p>SW-C events are reported by <code>Dem_SetEventStatus</code> API and BSW events are reported by <code>Dem_ReportErrorStatus</code> API.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ <code>DEM_EVENT_KIND_BSW</code>: The event is assigned to a BSW module. ▶ <code>DEM_EVENT_KIND_SWC</code>: The event is assigned to a SW-C.
Multiplicity	1..1
Type	ENUMERATION
Default value	<code>DEM_EVENT_KIND_SWC</code>
Range	<code>DEM_EVENT_KIND_BSW</code> <code>DEM_EVENT_KIND_SWC</code>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC



Parameter Name	DemMaxNumberFreezeFrameRecords
Label	Freeze Frame Records
Description	<p>This parameter defines the maximum number of the respective freeze frame records that can be stored for this event. Therefore, all these freeze frame records have the same freeze frame class.</p> <p>This parameter is available for calculated record numeration only (refer to DemTypeOfFreezeFrameRecordNumeration).</p> <p>The freeze frame record update mechanism depends on this parameter:</p> <ul style="list-style-type: none"> ▶ If DemMaxNumberFreezeFrameRecords is 1, then the first entry (which is the only entry) will not be updated when the event re-occurs. ▶ If DemMaxNumberFreezeFrameRecords (N) is greater than 1, then the last (Nth) entry is updated on the re-occurrence of the event, and the previous (1 to N-1) entries will remain unchanged when all the available freeze frame record slots for this event are occupied.
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<=255 >=1
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemEvtCmbCommonParamMaster
Label	Combined DTC Master Configuration Event
Description	<p>Define the "master configuration" event for combined DTCs, which is one selected event to configure the following common configuration parameters:</p> <ul style="list-style-type: none"> ▶ DemEventDestination ▶ DemEventPriority ▶ DemFreezeFrameRecNumClassRef ▶ DemFFPrestorageSupported ▶ DemFreezeFrameClassRef ▶ DemDevFreezeFrameAClassRef ▶ DemDevFreezeFrameBClassRef



	<ul style="list-style-type: none"> ▶ DemExtendedDataClassRef ▶ DemAgingAllowed ▶ DemAgingCycleCounterThreshold ▶ DemAgingCycleRef ▶ DemEventFailureCycleCounterThreshold ▶ DemEventFailureCycleRef ▶ DemCallbackClearEventAllowed ▶ DemCallbackClearEventAllowedFnc ▶ DemEventSignificance ▶ DemMinRatioEvent ▶ DemEventOBDReadinessGroup <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ This parameter is enabled only if the event is combined. ▶ true: This event is used as master for the combined DTC. ▶ false: This event is not used as master for the combined DTC.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemStatusBitStorageTestFailedPerEvent	
Label	Status Bit Storage Test Failed	
Description	<p>When enabled, this parameter will overwrite the value of general configuration parameter DemStatusBitStorageTestFailed for this event.</p> <ul style="list-style-type: none"> ▶ true: permanent storage activated ▶ false: permanent storage deactivated 	
Multiplicity	0..1	
Type	BOOLEAN	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	



Parameter Name	DemDTCClassRef	
Label	DTC Class	
Description	<p>This parameter defines the DTC configuration associated with the diagnostic event.</p> <p>It is allowed to have events without a DTC (e.g. for ECU-internal events triggering safety reactions without being reported via diagnostic communication). The same DemDTCClass can be used from several events, to combine these.</p> <p>If this parameter is disabled, the parameter DemStoreInternalEvents defines how internal events are stored.</p>	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DemExtendedDataClassRef	
Label	Extended Data Class	
Description	<p>This reference defines the link to an extended data class sampler.</p> <p>If the event is used for event combination, this parameter is editable only if this event is configured as Combined DTC Master Configuration Event (DemEvtCmbCommonParamMaster = TRUE).</p> <p>Otherwise this parameter is not editable and derives its configuration value from the associated event with DemEvtCmbCommonParamMaster enabled. In that case, a warning may be generated mentioning "value does not refer to nodes". This warning can be ignored accordingly.</p>	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DemFreezeFrameClassRef	
Label	Freeze Frame Class	
Description	<p>These references define the links to a freeze frame class sampler.</p> <p>The number of linked classes complies with the possible captured freeze frames, which are also reported from UDS service 0x19.</p>	



	If the event is used for event combination, this parameter is editable only if this event is configured as Combined DTC Master Configuration Event (DemEvtCmbCommonParamMaster = TRUE). Otherwise this parameter is not editable and derives its configuration value from the associated event with DemEvtCmbCommonParamMaster enabled. In that case, a warning may be generated mentioning "value does not refer to nodes". This warning can be ignored accordingly.
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemJ1939FreezeFrameClassRef
Label	J1939 Freeze Frame Class
Description	<p>These references define the links to a J1939 freeze frame class sampler.</p> <p>It shall have the following SPNs configured in the given order:</p> <ul style="list-style-type: none"> ▶ SPN 899 ▶ SPN 102 ▶ SPN 190 ▶ SPN 92 ▶ SPN 110 ▶ SPN 84 ▶ Manufacture-specific information
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC R4.2.2

Parameter Name	DemFreezeFrameRecNumClassRef
Label	Freeze Frame Record Number Class
Description	This parameter defines the list of dedicated freeze frame record numbers associated with the diagnostic event. These record numbers are assigned to the freeze frame records (instead of calculated record numbers).



	<p>This parameter is available for configured record numeration only (refer to DemTypeOfFreezeFrameRecordNumeration).</p> <p>If the event is used for event combination, this parameter is editable only if this event is configured as Combined DTC Master Configuration Event (DemEvtCmbCommonParamMaster = TRUE).</p> <p>Otherwise this parameter is not editable and derives its configuration value from the associated event with DemEvtCmbCommonParamMaster enabled. In that case, a warning may be generated mentioning "value does not refer to nodes". This warning can be ignored accordingly.</p>
Multiplicity	1..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemDevFreezeFrameAClassRef
Label	Development Freeze Frame A Class
Description	<p>Parameter used for pointing out the data for development freeze frame A. If this parameter is not enabled, there will be no development freeze frame A saved for this event.</p> <p>Development Freeze Frames are not supported if Event Combination is enabled.</p>
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemDevFreezeFrameARecNum
Label	Development Freeze Frame A Record Number
Description	The record number for development freeze frame A for this event.
Multiplicity	1..1
Type	INTEGER
Default value	1
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH
Parameter Name	DemDevFreezeFrameBClassRef



Label	Development Freeze Frame B Class	
Description	<p>Parameter used for pointing out the data for development freeze frame B. If this parameter is not enabled, there will be no development freeze frame B saved for this event.</p> <p>Development Freeze Frames are not supported if Event Combination is enabled.</p>	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemDevFreezeFrameBRecNum	
Label	Development Freeze Frame B Record Number	
Description	The record number for development freeze frame B for this event.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

5.3.1.6. DemCallbackClearEventAllowed

Parameters included		
Parameter name	Multiplicity	
DemCallbackClearEventAllowedFnc	0..1	
DemClearEventAllowedBehavior	1..1	

Parameter Name	DemCallbackClearEventAllowedFnc	
Label	ClearEventAllowed Callback C-Function	
Description	<p>For a description of the callback behavior, refer to the parent container.</p> <p>The expected C callback prototype is:</p> <pre>Std_ReturnType <Mod>_- DemClearEventAllowed<ForCondition> (boolean* Allowed)</pre> <p>To enable usage of this C callback function, proceed as follows:</p>	



	<ol style="list-style-type: none"> 1. Activate parent container DemCallbackClearEventAllowed and this parameter 2. Configure the related C function name, e.g. <code>Cdd_DemClearEventAllowed_0</code> 3. Configure the related C header file inclusion using the parameter DemHeaderFileInclusion
Multiplicity	0..1
Type	FUNCTION-NAME
Default value	Module_DemClearEventAllowed
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemClearEventAllowedBehavior	
Label	ClearEventAllowed Behavior	
Description	<p>Defines the resulting UDS status byte for the related event, which must not be cleared according to the ClearEventAllowed callback.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ <code>DEM_NO_STATUS_BYTE_CHANGE</code>: The event UDS status byte remains unchanged. ▶ <code>DEM_ONLY_THIS_CYCLE_AND_READINESS</code>: The failure bits (<code>TestFailedThisOperationCycle & TestFailedSinceLastClear</code>) are set to 0 and the readiness bits (<code>TestNotCompletedSinceLastClear & TestNotCompletedThisOperationCycle</code>) are set to 1 in the event UDS status byte. All other bits in the event UDS status byte remain unchanged. 	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	<code>DEM_NO_STATUS_BYTE_CHANGE</code>	
Range	<code>DEM_NO_STATUS_BYTE_CHANGE</code> <code>DEM_ONLY_THIS_CYCLE_AND_READINESS</code>	
Configuration class	VariantPreCompile: VariantPreCompile	
Origin	AUTOSAR_ECUC	



5.3.1.7. DemCallbackEventDataChanged

Parameters included	
Parameter name	Multiplicity
DemCallbackEventDataChangedFnc	0..1

Parameter Name	DemCallbackEventDataChangedFnc	
Label	EventDataChanged Callback C-Function	
Description	<p>For a description of the callback behavior, refer to the parent container.</p> <p>The expected C callback prototype is:</p> <pre>void <Mod>_DemTriggerOnEventData(EventIdType EventId)</pre> <p>To enable usage of this C callback function, proceed as follows:</p> <ol style="list-style-type: none"> 1. Activate parent container DemCallbackEventDataChanged and this parameter 2. Configure the related C function name, e.g. Cdd_DemTriggerOnEvent-Data 3. Configure the related C header file inclusion using the parameter Dem-HeaderFileInclusion 	
Multiplicity	0..1	
Type	FUNCTION-NAME	
Default value	Module_DemTriggerOnEventData	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.3.1.8. DemCallbackEventStatusChanged

Parameters included	
Parameter name	Multiplicity
DemCallbackEventStatusChangedFnc	0..1

Parameter Name	DemCallbackEventStatusChangedFnc
Label	EventStatusChanged Callback C-Function
Description	For a description of the callback behavior, refer to the parent container.



	<p>The expected C callback prototype is:</p> <pre>void &lt;Mod&gt;_DemTriggerOnEventStatus(Dem_EventIdType EventId, Dem_EventStatusExtendedType EventStatusOld, Dem_EventStatusExtendedType EventStatusNew)</pre> <p>To enable usage of this C callback function, proceed as follows:</p> <ol style="list-style-type: none"> 1. Add callback function in the callback list DemCallbackEventStatusChanged 2. Activate this parameter 3. Configure the related C function name, e.g. Cdd_DemTriggerOnEventStatus 4. Configure the related C header file inclusion using the parameter Dem-HeaderFileInclusion
Multiplicity	0..1
Type	FUNCTION-NAME
Default value	Module_DemTriggerOnEventStatus
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

5.3.1.9. DemCallbackInitMForE

Parameters included	
Parameter name	Multiplicity
DemCallbackInitMForEFnc	0..1

Parameter Name	DemCallbackInitMForEFnc
Label	InitMonitorForEvent Callback C-Function
Description	<p>For a description of the callback behavior, refer to the parent container.</p> <p>The expected C callback prototype is:</p> <pre>Std_ReturnType &lt;Mod&gt;_- DemInitMonitorFor&lt;EventName&gt; (Dem_InitMonitorReason- Type InitMonitorReason)</pre> <p>To enable usage of this C callback function, proceed as follows:</p> <ol style="list-style-type: none"> 1. Activate parent container DemCallbackInitMForE and this parameter 2. Configure the related C function name, e.g. Cdd_DemInitMonitor1



	3. Configure the related C header file inclusion using the parameter Dem-HeaderFileInclusion
Multiplicity	0..1
Type	FUNCTION-NAME
Default value	Module_DemCallbackInitMForEvent
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.3.1.10. DemEventClass

Containers included		
Container name	Multiplicity	Description
DemDebounceAlgorithm-Class	1..1	Label: Debounce Algorithm Class Debounce algorithm class: counter-based, time-based, or monitor internal. Additionally the implementation-specific Debounce algorithm class 'frequency based' is provided.
DemIndicatorAttribute	0..255	This container contains the event specific configuration of Indicators.

Parameters included	
Parameter name	Multiplicity
DemAgingAllowed	1..1
DemAgingCycleCounterThreshold	1..1
DemOBDAgingCycleCounterThreshold	1..1
DemConsiderPtoStatus	1..1
DemMinRatioEvent	1..1
DemEventDestination	1..1
DemEventFailureCycleCounterThreshold	0..1
DemEventOBDReadinessGroup	1..1
DemEventPriority	1..1
DemEventSignificance	0..1
DemFFPrestorageSupported	1..1
DemAgingCycleRef	0..1
DemEnableConditionGroupRef	0..1

**Parameters included**

DemEventAvailable	1..1
DemEventFailureCycleRef	0..1
DemOperationCycleRef	1..1
DemStorageConditionGroupRef	0..1

Parameter Name	DemAgingAllowed	
Label	Aging Allowed	
Description	<p>Switch to allow aging/unlearning of the event or not.</p> <ul style="list-style-type: none"> ▶ true: aging allowed ▶ false: aging not allowed <p>If the event is used for event combination, this parameter is enabled only if this event is configured as Combined DTC Master Configuration Event (DemEvtCmbCommonParamMaster = TRUE). Otherwise this parameter is not enabled and derives its configuration value from the associated event with DemEvtCmbCommonParamMaster enabled.</p> <p>See also Dem_SetOperationCycleState requires tested cycles and Dem_SetAgingCycleState requires tested cycles option for selecting the desired aging behavior.</p> <p>See also aging based on healing completion option for selecting the desired aging behavior.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemAgingCycleCounterThreshold	
Label	Aging Cycle Counter Threshold	
Description	<p>Number of aging cycles needed to unlearn/delete the event.</p> <p>If the event is used for event combination, this parameter is editable only if this event is configured as Combined DTC Master Configuration Event (DemEvtCmbCommonParamMaster = TRUE). Otherwise this parameter is not editable and derives its configuration value from the associated event with DemEvtCmbCommonParamMaster enabled.</p>	



	See also Dem_SetOperationCycleState requires tested cycles and Dem_SetAgingCycleState requires tested cycles option for selecting the desired aging behavior.
Multiplicity	1..1
Type	INTEGER
Default value	40
Range	<=255 >=1
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemOBDAgingCycleCounterThreshold
Label	OBD Aging Cycle Counter Threshold
Description	Number of aging cycles needed to unlearn/delete an OBD event. This threshold is checked against OCC5 counter.
Multiplicity	1..1
Type	INTEGER
Default value	40
Range	<=255 >=1
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemConsiderPtoStatus
Label	Consider PTO Status
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> This parameter is TRUE, when the event is affected by the Dem PTO handling.
Multiplicity	1..1
Type	BOOLEAN
Default value	false



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DemMinRatioEvent
Label	Minimum Ratio Event
Description	<p>Selects if the event using this class is subject of minimum ratio requirements as specified in Amend section 1968.2, title 13, California Code of Regulations. This information is used to select which conditions to check when performing clear of permanent fault codes.</p> <p>If the event is used for event combination, this parameter is enabled only if this event is configured as Combined DTC Master Configuration Event (DemEvtCmbCommonParamMaster = TRUE). Otherwise, this parameter is not enabled and derives its configuration value from the associated event with DemEvtCmbCommonParamMaster enabled.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PreCompile:
	VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemEventDestination
Label	Event Destination
Description	<p>The event destination assigns events to none, one or multiple origins.</p> <p>If no event destination is assigned to a specific event, the event is handled internally and is not visible externally to the Dcm (<i>currently not supported</i>).</p> <p>If more than one event destination is assigned to a specific event, the event can be present in the corresponding origins (<i>currently not supported</i>).</p> <p>If the event is used for event combination, this parameter is enabled only if this event is configured as Combined DTC Master Configuration Event (DemEvtCmbCommonParamMaster = TRUE). Otherwise, this parameter is not enabled and derives its configuration value from the associated event with DemEvtCmbCommonParamMaster enabled.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_DTC_ORIGIN_PRIMARY_MEMORY: Event information located in the primary memory.



	<ul style="list-style-type: none"> ▶ DEM_DTC_ORIGIN_SECONDARY_MEMORY: Event information located in the secondary memory. ▶ DEM_DTC_ORIGIN_MIRROR_MEMORY: Event information located in the mirror memory.
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_DTC_ORIGIN_PRIMARY_MEMORY
Range	DEM_DTC_ORIGIN_PRIMARY_MEMORY DEM_DTC_ORIGIN_SECONDARY_MEMORY DEM_DTC_ORIGIN_MIRROR_MEMORY
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemEventFailureCycleCounterThreshold
Label	Event Failure Cycle Counter Threshold
Description	<p>Defines the number of failure cycles for the event based fault confirmation.</p> <p>In case the failure cycle equals the operations cycle of the event, this means: If the failure cycle counter threshold is configured to value N and in the Nth consecutive operation cycle with at least one FAILED qualification, the ConfirmedDTC status bit shall be set with the FAILED qualification.</p> <p>In case the failure cycle is different to the operations cycle of the event, this means: If the failure cycle counter threshold is configured to value N and the failure cycle is started N times with the event being qualified as FAILED in its current operation cycle (i.e. TFTOC is set), the ConfirmedDTC status bit shall be set.</p> <p>Operation Cycles without any qualification will not influence this behavior.</p> <p>An Operation Cycle with only PASSED qualification(s) will reset the Event Failure Cycle Counter.</p> <p>If the parameter is disabled, then fault confirmation is switched off (which equals the setting DemEventFailureCycleCounterThreshold is 1 and DemEventFailureCycleRef is equal to DemOperationCycleRef).</p> <p>If the event is used for event combination, this parameter is editable only if this event is configured as Combined DTC Master Configuration Event (DemEvtCmbCommonParamMaster = TRUE). Otherwise, this parameter is not editable and</p>



	derives its configuration value from the associated event with DemEvtCmbCommonParamMaster enabled.
Multiplicity	0..1
Type	INTEGER
Default value	1
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemEventOBDReadinessGroup
Label	Event OBD Readiness Group
Description	<p>This parameter specifies the Event OBD Readiness group for PID \$01 and PID \$41 computation.</p> <p>This parameter is only applicable for emission-related ECUs.</p> <p>If the event is used for event combination, this parameter is editable only if this event is configured as Combined DTC Master Configuration Event (DemEvtCmbCommonParamMaster = TRUE). Otherwise, this parameter is not editable and derives its configuration value from the associated event with DemEvtCmbCommonParamMaster enabled.</p> <p>If calibration support is enabled, the number of events per readiness group is limited by parameter DemMaxNumEventsPerReadinessGroup.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_OBD_RDY_AC: A/C system component - spark. ▶ DEM_OBD_RDY_BOOSTPR: Boost Pressure System - compr. ▶ DEM_OBD_RDY_CAT: Catalyst - spark. ▶ DEM_OBD_RDY_CMPPRCMPT: Comprehensive component spark, compr. ▶ DEM_OBD_RDY_EGSENS: Exhaust Gas Sensor - compr. ▶ DEM_OBD_RDY_EGR: EGR system - spark, compr. ▶ DEM_OBD_RDY_EVAP: Evaporative system - spark. ▶ DEM_OBD_RDY_FLSYS: Fuel system - spark, compr. ▶ DEM_OBD_RDY_FLSYS_NONCONT: Non Continuous Fuel system - spark, compr ▶ DEM_OBD_RDY_HCCAT: Non-Methan HC Catalyst - compr. ▶ DEM_OBD_RDYHTCAT: Heated catalyst - spark.



	<ul style="list-style-type: none"> ▶ DEM_OBD_RDY_MISF: Misfire - spark, compr. ▶ DEM_OBD_RDY_NONE: None - spark, compr. ▶ DEM_OBD_RDY_NOXCAT: NOx Catalyst - compr. ▶ DEM_OBD_RDY_O2SENS: Oxygen sensor - spark. ▶ DEM_OBD_RDY_O2SENSHT: Oxygen sensor heater - spark. ▶ DEM_OBD_RDY_PMFLT: Particle Matters Filter - compr. ▶ DEM_OBD_RDY_SECAIR: Secondary air system - spark.
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_OBD_RDY_NONE
Range	DEM_OBD_RDY_AC DEM_OBD_RDY_BOOSTPR DEM_OBD_RDY_CAT DEM_OBD_RDY_CMPPRCMPT DEM_OBD_RDY_EGSENS DEM_OBD_RDY_EGR DEM_OBD_RDY_EVAP DEM_OBD_RDY_FLSYS DEM_OBD_RDY_FLSYS_NONCONT DEM_OBD_RDY_HCCAT DEM_OBD_RDY_HTCAT DEM_OBD_RDY_MISF DEM_OBD_RDY_NONE DEM_OBD_RDY_NOXCAT DEM_OBD_RDY_O2SENS DEM_OBD_RDY_O2SENSHT DEM_OBD_RDY_PMFLT DEM_OBD_RDY_SECAIR
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	DemEventPriority
Label	Event Priority



Description	<p>Priority of the event, in view of full event buffer.</p> <p>The event priority is defined as a ranking of DTCs based upon level of importance. The value 1 represents the highest priority and each value sequentially larger results into a lower importance.</p> <p>This parameter is available only when event displacement is enabled (refer to DemEventDisplacementSupport).</p> <p>If the event is used for event combination, this parameter is enabled only if this event is configured as Combined DTC Master Configuration Event (DemEvtCmbCommonParamMaster = TRUE). Otherwise this parameter is not enabled and derives its configuration value from the associated event with DemEvtCmbCommonParamMaster enabled.</p>
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<p><=255</p> <hr/> <p>>=1</p>
Configuration class	<p>PreCompile:</p> <div style="display: flex; justify-content: space-between;"> VariantPreCompile </div>
Origin	AUTOSAR_ECUC

Parameter Name	DemEventSignificance
Label	Event Significance
Description	<p>Significance of the event, which indicates additional information concerning fault classification and resolution.</p> <p>It can be mapped as Dem-internal data element. It shall be configured, if it is a part of event related data.</p> <p>If the event is used for event combination, this parameter is editable only if this event is configured as Combined DTC Master Configuration Event (DemEvtCmbCommonParamMaster = TRUE). Otherwise, this parameter is not editable and derives its configuration value from the associated event with DemEvtCmbCommonParamMaster enabled.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_EVENT_SIGNIFICANCE_FAULT: failure, which affects the component/ECU itself.



	► DEM_EVENT_SIGNIFICANCE_OCCURRENCE: issue, which indicates additional information concerning insufficient system behavior.
Multiplicity	0..1
Type	ENUMERATION
Default value	DEM_EVENT_SIGNIFICANCE_FAULT
Range	DEM_EVENT_SIGNIFICANCE_FAULT DEM_EVENT_SIGNIFICANCE_OCCURRENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	DemFFPrestorageSupported
Label	Freeze Frame Prestorage Support
Description	If this parameter is set to true, then the Prestorage of FreezeFrames is supported by the assigned event. Currently combined events do not support Freeze Frame Prestorage. This parameter is useful to calculate the buffer size.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	DemAgingCycleRef
Label	Aging Cycle
Description	Reference to the cycle which is triggering the aging of the event. This can either be the same as the operation cycle of the event, or a separate aging cycle reported via API Dem_SetAgingCycleState. If external aging is configured (refer to DemAgingCycleCounterProcessing), this parameter is not used. If the common operation cycle support is enabled (DemCommonOperationCycleSupport = true), this parameter (DemAgingCycleRef) is disabled. If the event is used for event combination, this parameter is editable only if this event is configured as Combined DTC Master Configuration Event (DemEvtCmbCommonParamMaster = TRUE). Otherwise this parameter is not editable and derives its configuration value from the associated event with DemEvtCmbCommonParamMaster enabled.



Multiplicity	0..1	
Type	CHOICE-REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU V1.0.0	

Parameter Name	DemEnableConditionGroupRef	
Label	Enable Condition Group	
Description	References an enable condition group.	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU V1.0.0	

Parameter Name	DemEventAvailable	
Label	Event Available	
Description	<p>This parameter configures, whether the related event is available or not regarding to event processing.</p> <ul style="list-style-type: none"> ▶ TRUE: the related event will be completely processed as usual ▶ FALSE: the related event is treated as if it does not exist <p>Note: If DemCalibrationSupport is enabled, DemClearDTCOfDisabledEvents can be enabled to allow ClearDTC() request for <code>DEM_DTC_GROUP_ALL_DTCs</code> to delete even disabled events e.g. when DemEventAvailable was calibrated to 'false' when an event entry already exists.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DemEventFailureCycleRef	
Label	Event Failure Cycle	
Description	<p>Failure cycle for the event based fault confirmation.</p> <p>START report of this failure cycle triggers an update of the failure counter, if the event has been qualified as failed at least once during its event operation cycle.</p>	



	<p>This reference promotes an operation cycle to a failure cycle.</p> <p>If an Event Failure Cycle is configured, also an Event Failure Cycle Counter Threshold (DemEventFailureCycleCounterThreshold) must be configured.</p> <p>If the common operation cycle support is enabled (DemCommonOperationCycleSupport = true), this parameter (DemEventFailureCycleRef) is disabled.</p> <p>If the event is used for event combination, this parameter is editable only if this event is configured as Combined DTC Master Configuration Event (DemEvtCmbCommonParamMaster = TRUE). Otherwise this parameter is not editable and derives its configuration value from the associated event with DemEvtCmbCommonParamMaster enabled.</p> <p>If the operation cycle (DemOperationCycleRef) is of type DEM_OPCYC_OBD_DCY, the event failure cycle must be the same.</p>
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC V1.0.0

Parameter Name	DemOperationCycleRef
Label	Operation Cycle Reference
Description	Kind of operation cycle for the event (e.g. power cycle, driving cycle, ...).
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC V1.0.0

Parameter Name	DemStorageConditionGroupRef
Label	Storage Condition Group
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> References a storage condition group.
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC V1.0.0



5.3.1.11. DemDebounceAlgorithmClass

Containers included		
Container name	Multiplicity	Description
DemDebounceMonitorInternal	1..1	This container contains the configuration (parameters) for monitor internal debouncing.
DemDebounceCounterBased	1..1	This container contains the configuration (parameters) for counter-based debouncing.
DemDebounceTimeBase	1..1	This container contains the configuration (parameters) for time based debouncing.
DemDebounceFrequency-Based	1..1	This container contains the configuration (parameters) for DemDebounceFrequencyBased.

5.3.1.12. DemDebounceMonitorInternal

Containers included		
Container name	Multiplicity	Description
DemCallbackGetFDC	0..1	<p>Label: Fault Detection Counter Callback Function</p> <p>The presence of this container indicates, that the Dem has access to a "GetFaultDetectionCounter" callback, which the Dem will call to obtain the value of the fault detection counter.</p> <p>In case the container has a <code>DemCallbackGetFDCFn</code>, this parameter defines the name of the function that the Dem will call. If this callback function returns <code>E_NOT_OK</code>, meaning the fetched FDC may be invalid, the FDC will be considered to have the value equal to 0.</p> <p>In case there is no <code>DemCallbackGetFDCFn</code>, the Dem will have a R-Port requiring the interface <code>CallbackGetFault-DetectionCounter</code>, whose name is generated by using the unique callback-prefix followed by the event name.</p>

5.3.1.13. DemCallbackGetFDC

Parameters included	
Parameter name	Multiplicity

**Parameters included**

DemCallbackGetFDCFnc	0..1
--------------------------------------	------

Parameter Name	DemCallbackGetFDCFnc
Label	GetFaultDetectionCounter Callback C-Function
Description	<p>For a description of the callback behavior, refer to the parent container.</p> <p>The expected C callback prototype is:</p> <pre>Std_ReturnType <Mod>_- DemGetFaultDetectionCounter<ForEvent> (sint8* FaultDe- tectionCounter)</pre> <p>To enable usage of this C callback function, proceed as follows:</p> <ol style="list-style-type: none"> 1. Activate parent container DemCallbackGetFDC and this parameter 2. Configure the related C function name, e.g. <code>Cdd_DemGetFaultDetec- tionCounterForEvent</code> 3. Configure the related C header file inclusion using the parameter Dem- HeaderFileInclusion
Multiplicity	0..1
Type	FUNCTION-NAME
Default value	Module_DemGetFaultDetectionCounterForEvent
Configuration class	PreCompile: <input type="text"/> VariantPreCompile
Origin	AUTOSAR_ECUC

5.3.1.14. DemDebounceCounterBased**Parameters included**

Parameter name	Multiplicity
DemDebounceBehavior	1..1
DemDebounceCounterDecrementStepSize	1..1
DemDebounceCounterFailedThreshold	1..1
DemDebounceCounterUnconfirmedThreshold	1..1
DemDebounceCounterIncrementStepSize	1..1
DemDebounceCounterJumpDown	1..1
DemDebounceCounterJumpDownValue	1..1

**Parameters included**

DemDebounceCounterJumpUp	1..1
DemDebounceCounterJumpUpValue	1..1
DemDebounceCounterPassedThreshold	1..1
DemDebounceCounterStorage	1..1

Parameter Name	DemDebounceBehavior
Label	Event Debounce Algorithm Behavior
Description	<p>This parameter defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled.</p> <ul style="list-style-type: none"> ▶ DEM_DEBOUNCE_FREEZE: The event debounce counter will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and ControlDTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. SetEventStatus). ▶ DEM_DEBOUNCE_RESET: The event debounce counter will be reset to initial value if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. The qualification of the event will be restarted with the next valid event report.
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_DEBOUNCE_FREEZE
Range	DEM_DEBOUNCE_FREEZE DEM_DEBOUNCE_RESET
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemDebounceCounterDecrementStepSize
Label	Counter Decrement
Description	Defines the step size for decrement of the internal fault detection counter (PREPASSED).
Multiplicity	1..1
Type	INTEGER
Default value	1



Range	<=32768
	>=1
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemDebounceCounterFailedThreshold	
Label	Counter Failed Threshold	
Description	Defines the value of the internal fault detection counter, which indicates the failed status.	
Multiplicity	1..1	
Type	INTEGER	
Default value	127	
Range	<=32767	
	>=1	
Configuration class	VariantPreCompile: VariantPreCompile	
Origin	AUTOSAR_ECUC	

Parameter Name	DemDebounceCounterUnconfirmedThreshold	
Label	Counter Unconfirmed Threshold	
Description	<p>Defines the value of the internal fault detection counter, which indicates the "unconfirmedDTC" status.</p> <p>When the fault detection counter reaches the unconfirmedDTC threshold, freeze frame and/or extended data is sampled (asynchronously).</p> <p>In case unconfirmedDTC threshold is set to be same as failed threshold, the unconfirmedDTC threshold is deactivated for this event.</p> <p>To enable editing of this parameter, DemFreezeFrameCapture and/or DemExtendedDataCapture need to be set to DEM_TRIGGER_ON_FDC_THRESHOLD.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	127	
Configuration class	PreCompile: VariantPreCompile	
Origin	Elektrobit Automotive GmbH	



Parameter Name	DemDebounceCounterIncrementStepSize	
Label	Counter Increment	
Description	Defines the step size for increment of the internal fault detection counter (PRE-FAILED).	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=32767	>=1
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemDebounceCounterJumpDown	
Label	Jump-Down	
Description	Switch for the activation of Jump-Down. ▶ true: Jump-Down activated ▶ false: Jump-Down deactivated	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemDebounceCounterJumpDownValue	
Label	Jump-Down Value	
Description	Jump-Down value of the internal fault detection counter.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemDebounceCounterJumpUp	
-----------------------	---------------------------------	--



Label	Jump-Up	
Description	Switch for the activation of Jump-Up. ► true: Jump-Up activated ► false: Jump-Up deactivated	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemDebounceCounterJumpUpValue	
Label	Jump-Up Value	
Description	Jump-Up value of the internal fault detection counter.	
Multiplicity	1..1	
Type	INTEGER	
Default value	0	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemDebounceCounterPassedThreshold	
Label	Counter Passed Threshold	
Description	Defines the value of the internal fault detection counter, which indicates the passed status.	
Multiplicity	1..1	
Type	INTEGER	
Default value	-128	
Range	<=-1 >=-32768	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemDebounceCounterStorage	
Label	Debounce Counter Nv Storage	
Description	Switch to store the debounce counter value non-volatile or not.	



	<ul style="list-style-type: none"> ▶ true: debounce counter value shall be stored non-volatile ▶ false: debounce counter value is volatile <p>DemOperationCycleStatusStorage must be enabled in order to use this feature.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.3.1.15. DemDebounceTimeBase

Parameters included	
Parameter name	Multiplicity
DemDebounceBehavior	1..1
DemDebounceTimeFailedThreshold	1..1
DemDebounceTimePassedThreshold	1..1

Parameter Name	DemDebounceBehavior
Label	Event Debounce Algorithm Behavior
Description	<p>This parameter defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled.</p> <ul style="list-style-type: none"> ▶ DEM_DEBOUNCE_FREEZE: The event debounce timer will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and ControlDTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. SetEventStatus). ▶ DEM_DEBOUNCE_RESET: The event debounce timer will be reset to initial value if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. The qualification of the event will be restarted with the next valid event report. ▶ DEM_DEBOUNCE_CONTINUE: The event time-based debouncing algorithm will continue on the next Dem_MainFunction() call if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled.



Multiplicity	1..1	
Type	ENUMERATION	
Default value	DEM_DEBOUNCE_CONTINUE	
Range	DEM_DEBOUNCE_CONTINUE DEM_DEBOUNCE_FREEZE DEM_DEBOUNCE_RESET	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemDebounceTimeFailedThreshold	
Label	Event Failed Timeout (s)	
Description	Defines the time out duration for "Event Failed" qualification. The AUTOSAR configuration standard is to use SI units, so this parameter is defined as float value in seconds.	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.01	
Range	<=3600 >=0.001	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemDebounceTimePassedThreshold	
Label	Event Passed Timeout (s)	
Description	Defines the time out duration for "Event Passed" qualification. The AUTOSAR configuration standard is to use SI units, so this parameter is defined as float value in seconds.	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.01	
Range	<=3600 >=0.001	



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

5.3.1.16. DemDebounceFrequencyBased

Parameters included	
Parameter name	Multiplicity
DemDebounceBehavior	1..1
DemDurationOfTimeWindow	1..1
DemThresholdForEventTestedFailed	1..1
DemThresholdForEventTestedPassed	1..1
Parameter Name	DemDebounceBehavior
Label	Event Debounce Algorithm Behavior
Description	<p>This parameter defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled.</p> <ul style="list-style-type: none"> ▶ DEM_DEBOUNCE_FREEZE: The event debounce timer frequency will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and ControlDTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. SetEventStatus). ▶ DEM_DEBOUNCE_RESET: The event debounce timer frequency will be reset to initial value if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. The qualification of the event will be restarted with the next valid event report. ▶ DEM_DEBOUNCE_CONTINUE: The event frequency-based debouncing algorithm will continue on the next Dem_MainFunction() call if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled.
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_DEBOUNCE_CONTINUE
Range	DEM_DEBOUNCE_CONTINUE DEM_DEBOUNCE_FREEZE



	DEM_DEBOUNCE_RESET
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemDurationOfTimeWindow	
Label	Time Window (s)	
Description	<p>Defines duration of the Time Window.</p> <p>The AUTOSAR configuration standard is to use SI units, so this parameter is defined as float value in seconds.</p>	
Multiplicity	1..1	
Type	FLOAT	
Default value	0.01	
Range	<p><=3600</p> <p>>=0</p>	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemThresholdForEventTestedFailed	
Label	Event Failed Threshold	
Description	Defines the threshold for FAILED-detection.	
Multiplicity	1..1	
Type	INTEGER	
Default value	10	
Range	<p><=65535</p> <p>>=1</p>	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemThresholdForEventTestedPassed	
Label	Event Passed Threshold	
Description	Defines the threshold for PASSED-detection.	
Multiplicity	1..1	
Type	INTEGER	



Default value	10
Range	<=65535 >=1
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.3.1.17. DemIndicatorAttribute

Parameters included	
Parameter name	Multiplicity
DemIndicatorBehaviour	1..1
DemIndicatorFailureCycleCounterThreshold	0..1
DemIndicatorFailureCycleSource	1..1
DemIndicatorHealingCycleCounterThreshold	1..1
DemIndicatorFailureCycleRef	0..1
DemIndicatorHealingCycleRef	1..1
DemIndicatorRef	1..1

Parameter Name	DemIndicatorBehaviour
Label	Indicator Behaviour
Description	<p>Behaviour of the linked indicator.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_INDICATOR_BLINKING: The indicator blinks when the event has status FAILED. ▶ DEM_INDICATOR_BLINK_CONT: The indicator is active and blinks when the event has status FAILED. ▶ DEM_INDICATOR_CONTINUOUS: The indicator is active when the event has status FAILED. ▶ DEM_INDICATOR_SLOW_FLASH: Flash Indicator Lamp should be set to Fast Flash. ▶ DEM_INDICATOR_FAST_FLASH: Flash Indicator Lamp should be set to Slow Flash.
Multiplicity	1..1



Type	ENUMERATION	
Default value	DEM_INDICATOR_CONTINUOUS	
Range	DEM_INDICATOR_CONTINUOUS DEM_INDICATOR_BLINKING DEM_INDICATOR_BLINK_CONT DEM_INDICATOR_SLOW_FLASH DEM_INDICATOR_FAST_FLASH	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemIndicatorFailureCycleCounterThreshold	
Label	Indicator Failure Cycle Counter Threshold	
Description	<p>Defines the number of failure cycles for the WarningIndicatorOnCriteria.</p> <p>In case the indicator failure cycle equals the operations cycle of the event, this means: If the indicator failure cycle counter threshold is configured to value N and in the Nth consecutive operation cycle with at least one FAILED qualification, the WIR status bit shall be set and the indicator shall be switched on.</p> <p>In case the indicator failure cycle is different to the operations cycle of the event, this means: If the indicator failure cycle counter threshold is configured to value N and the indicator failure cycle is started N times with the event being qualified as FAILED (i.e. TF is set), the WIR status bit shall be set and the indicator shall be switched on.</p> <p>Operation Cycles without any qualification will not influence this behavior.</p> <p>An Operation Cycle with only PASSED qualification(s) will reset the Indicator Failure Cycle Counter.</p> <p>This parameter is available only if DemIndicatorFailureCycleSource is set to DEM_FAILURE_CYCLE_INDICATOR.</p>	
Multiplicity	0..1	
Type	INTEGER	
Default value	1	
Range	<=255 >=1	
Configuration class	PreCompile:	VariantPreCompile



Origin	AUTOSAR_ECUC
Parameter Name	DemIndicatorFailureCycleSource
Label	Indicator Failure Cycle Source
Description	<p>This parameter defines, which failure cycle is used for the <code>WarningIndicatorOnCriteria</code> handling.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ <code>DEM_FAILURE_CYCLE_EVENT</code>: The event based failure cycle <code>DemEventFailureCycleRef</code> configured in <code>DemEventClass</code>, is used. Therefore, the parameters <code>DemIndicatorFailureCycleRef</code> and <code>DemIndicatorFailureCycleCounterThreshold</code> are not used for this indicator attribute of the event. ▶ <code>DEM_FAILURE_CYCLE_INDICATOR</code>: An indicator based failure cycle is used, defined by <code>DemIndicatorFailureCycleRef</code> and <code>DemIndicatorFailureCycleCounterThreshold</code>.
Multiplicity	1..1
Type	ENUMERATION
Default value	<code>DEM_FAILURE_CYCLE_EVENT</code>
Range	<code>DEM_FAILURE_CYCLE_EVENT</code> <code>DEM_FAILURE_CYCLE_INDICATOR</code>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemIndicatorHealingCycleCounterThreshold
Label	Indicator Healing Cycle Counter Threshold
Description	<p>Defines the number of healing cycles for the <code>WarningIndicatorOffCriteria</code>.</p> <p>If value 0 is selected, the behavior for healing is the same as for value 1.</p> <p>The <code>WarningIndicatorOffCriteria</code> processing is done at the beginning of the operation cycle to decrease run time of <code>Dem_Shutdown()</code> during ending of all operation cycles.</p>
Multiplicity	1..1
Type	INTEGER
Default value	1
Configuration class	VariantPreCompile: VariantPreCompile



Origin	AUTOSAR_ECUC
Parameter Name	DemIndicatorFailureCycleRef
Label	Indicator Failure Cycle
Description	<p>Kind of failure cycle for the indicator controlled by the respective event used for the <code>WarningIndicatorOnCriteria</code>.</p> <p>If the common operation cycle support is enabled (<code>DemCommonOperationCycleSupport = true</code>), this parameter (<code>DemIndicatorFailureCycleRef</code>) is disabled.</p> <p>This parameter is available only if <code>DemIndicatorFailureCycleSource</code> is set to <code>DEM_FAILURE_CYCLE_INDICATOR</code> and <code>DemCommonOperationCycleSupport</code> is disabled.</p> <p>If the operation cycle (<code>DemOperationCycleRef</code>) is of type <code>DEM_OPCYC_OBD_DCY</code>, the MIL indicator failure cycle must be the same.</p>
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	DemIndicatorHealingCycleRef
Label	Indicator Healing Cycle
Description	<p>Kind of healing cycle for the indicator controlled by the respective event used for the <code>WarningIndicatorOffCriteria</code>.</p> <p>If the common operation cycle support is enabled (<code>DemCommonOperationCycleSupport = true</code>), this parameter (<code>DemIndicatorHealingCycleRef</code>) is disabled.</p> <p>If the operation cycle (<code>DemOperationCycleRef</code>) is of type <code>DEM_OPCYC_OBD_DCY</code>, the MIL indicator healing cycle must be the same.</p>
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	DemIndicatorRef
Label	Indicator
Description	Reference to the used indicator.
Multiplicity	1..1



Type	REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.3.1.18. DemPidClass

Containers included		
Container name	Multiplicity	Description
DemPidDataElement	1..255	<p>This container contains the different data elements contained in the specific PID.</p> <p>Note: The order of data elements per PID in this configuration parameter must accord to the order of data elements configured in DcmDspPidDataPos. Otherwise the API Dem_DcmReadDataOfOBDFreezeFrame() may return wrong data elements.</p> <p>Note: The API Dem_GetNextFreezeFrameData with record number 0 will return The PidDataElements Byte aligned without padding bits. The bit order inside the Pid may be different in service 0x19 0x04 RecordNumber 0 (reportDTCSnapshotRecordByDTCNumber) than in Service 0x02 (Request powertrain freeze frame data), because the padding information is not available inside the Dem.</p>

Parameters included	
Parameter name	Multiplicity
DemPidIdentifier	1..1

Parameter Name	DemPidIdentifier
Label	PID Identifier
Description	<p><i>Identifier of the PID</i></p> <p>The PID Identifier 0xFF which is reserved by ISO (refer to J1979-DA:2014 TABLE B138) will be used as "PID disabled" calibration option. Therefore, PIDs with this identifier (0xFF) will not be reported via service 0x02, 0x19 and 0xAF. The configuration parameter DemPidIdentifier will be limited to [0x00..0xFE].</p>
Multiplicity	1..1
Type	INTEGER



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

5.3.1.19. DemPidDataElement

Parameters included	
Parameter name	Multiplicity
DemPidDataElementClassRef	1..1

Parameter Name	DemPidDataElementClassRef
Label	Pid Data Element Class
Description	This reference contains the link to a data element class. Both DemExternalCS-DataElementClass and DemExternalSRDataElementClass references are available, as internal data elements can not be referenced in the current implementation.
Multiplicity	1..1
Type	CHOICE-REFERENCE
Configuration class	VariantPreCompile:
Origin	AUTOSAR_ECU

5.3.1.20. DemAdvDisplacementPassiveTableLine

Containers included		
Container name	Multiplicity	Description
DemPassiveEventRefs	1..255	<i>List of references to DemEventParameter corresponding to an event currently stored in an event memory entry and subject for displacement.</i>

Parameters included	
Parameter name	Multiplicity
DemReportedEventRef	1..1

Parameter Name	DemReportedEventRef
Label	Reported Event Reference
Description	<i>Reference to the DemEventParameter corresponding to the event currently reported and trying to get an event memory entry.</i>



Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.3.1.21. DemPassiveEventRefs

Parameters included	
Parameter name	Multiplicity
DemPassiveEventRef	1..1

Parameter Name	DemPassiveEventRef
Label	Passive Event Reference
Description	<i>Reference to a DemEventParameter corresponding to the event currently stored in an event memory entry and subject for displacement.</i>
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

5.3.1.22. DemGeneral

Containers included		
Container name	Multiplicity	Description
DemClient	1..254	This container contains possible clients that are using the Dem APIs.
DemServiceAPI	1..1	Label: Service API Parameters Container for configuration of the service API of Dem.
DemBSWDistribution	1..1	This container contains the configuration parameters controlling BSW distribution of the Dem module on multi-core platforms. The configuration container is editable only if OsNumberOfCores exists and is greater than 1.



Containers included		
DemCallbackDTCStatusChanged	0..n	<p>The presence of this container indicates, that the Dem has access to a "DTCStatusChanged" callback, which the Dem will call to notify other components about the change in the status of a DTC.</p> <p>In case there is a <code>DemCallbackDTCStatusChangedFnc</code>, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no <code>DemCallbackDTCStatusChangedFnc</code>, the Dem will have an R-Port requiring the interface <code>CallbackDTCStatusChanged</code> whose name is generated by using the unique callback-prefix followed by the event name.</p>
DemDataElementClass	0..65500	<p>Label: Data Element Class</p> <p>This container contains the configuration (parameters) for an internal/external data element class.</p>
DemDidClass	0..65534	<p>This container contains the configuration (parameters) for a data Id class. It is assembled out of one or several data elements.</p>
DemEnableCondition	0..255	<p>This container contains the configuration (parameters) for enable conditions.</p>
DemEnableConditionGroup	0..254	<p>This container contains the configuration (parameters) for enable condition groups.</p> <p>A maximum of 254 enable condition groups can be configured, because the value of 255 is used for internal processing</p>
DemMultiEventTriggering	0..8190	<p>This container contains the configuration of a master event that will trigger other slave events whenever the master event is reported.</p> <p>If <code>DemMultiEventTriggeringAlternativeBehavior</code> is <code>DEM_MULTIEVENTTRIGGERING_ALT_NONE</code> the master event is processed along with slave events.</p> <p>If <code>DemMultiEventTriggeringAlternativeBehavior</code> is <code>DEM_MULTIEVENTTRIGGERING_ALT_EXCLUSIVE_REPLACE</code> the master event is not processed, except case when none slave event is enabled then only the master event is processed as it would be done if no <code>DemMultiEventTriggering</code> is configured for the master event.</p>



Containers included		
DemExtendedDataClass	0..256	This class contains the combinations of extended data records for an extended data class.
DemExtendedDataRecord-Class	0..253	This container contains the configuration (parameters) for an extended data record class. It is assembled out of one or several data elements
DemFreezeFrameClass	0..65534	This container contains the combinations of DIDs for a non OBD-relevant freeze frame class.
DemFreezeFrameRecNum-Class	0..255	This container contains a list of dedicated, different freeze frame record numbers assigned to an event. The order of record numbers in this list is assigned to the chronological order of the respective freeze frame records. This container is available for configured record numeration only (refer to DemTypeOfFreezeFrameRecordNumeration).
DemSPNClass	0..524287	Label: SPN Class This container contains the configuration (parameters) for a SPN.
DemJ1939FreezeFrameClass	0..254	This container contains the combinations of SPNs for a J1939 relevant freeze frame.
DemGeneralOBD	0..1	<i>The functionality related to this parameter is not supported by the current implementation.</i> This container contains the general OBD-specific configuration (parameters) of the Dem module.
DemGeneralJ1939	1..1	This container contains the general J1939-specific configuration (parameters) of the Dem module.
DemGroupOfDTC	0..30	This container contains the configuration (parameters) for DTC groups. At least the following DTC groups should exist: <ul style="list-style-type: none"> ▶ DEM_DTC_GROUP_EMISSION_REL_DTCS: with Group DTC Value = 0 ▶ DEM_DTC_GROUP_POWERTRAIN_DTCS ▶ DEM_DTC_GROUP_CHASSIS_DTCS ▶ DEM_DTC_GROUP_BODY_DTCS ▶ DEM_DTC_GROUP_NETWORK_COM_DTCS



Containers included		
		<p>Events which are not assigned to any DTC group will be deleted, if DEM_DTC_GROUP_ALL_DTCS is requested.</p> <p><i>Remark:</i> DEM_DTC_GROUP_ALL_DTCS must not be configured.</p>
DemIndicator	0..255	<p>This container contains the configuration (parameters) for Indicators.</p> <p>Note that this container definition does not explicitly define a symbolic name parameter. Instead, the short name of the container will be used in the Ecu Configuration Description to specify the symbolic name of the INDICATOR_NAME.</p>
DemNvRamBlockId	0..n	<p>NvM Block Configuration</p> <p>This container contains the configuration (parameters) for a non-volatile memory block reference, which is used from the Dem. If the permanent storage of event memory entries is required, at least the DEM_NVM_BLOCK_ID_DEFAULT reference (refer for details below) needs to be configured. Otherwise no NvM block and block reference need to be configured, so the data are stored only volatile. The Nv data will be stored in SEC_VAR_SAVED_ZONE, see R4.0.3 MemMap SWS: [MEMMAP022]</p> <p>Permanent storage is required in any of the following Dem configurations:</p> <ul style="list-style-type: none"> ▶ DemImmediateNvStorage is enabled for at least one event In this case DemCallbackMemStackMainFuncTrigger must be configured to call the memory stack main functions. ▶ Clear DTC Behavior is set to one of the NONVOLATILE options Also in this case DemCallbackMemStackMainFuncTrigger must be configured. ▶ Permanent memory entries are configured DemMaxNumberEventEntryPermanent <p>The number of blocks that are necessary depends on the implementation and configuration (e.g. number of used event memories) of the Dem module.</p>



Containers included

		<p>Service needs calculator</p> <p>The required NvM block references (DemNvRamBlockId's) need to be configured by the user (first). All required NvM blocks will be configured and referenced with DemNvRamBlockIdRef automatically, as per the current configuration of Dem. The user does not need to create the NvM blocks for Dem, nor configure these parameters of those created blocks again:</p> <ul style="list-style-type: none"> ▶ NvMBlockManagementType ▶ NvMBlockCrcType ▶ NvMBlockUseCrc ▶ NvMRamBlockDataAddress ▶ NvMBlockWriteProt ▶ NvMResistantToChangedSw ▶ NvMSelectBlockForReadAll ▶ NvMSelectBlockForWriteAll ▶ NvMWriteBlockOnce ▶ NvMBlockJobPriority <p>NvM Block Configurations</p> <p>For all NvM blocks, a block CRC is recommended.</p> <p>For all NvM blocks, the block length (depending on the Dem configuration) can be gathered from the map-file based on the RAM block variable.</p> <ul style="list-style-type: none"> ▶ DEM_NVM_BLOCK_ID_DEFAULT reference: One native/redundant NvM block is required, with <code>NvM_ReadAll()</code> and <code>NvM_WriteAll()</code> enabled. The RAM block data address shall be <code>&Dem_NvData</code>. In case DemClearDTCBehavior is set to one of the NONVOLATILE options, the NvM option NvM-BlockUseSyncMechanism has to be enabled and the callbacks <code>NvMReadRamBlockFromNvCallback</code>, <code>NvMWriteRam-BlockToNvCallback</code> and <code>NvMSingleBlockCallback</code> have to be configured.
--	--	---



Containers included

		<p>NvMReadRamBlockFromNvCallback shall be <code>Dem_-NvMReadCopyEventStatusBlock</code>.</p> <p>NvMWWriteRamBlockToNvCallback shall be <code>Dem_-NvMWWriteCopyEventStatusBlock</code>. The NvM single block callback shall be <code>Dem_NvMEventStatusBlockCallback</code> configured manually.</p> <ul style="list-style-type: none"> ▶ DEM_NVM_BLOCK_ID_PRIMARY reference: One dataset NvM block shall be configured, with <code>NvM_ReadAll()</code> and <code>NvM_WriteAll()</code> disabled. The RAM block data address shall be <code>&Dem_NvGateEntryPrimaryData</code>. The NvM single block callback shall be <code>Dem_NvMGateEntryPrimaryBlockCallback</code>. ▶ DEM_NVM_BLOCK_ID_SECONDARY reference: One dataset NvM block shall be configured if there is at least one event with origin <code>DEM_DTC_ORIGIN_SECONDARY_MEMORY</code>, with <code>NvM_ReadAll()</code> and <code>NvM_WriteAll()</code> disabled. The RAM block data address shall be <code>&Dem_NvGateEntrySecondaryData</code>. The NvM single block callback shall be <code>Dem_NvMGateEntrySecondaryBlockCallback</code>. ▶ DEM_NVM_BLOCK_ID_MIRROR reference: One dataset NvM block shall be configured if there is at least one event with origin <code>DEM_DTC_ORIGIN_MIRROR_MEMORY</code>, with <code>NvM_ReadAll()</code> and <code>NvM_WriteAll()</code> disabled. The RAM block data address shall be <code>&Dem_NvGateEntryMirrorData</code>. The NvM single block callback shall be <code>Dem_NvMGateEntryMirrorBlockCallback</code>. ▶ DEM_NVM_BLOCK_ID_PERMANENT reference: One native/redundant NvM block shall be configured if the permanent memory for OBD is used (<code>DemMaxNumberEventEntryPermanent > 0</code>), with <code>NvM_ReadAll()</code> enabled and <code>NvM_WriteAll()</code> disabled. The explicit synchronization mechanism shall be enabled for this NvM block. The RAM block data address is not used.
--	--	--



Containers included		
		<p>The NvM block size shall be (<code>DemMaxNumberEventEntryPermanent * 2</code>) bytes.</p> <p>The NvM single block callback shall be <code>Dem_-NvMWriteFinishedPermanentMemory</code>.</p> <p><code>NvMReadRamBlockFromNvCallback</code> shall be <code>Dem_-NvMReadCopyPermanentMemory</code>.</p> <p><code>NvMWriteRamBlockToNvCallback</code> shall be <code>Dem_-NvMWriteCopyPermanentMemory</code>.</p> <p>For more information on the NvM configuration, see the <i>AutoCore Generic Memory Stack documentation</i> and consider the Dem integration issues in the <i>AutoCore Generic Diagnostic Stack documentation</i>.</p>
DemOperationCycle	1..255	Note that this container definition does not explicitly define a symbolic name parameter. Instead, the short name of the container will be used in the Ecu Configuration Description to specify the symbolic name of the operation cycle name.
DemAgingCycle	0..256	<p>Note that this container definition does not explicitly define a symbolic name parameter. Instead, the short name of the container will be used in the Ecu Configuration Description to specify the symbolic name of the aging cycle name. These aging cycles are reported via API <code>Dem_SetAgingCyclesState</code> only.</p> <p>Note that when <code>DemExtendedDataCapture</code> and <code>DemFreezeFrameCapture</code> are set to <code>DEM_TRIGGER_ON_FDC_-THRESHOLD</code> and <code>DemAgingAllowed</code> is enabled, aging through aging cycles is not allowed therefore it is meaningless to configure aging cycles if the above preconditions are met.</p>
DemRatioid	0..65535	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This container contains the OBD specific ratio Id configuration. It is related to a specific event, a FID, and an IUMPR group.</p> <p>Note that this container definition does not explicitly define a symbolic name parameter. Instead, the short name of the container will be used in the Ecu Configuration Description to specify the symbolic name of the ratio Id name.</p>



Containers included		
DemEventStatusPort	1..1	This container is used for grouping the parameters used for setting the queue length of the Client/Server ComSpec Operations for the EventStatus Port.
DemOpCyclePort	1..1	This container is used for grouping the parameters used for setting the queue length of the Client/Server ComSpec Operations for the OpCycle Port.
DemAgingCyclePort	1..1	This container is used for grouping the parameters used for setting the queue length of the Client/Server ComSpec Operations for the AgingCycle Port.
DemEnableCondPort	1..1	This container is used for grouping the parameters used for setting the queue length of the Client/Server ComSpec Operations for the EnableCond Port.
DemStorageCondPort	1..1	This container is used for grouping the parameters used for setting the queue length of the Client/Server ComSpec Operations for the StorageCond Port.
DemIndStatusPort	1..1	This container is used for grouping the parameters used for setting the queue length of the Client/Server ComSpec Operations for the IndStatus Port.
DemControlDTCSuppression-Port	1..1	This container is used for grouping the parameters used for setting the queue length of the Client/Server ComSpec Operations for the ControlDTCSuppression Port.
DemCddIfPort	1..1	This container is used for grouping the parameters used for setting the queue length of the Client/Server ComSpec Operations for the CddIf Port.
DemResetReadinessPort	1..1	This container is used for grouping the parameters used for setting the queue length of the Client/Server ComSpec Operations for the ResetReadiness Port.
DemPowerTakeOffStatusPort	1..1	This container is used for grouping the parameters used for setting the queue length of the Client/Server ComSpec Operations for the PowerTakeOffStatus Port.
DemEventPort	1..1	This container is used for grouping the parameters used for setting the queue length of the Client/Server ComSpec Operations for the Event Port.
DemEventInfoPort	1..1	This container is used for grouping the parameters used for setting the queue length of the Client/Server ComSpec Operations for the EventInfo Port.



Containers included

DemPfcCycleQualifiedPort	1..1	This container is used for grouping the parameters used for setting the queue length of the Client/Server ComSpec Operations for the PfcCycleQualified Port.
DemOverflowIndPort	1..1	This container is used for grouping the parameters used for setting the queue length of the Client/Server ComSpec Operations for the OverflowInd Port.
DemStorageCondition	0..255	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This container contains the configuration (parameters) for storage conditions.</p>
DemStorageConditionGroup	0..255	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This container contains the configuration (parameters) for storage condition groups.</p>

Parameters included

Parameter name	Multiplicity
DemAgingCycleCounterProcessing	1..1
DemAgingRequiresTestedCycleSetOperationCycleState	1..1
DemAgingRequiresTestedCycleSetAgingCycleState	1..1
DemAgingBasedOnHealingCompletionSupport	1..1
DemBswErrorBufferSize	1..1
DemMaxHandledPassedEventsPerScheduling	0..1
DemMaxHandledFailedEventsPerScheduling	0..1
DemCallbackEventDataChangedASR403Behaviour	1..1
DemCallbackEventStatusChangedOpCycStartSupport	1..1
DemCalloutDynamicDTCFnc	0..1
DemClearDTCBehavior	1..1
DemMaxNumberClearEventsPerCycle	1..1
DemSuppressionSupport	1..1
DemAvailabilitySupport	1..1
DemDebounceCounterBasedSupport	1..1
DemDebounceTimeBasedSupport	1..1



Parameters included

DemDebounceFrequencyBasedSupport	1..1
DemMaxNumberDebCounterClasses	1..1
DemMaxNumberDebTimeClasses	1..1
DemDevErrorDetect	1..1
DemRteUsage	1..1
DemDcmUsage	1..1
DemGetDTCByOccurrenceTimeSupport	1..1
DemDtcStatusAvailabilityMask	1..1
DemEnableConditionSupport	1..1
DemEventCombinationSupport	1..1
DemEventDisplacementSupport	1..1
DemEventDisplacementStrategy	1..1
DemEventDisplacementCriterionTNCTOC	1..1
DemAdvDisplacementOcc1Limit	1..1
DemExtendedDataCapture	1..1
DemFreezeFrameCapture	1..1
DemOperationCycleProcessing	1..1
DemHeaderFileInclusion	0..n
DemImmediateNvStorageLimit	0..1
DemNvStorageEmptyEventMemoryEntries	1..1
DemUserDefMemoryId	0..1
DemMaxNumberEventEntryMirror	1..1
DemMaxSizeFreezeFrameEntryMirror	1..1
DemMaxNumberEventEntryPermanent	1..1
DemMaxNumberEventEntryPrimary	1..1
DemMaxSizeFreezeFrameEntryPrimary	1..1
DemMaxNumberEventEntrySecondary	1..1
DemMaxSizeFreezeFrameEntrySecondary	1..1
DemMaxNumberPrestoredFF	1..1
DemOBDSupport	1..1
DemOBDIumprFunctionsEnabled	1..1



Parameters included

DemCalibrationSupport	1..1
DemClearDTCOfDisabledEvents	1..1
DemCommonOperationCycleSupport	1..1
DemAgingAllowedSeperateFlag	1..1
DemMaxNumberEventClasses	1..1
DemMaxNumberIndicatorClasses	1..1
DemOBDSupportKind	1..1
DemOccurrenceCounterProcessing	1..1
DemOperationCycleStatusStorage	1..1
DemPTOSupport	1..1
DemStatusBitHandlingTestFailedSinceLastClear	1..1
DemResetPendingBitOnOverflow	1..1
DemUpdateOccOrderOnEventEntryUpdate	1..1
DemResetConfirmedBitOnOverflow	1..1
DemStatusBitStorageTestFailed	1..1
DemStorageConditionSupport	1..1
DemImmediateStorageInternalEvents	1..1
DemIntermediateNvStorageOfDTCStatus	1..1
DemTaskTime	1..1
DemTriggerDcmReports	1..1
DemTriggerDltReports	1..1
DemTriggerFiMReports	1..1
DemSupportFiMFDCThreshold	1..1
DemTriggerMonitorInitBeforeClearOk	1..1
DemCallbackInitMForEReenabledSupport	1..1
DemMaxHandledInitMonitorReenabledPerScheduling	1..1
DemTypeOfDTCSupported	1..1
DemTypeOfFreezeFrameRecordNumeration	1..1
DemCommonFFDataClassRef	0..1
DemDevFreezeFrameSupport	1..1
DemCommonFFDataDevAClassRef	0..1



Parameters included

DemMaxNumberDevFreezeFrameAEntry	1..1
DemMaxSizeDevFreezeFrameAEntry	1..1
DemMaxNumberDevFreezeFrameBEntry	1..1
DemMaxSizeDevFreezeFrameBEntry	1..1
DemStoreInternalEvents	1..1
DemVersionInfoApi	1..1
DemMILIndicatorRef	0..1
DemMaxNumEventsPerReadinessGroup	1..1
DemJ1939Support	1..1
DemAmberWarningLampIndicatorRef	0..1
DemProtectLampIndicatorRef	0..1
DemRedStopLampIndicatorRef	0..1
DemOBDForceMisfireMonitoringReady	1..1
DemOBDForceFuelSysMonitoringReady	1..1
DemOBDForceCCMonitoringReady	1..1
DemNvDataConfigSignatureUsed	1..1
DemNvDataConfigSignatureCalcSet	1..1
DemCallbackMemStackMainFuncTrigger	1..1
DemUserControlledWIRBitSupport	1..1
DemDataElementDefaultEndianness	1..1
DemIncludeProjectSpecificCustomizationSupport	1..1

Parameter Name	DemAgingCycleCounterProcessing
Label	Aging Cycle Counter Processing
Description	<p>This configuration switch defines, whether the aging counter is calculated Dem- internally or provided via <code>Dem_SetAgingCycleCounterValue</code>.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ <code>DEM_PROCESS_AGINGCTR_EXTERN</code>(not supported): based on API <code>Dem_SetAgingCycleCounterValue</code>. Not supported, because the feature was removed from AUTOSAR R4.3.0. ▶ <code>DEM_PROCESS_AGINGCTR_INTERN</code>: based on reported cycle states.
Multiplicity	1..1



Type	ENUMERATION	
Default value	DEM_PROCESS_AGINGCTR_INTERN	
Range	DEM_PROCESS_AGINGCTR_EXTERN DEM_PROCESS_AGINGCTR_INTERN	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DemAgingRequiresTestedCycleSetOperationCycleState	
Label	Aging Requires Tested Cycle For Dem_SetOperationCycleState	
Description	<p>This configuration switch defines whether the aging cycle counter is processed for every call of aging cycle using the Dem_SetOperationCycleState() API or if only tested aging cycle are considered.</p> <ul style="list-style-type: none"> ▶ <code>true</code> : The aging counter shall be updated only if the TNCTOC and TFTOC bits are cleared. ▶ <code>false</code> : The aging counter shall be updated only if TFTOC and TF bits are not set. <p>Note: This configuration parameter is derived from the AUTOSAR parameter DemAgingRequieresTestedCycle.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemAgingRequiresTestedCycleSetAgingCycleState	
Label	Aging Requires Tested Cycle For Dem_SetAgingCycleState	
Description	<p>This configuration switch defines whether the aging cycle counter is processed for every call of aging cycle using the Dem_SetAgingCycleState() API or if only tested aging cycle are considered.</p> <ul style="list-style-type: none"> ▶ <code>true</code> : The aging counter shall be updated only if the TNCTOC and TFTOC bits are cleared. ▶ <code>false</code> : The aging counter shall be updated only if TF bit is not set. <p>Note: This configuration parameter is derived from the AUTOSAR parameter DemAgingRequieresTestedCycle.</p>	



Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	
Parameter Name	DemAgingBasedOnHealingCompletionSupport	
Label	Aging Based on Healing Completed Support	
Description	<p>If enabled, aging processing shall not be started before healing is completed, in accordance with AUTOSAR R4.2.2.</p> <ul style="list-style-type: none"> ▶ <code>true</code> = The aging counter shall be updated only after healing is completed (R4.2.2). ▶ <code>false</code> = The aging counter shall be updated in parallel with healing (R4.0.-3). <p><i>Feature enabled:</i></p> <p>The aging counter initialization will not be impacted by this feature. The aging process will be delayed with the time necessary for the Warning Indicator Requested healing.</p> <p>If an event refers to an Extended Data Class which is made of Data Element Class of type <code>DemInternalDataElementClass</code> having assigned <code>DEM_AGINGCTR</code>, the value won't start changing unless the WIR bit is healed.</p> <p><i>Feature disabled:</i></p> <p>The effect of the enabled feature is also achievable by extending the Aging Cycle Counter Threshold of an event with its Indicator Healing Cycle Counter Threshold. Take into account that <code>DEM_AGINGCTR</code> will report a maximum value of 255, which can be different than 0 even if the WIR bit is not healed.</p> <p><i>Constraints:</i></p> <p>Aging based on healing completed is not allowed when <code>DemFreezeFrameCapture</code> and/or <code>DemExtendedDataCapture</code> are set to <code>DEM_TRIGGER_ON_FDC_THRESHOLD</code>.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemBswErrorBufferSize	
Label	Error-Queue Buffer Size	
Description	<p>Maximum number of elements in internal error-queue for handling of BSW event reports.</p> <p><i>Additional extensions of this version:</i></p> <p>After Dem is initialized (<code>Dem_Init</code>), this queue is also used to hold SW-C event reports to be entered into the event memory by the Dem main function.</p> <p>After Dem is shut down (<code>Dem_Shutdown</code>), this queue is still used to hold BSW event reports to be processed by the Dem main function.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	10	
Range	<p><=255</p> <hr/> <p>>=1</p>	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemMaxHandledPassedEventsPerScheduling	
Label	Schedule passed events max	
Description	The maximum number of passed events in the error-queue processed per <code>Dem_MainFunction</code> call. If this maximum is reached the <code>Dem_MainFunction</code> suspends the handling of the error-queue and continues in the next call.	
Multiplicity	0..1	
Type	INTEGER	
Range	<p><=255</p> <hr/> <p>>=1</p>	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	
Parameter Name	DemMaxHandledFailedEventsPerScheduling	
Label	Schedule failed events max	



Description	The maximum number of failed/unconfirmed events in the error-queue processed per Dem_MainFunction call. If this maximum is reached the Dem_MainFunction suspends the handling of the error-queue and continues in the next call.	
Multiplicity	0..1	
Type	INTEGER	
Range	<=255 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemCallbackEventDataChangedASR403Behaviour	
Label	EventDataChanged Callback Enable AUTOSAR 4.0.3 Behavior	
Description	<p>This configuration switch defines whether the "EventDataChanged" callback triggering behavior will be done according to ASR4.0.3 or not. Enables AUTOSAR 4.0.3 usage functionality.</p> <p>Note: This configuration switch has default value set to "true", which will ensure backward compatibility and trigger the callback according to ASR4.0.3.</p> <ul style="list-style-type: none"> ▶ true: EventDataChanged callback will be triggered according to ASR4.0.3. ▶ false: EventDataChanged callback will be triggered according to ASR4.2.-1. According to ASR4.2.1 the number of triggering point for the callback are less than ASR4.0.3 (EB interpretation) 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemCallbackEventStatusChangedOpCycStartSupport	
Label	Event Status Changed Callbacks for Start of the Operation Cycle Support	
Description	<p>This configuration switch defines whether the "EventStatusChanged/DTCStatusChanged" callbacks are called from within Dem_SetOperationCycleState() on state DEM_CYCLE_STATE_START.</p> <ul style="list-style-type: none"> ▶ true : The Dem_SetOperationCycleState() with state DEM_CYCLE_STATE_START will trigger the status changed callbacks. 	



	<ul style="list-style-type: none"> ▶ <code>false</code>: The <code>Dem_SetOperationCycleState()</code> with state <code>DEM_CYCLE_STATE_START</code> will not trigger the status changed callbacks. This option offers the user the possibility to delete the trigger point for "EventStatusChanged/DTCStatusChanged" callbacks reducing so the execution time. Such an option is of interest, for projects where the application does not need to know the evolution of the DTC status for events assigned to the power cycle. <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ <code>DemTriggerFiMReports</code> ▶ <code>DemTriggerDcmReports</code> ▶ <code>DemCallbackDTCStatusChanged</code> ▶ <code>DemCallbackEventStatusChanged</code>
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	PreCompile: <input type="text" value="VariantPreCompile"/>
Origin	Elektrobit Automotive GmbH

Parameter Name	<code>DemCalloutDynamicDTCFnc</code>
Label	'Get Dynamic DTC value' Callout C-Function
Description	<p>If this parameter is configured, the Dem gets the UDS DTC value from the configured callout function dynamically instead of using the configured UDS DTC value.</p> <p>The function must have a prototype compatible to:</p> <pre>extern uint32 Appl_GetDynamicDTC(Dem_EventIdType EventId, uint32 DTC);</pre> <p>which is declared in <code>Dem_Cfg.h</code>.</p> <p>The returned DTC value</p> <ul style="list-style-type: none"> ▶ shall not be equal to 0x000000 ▶ shall be in the same group as the original DTC value ▶ shall be unique <p>See also the integration guideline in the Diagnostic Stack user's guide.</p>
Multiplicity	0..1



Type	FUNCTION-NAME	
Default value	Appl_GetDynamicDTC	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemClearDTCBehavior	
Label	Clear DTC Behavior	
Description	<p>Defines the clearing process of diagnostic information for volatile and non-volatile memory and the positive response handling for the Dcm module.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_CLRRESP_NONVOLATILE_FINISH: Return DEM_CLEAR_OK after volatile and non-volatile event memory data cleared. ▶ DEM_CLRRESP_NONVOLATILE_TRIGGER: Return DEM_CLEAR_OK after volatile event memory data cleared and non-volatile event memory clearing is triggered. ▶ DEM_CLRRESP_VOLATILE: Return DEM_CLEAR_OK after volatile event memory data cleared. <p>If any of the NONVOLATILE behaviors is selected, the event memory is switched from one large NvM block to a dataset (refer to DemNvRamBlockId) being able to store each event entry separately.</p> <p>If DemClearDTCBehavior is set to DEM_CLRRESP_NONVOLATILE_FINISH no monitor is allowed to trigger a write operation of an NvM block with a higher priority than the priorities of the Dem NvM blocks, if this write operation could lead to a conflict with status data of not finally deleted DTC.</p>	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	DEM_CLRRESP_VOLATILE	
Range	DEM_CLRRESP_NONVOLATILE_FINISH DEM_CLRRESP_NONVOLATILE_TRIGGER DEM_CLRRESP_VOLATILE	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemMaxNumberClearEventsPerCycle	
----------------	--	--



Label	Maximum number of cleared events per cycle
Description	<p>This parameter defines the maximum number of events which are processed per Dem_MainFunction() cycle on a ClearDTC request. It affects all ClearDTC interfaces.</p> <p>A large value for DemMaxNumberClearEventsPerCycle will lead to an increased run time for Dem_MainFunction() while processing the ClearDTC request.</p> <p>Very small DemMaxNumberClearEventsPerCycle values may noticeable increase the total execution time of a ClearDTC request.</p> <p>By configuring this parameter equal to the number of events of the largest event combination, it is ensured that all events of a combination will be cleared consistently in a single Dem_MainFunction() cycle.</p>
Multiplicity	1..1
Type	INTEGER
Default value	5
Range	<p><=1023</p> <hr/> <p>>=1</p>
Configuration class	<p>VariantPreCompile: VariantPreCompile</p>
Origin	Elektrobit Automotive GmbH

Parameter Name	DemSuppressionSupport
Label	DTC Suppression Support
Description	<p>This configuration switch defines whether support for dynamic DTC suppression is enabled or not.</p> <p>A suppressed DTC is not visible by the tester but can be processed continuously by the diagnostic monitor.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_NO_SUPPRESSION: DTC suppression is not supported. ▶ DEM_DTC_SUPPRESSION: DTC suppression is supported. <p>If enabled, Dem_SetDTCSuppression() can be used to dynamically change the DTC suppression status at run-time. The DTC suppression status is stored volatile, that is, the desired status needs to be set again within each power cycle.</p>



	<p>A DTC cannot be suppressed if event data for the related event or combined events exist. This condition is only checked for DTCs which are not already suppressed.</p> <p>A DTC can be suppressed even if the DTC record update was disabled via Dem_DisableDTCRecordUpdate. The DTC record protection is not implicitly released if a DTC is being suppressed.</p> <p>A suppressed DTC is not visible for Dem_ClearDTC() and therefore cannot be deleted. There is one exception: if DemClearDTCOfDisabledEvents is enabled, a ClearDTC() request for DEM_DTC_GROUP_ALL_DTCs also deletes events which are mapped to suppressed DTCs.</p> <p>Note: Dem_GetDTCOfEvent() returns the DTC value that is mapped to an event Id by DTC configuration or the re-calibrated/dynamically changed DTC value respectively (see DemCalloutDynamicDTCFnc). The DTC value is returned regardless of its current suppression status.</p> <p>Note: The DTC suppression status is associated with the related event, not with the DTC value. If the DTC value is changed via calibration or side allocation call-out function, the suppression status of the DTC is not impacted i.e. if the original DTC was suppressed, the changed DTC is suppressed as well.</p>
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_NO_SUPPRESSION
Range	DEM_NO_SUPPRESSION DEM_DTC_SUPPRESSION
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemAvailabilitySupport
Label	Event Availability Support
Description	<p>This configuration switch defines whether support for dynamic availability of events is enabled or not.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_NO_AVAILABILITY: Dynamic event availability is not supported. ▶ DEM_EVENT_AVAILABILITY: Dynamic event availability is supported.



	<p>If enabled, the value of DemEventAvailable is used as initial value for event availability and may be changed dynamically via Dem_SetEventAvailable().</p> <p>The event available status is stored volatile, that is, the desired status needs to be set again within each power cycle.</p> <p>If an event is set to unavailable, the corresponding event is treated as if it is not configured in the system.</p> <p>An event can only be set to unavailable if no event data for the event exists and none of the 'TF', 'PDTC', 'CDTC' or 'WIR' event status flags is set.</p> <p>Note: If DemAvailabilitySupport is enabled, DemEventAvailable can still be changed via calibration. However, in this case, the calibrated value takes effect not before next power cycle.</p>
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_NO_AVAILABILITY
Range	DEM_NO_AVAILABILITY DEM_EVENT_AVAILABILITY
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemDebounceCounterBasedSupport
Label	Debounce Counter Based Support
Description	<p>This configuration switch defines whether support for counter-based debouncing is enabled or not.</p> <ul style="list-style-type: none"> ▶ true: counter-based debouncing support is enabled ▶ false: counter-based debouncing support is disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemDebounceTimeBasedSupport
Label	Debounce Time Based Support



Description	This configuration switch defines whether support for time-based debouncing is enabled or not.
	<ul style="list-style-type: none"> ▶ <code>true</code>: time-based debouncing support is enabled ▶ <code>false</code>: time-based debouncing support is disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DemDebounceFrequencyBasedSupport
Label	Debounce Frequency Based Support
Description	This configuration switch defines whether support for frequency based debouncing is enabled or not.
	<ul style="list-style-type: none"> ▶ <code>true</code>: frequency based debouncing support is enabled ▶ <code>false</code>: frequency based debouncing support is disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemMaxNumberDebCounterClasses
Label	Number of Debounce Counter Based Classes
Description	<p>This value is only needed for calibration (without event combination).</p> <p>Specifies how many instances of the class DemDebounceCounterBased should be allocated at least.</p> <p>If less events are configured with DemDebounceCounterBased than this value, this value specifies the number of available DemDebounceCounterBasedClasses.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DemDebounceCounterBasedSupport: must be set to <code>true</code> to enable support for <i>Number of Debounce Counter Based Classes</i>.



	<ul style="list-style-type: none"> ▶ DemCalibrationSupport: must be set to <code>true</code> to enable support for <i>Number of Debounce Counter Based Classes</i>. ▶ DemEventCombinationSupport: must be disabled (<code>DEM_EVCOMB_DISABLED</code>) to enable support for <i>Number of Debounce Counter Based Classes</i>.
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<=65535
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemMaxNumberDebTimeClasses	
Label	Number of Debounce Time Based Classes	
Description	<p>This value is only needed for calibration (without event combination).</p> <p>Specifies how many instances of the class DemDebounceTimeBased should be allocated at least.</p> <p>If less events are configured with DemDebounceTimeBased than this value, this value specifies the number of available DemDebounceTimeBasedClasses.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DemDebounceTimeBasedSupport: must be set to <code>true</code> to enable support for <i>Number of Debounce Time Based Classes</i>. ▶ DemCalibrationSupport: must be set to <code>true</code> to enable support for <i>Number of Debounce Time Based Classes</i>. ▶ DemEventCombinationSupport: must be disabled (<code>DEM_EVCOMB_DISABLED</code>) to enable support for <i>Number of Debounce Time Based Classes</i>. 	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535	
Configuration class	PreCompile: VariantPreCompile	
Origin	Elektrobit Automotive GmbH	



Parameter Name	DemDevErrorDetect	
Label	Enable Development Error Detection	
Description	Activate/Deactivate the Development Error Detection and Notification. <ul style="list-style-type: none"> ▶ true: Development Error Detection and Notification activated ▶ false: Development Error Detection and Notification deactivated 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemRteUsage	
Label	Enable Rte Usage	
Description	This parameter enables the usage of the RTE for this module. For an easy integration it is recommended to disable the usage of the RTE at the beginning of the integration work.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemDcmUsage	
Label	Enable Dcm Usage	
Description	This parameter enables the usage of the Dcm. In case this parameter is set to false: <ul style="list-style-type: none"> ▶ Dem does not provide the API functions <code>Dem_DcmGetInfoTypeValue08()</code> and <code>Dem_DcmGetInfoTypeValue0B()</code> ▶ Dem does not include <code>Dcm_Types.h</code> 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile



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

Parameter Name	DemGetDTCByOccurrenceTimeSupport	
Label	Implement Dem_GetDTCByOccurrenceTime()	
Description	<p>Configuration switch defining whether support for retrieving stored DTCs according to the relevant occurrence time, specified as an input parameter for the API Dem_GetDTCByOccurrenceTime() via service 0x19, exists or not. This functionality applies to event IDs which belong to the primary memory while the event is qualified as FAILED for a number of times necessary to set the TF and CDTC status bits.</p> <ul style="list-style-type: none"> ▶ true: Dem_GetDTCByOccurrenceTime() exists and will return the DTC value in UDS format identified by the type of the occurrence time request. ▶ false: Dem_GetDTCByOccurrenceTime() does not exist. All functionality related to this API (e.g. the prototype, the definition of the function) are removed at pre-compile time. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemDtcStatusAvailabilityMask	
Label	Status Availability Mask	
Description	Mask for the supported DTC status bits by the Dem. This mask is used by UDS service 0x19.	
Multiplicity	1..1	
Type	INTEGER	
Default value	255	
Range	<=255 >=0	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemEnableConditionSupport	
Label	Enable Condition Support	



Description	This configuration switch defines whether support for enable conditions is enabled or not.	
	<ul style="list-style-type: none"> ▶ true: support for enable conditions is enabled ▶ false: support for enable conditions is disabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DemEventCombinationSupport
Label	Event Combination Support
Description	<p>This parameter defines the type of event combination supported by the Dem. Events can be combined by referring the same DTC in DemDTCClassRef.</p> <p>NOTE: If Event Combination is enabled, the following features are not supported:</p> <ul style="list-style-type: none"> ▶ Prestorage of freeze frames ▶ Development freeze frames <p>NOTE: If Event Combination is enabled the calibration schema differs. For details see DemCalibrationSupport.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_EVCOMB_DISABLED: Event combination not used. ▶ DEM_EVCOMB_TYPE1: The combined DTC is stored and updated in a single event memory entry. ▶ DEM_EVCOMB_TYPE2 (not supported): Each event is stored in a separate event memory location.
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_EVCOMB_DISABLED
Range	<hr/> DEM_EVCOMB_DISABLED <hr/> DEM_EVCOMB_TYPE1 <hr/> DEM_EVCOMB_TYPE2



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DemEventDisplacementSupport	
Label	Event Displacement Support	
Description	<p>This configuration switch defines whether support for event displacement is enabled or not.</p> <ul style="list-style-type: none"> ▶ <code>true</code>: event displacement support is enabled ▶ <code>false</code>: event displacement support is disabled <p>The displacement behavior is modified in detail by a number of other configuration parameters:</p> <ul style="list-style-type: none"> ▶ DemEventDisplacementStrategy ▶ DemEventDisplacementCriterionTNCTOC ▶ DemOBDEventDisplacement ▶ DemExtendedDataCapture/DemFreezeFrameCapture <p>Follow the links for a detailed description.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DemEventDisplacementStrategy	
Label	Event Displacement Strategy	
Description	<p>This configuration switch defines, which displacement strategy is followed.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ <code>DEM_DISPLACEMENT_PRIO_OCC</code>: Event memory entry displacement is enabled, by consideration of priority and occurrence (but without active/passive status). ▶ <code>DEM_DISPLACEMENT_FULL</code>: Event memory entry displacement is enabled, by consideration of priority, active/passive status, and occurrence. 	



	The overall support for event memory displacement is enabled/disabled by the configuration parameter DemEventDisplacementSupport. There is an overview of all configuration parameters available which affect the displacement behavior.
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_DISPLACEMENT_FULL
Range	DEM_DISPLACEMENT_PRIO_OCC DEM_DISPLACEMENT_FULL
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemEventDisplacementCriterionTNCTOC
Label	Event Displacement Criterion TNCTOC
Description	<p>This configuration switch activates/deactivates the ASR4.3.1 displacement behavior for events which are not tested in this operation cycle (TestNotCompletedThisOperationCycle bit == 1).</p> <p>When this switch is activated, existing events which are not tested in this operation cycle (TNCTOC bit == 1) and with the same priority as the new event will be considered during the displacement mechanism.</p> <ul style="list-style-type: none"> ▶ true: Event status bit TNCTOC is considered during the displacement mechanism in accordance with ASR4.3.1. ▶ false: Event status bit TNCTOC is not considered during the displacement mechanism (ASR4.0.3 compliant behavior). <p>If the vendor-specific extension <code>DEM_TRIGGER_ON_FDC_THRESHOLD</code> is selected for DemExtendedDataCapture and DemFreezeFrameCapture, a non-standard/vendor-specific displacement algorithm is activated. If so, it is not recommended to enable DemEventDisplacementCriterionTNCTOC because event entries with the same priority as the new event will be considered for displacement in this case.</p> <p>The overall support for event memory displacement is enabled/disabled by the configuration parameter DemEventDisplacementSupport. There is an overview of all configuration parameters available which affect the displacement behavior.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	false



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	
Parameter Name	DemAdvDisplacementOcc1Limit	
Label	Advanced Displacement OCC1 Limit	
Description	<p>The number of OCC1 counts used for selecting entry by advanced displacement algorithm.</p> <p>This container is available only, if the following is configured:</p> <ul style="list-style-type: none"> ▶ DemEventDisplacementSupport = 'true' ▶ DemEventDisplacementStrategy == DEM_DISPLACEMENT_FULL ▶ DemExtendedDataCapture == DEM_TRIGGER_ON_FDC_THRESHOLD (because Si30 is available only in this mode) 	
Multiplicity	1..1	
Type	INTEGER	
Default value	3	
Range	<p><=255</p> <p>>=1</p>	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	
Parameter Name	DemExtendedDataCapture	
Label	Extended Data Capture	
Description	<p>This parameter defines the point in time, when the extended data collection is done for the initial event memory entry.</p> <p>DemExtendedDataCapture must be configured with the same value as DemFreezeFrameCapture.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_TRIGGER_EVENT_MEMORY_STORAGE: Triggers the collection of extended data if the event is stored in fault memory. ▶ DEM_TRIGGER_TESTFAILED : Triggers the collection of extended data if the UDS DTC status bit 0 (TestedFailed) changes from 0 to 1. ▶ DEM_TRIGGER_TESTFAILED is only effectual for SWC events, BSW events will still be handled as DEM_TRIGGER_EVENT_MEMORY_STORAGE. 	



	<ul style="list-style-type: none"> ▶ <code>DEM_TRIGGER_ON_FDC_THRESHOLD</code> (vendor-specific extension): Triggers the collection of extended data if unconfirmedDTC threshold is reached (refer to <code>DemDebounceCounterUnconfirmedThreshold</code>) for the first time in an operation cycle. This changes the behavior of the following functionalities: <ul style="list-style-type: none"> ▶ A different, non-standard AUTOSAR, displacement algorithm is used (when enabled), i.e. it uses the OCC2 counter instead of the internal calculated occurrence order. It is recommended to not enable the standard AUTOSAR event displacement criteria <code>DemEventDisplacement-CriterionTNCTOC</code> and <code>DemOBDEventDisplacement</code>. The overall support for event memory displacement is enabled/disabled by the configuration parameter <code>DemEventDisplacementSupport</code>. There is an overview of all configuration parameters available which affect the displacement behavior. ▶ Development freeze frames of new events (when enabled) will displace development freeze frames of aged events, if the new event status reaches the unconfirmedDTC threshold and the development freeze frame memory space is full already. ▶ Aging and healing algorithms (when enabled) use vendor specific algorithms (otherwise use standard AUTOSAR algorithms).
Multiplicity	1..1
Type	ENUMERATION
Default value	<code>DEM_TRIGGER_EVENT_MEMORY_STORAGE</code>
Range	<hr/> <code>DEM_TRIGGER_EVENT_MEMORY_STORAGE</code> <hr/> <code>DEM_TRIGGER_TESTFAILED</code> <hr/> <code>DEM_TRIGGER_ON_FDC_THRESHOLD</code>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemFreezeFrameCapture
Label	Freeze Frame Capture
Description	<p>This parameter defines the point in time, when the freeze frame data collection is done for the initial event memory entry.</p> <p><code>DemFreezeFrameCapture</code> must be configured with the same value as <code>DemExtendedDataCapture</code>.</p> <p>Based on the value selected, it will impact the mechanism of OBD Freeze Frame reporting. (See <code>DemPidClass</code> for more details regarding OBD Freeze Frame.)</p>



	<p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_TRIGGER_EVENT_MEMORY_STORAGE: Triggers the collection of freeze frame data if the event is stored in fault memory. ▶ DEM_TRIGGER_TESTFAILED : Triggers the collection of freeze frame data if the UDS DTC status bit 0 (TestedFailed) changes from 0 to 1. ▶ DEM_TRIGGER_TESTFAILED is only effectual for SWC events, BSW events will still be handled as DEM_TRIGGER_EVENT_MEMORY_STORAGE. ▶ DEM_TRIGGER_ON_FDC_THRESHOLD (vendor-specific extension): Triggers the collection of freeze frame data if ConfirmedDTC bit is not set and if one of the following conditions holds: <ul style="list-style-type: none"> ▶ unconfirmedDTC threshold reached (refer to DemDebounceCounterUnconfirmedThreshold) for the first time in an operation cycle and event has never failed ▶ failed threshold reached for the first time in an operation cycle Also the behavior of several functionalities is changed as described in DemExtendedDataCapture.
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_TRIGGER_EVENT_MEMORY_STORAGE
Range	DEM_TRIGGER_EVENT_MEMORY_STORAGE DEM_TRIGGER_TESTFAILED DEM_TRIGGER_ON_FDC_THRESHOLD
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemOperationCycleProcessing
Label	Operation Cycle Processing
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This configuration switch defines whether the operation cycle is triggered by START/END reports or collecting an external counter value, which results in respective state changes. Currently DEM_PROCESS_OPCYC_STATE is only supported and hence it is set as default value.</p> <p>Range:</p>



	<ul style="list-style-type: none"> ▶ DEM_PROCESS_OPCYC_STATE: Operation cycle is triggered by DEM_CYCLE_STATE_START. ▶ DEM_PROCESS_OPCYC_COUNTER: By collecting an external counter value.
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_PROCESS_OPCYC_STATE
Range	DEM_PROCESS_OPCYC_STATE DEM_PROCESS_OPCYC_COUNTER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemHeaderFileInclusion
Label	Header File Name
Description	Name of the header file(s) to be included by the Dem module containing the used C-callback declarations.
Multiplicity	0..n
Type	STRING
Default value	Module_Cbk.h
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemImmediateNvStorageLimit
Label	Immediate Nv Storage Limit
Description	<p>This parameter defines the maximum number of occurrences, a specific event memory entry is allowed, to be stored in NVRAM immediately (refer to DemImmediateNvStorage).</p> <p>The Dem module continues to update the Dem-internal event memory but will not trigger the immediate storage of an event memory entry into NVRAM when its occurrence counter has exceeded the value defined by DemImmediateNvStorageLimit.</p> <p>However, in special cases like ClearDTC or an internal memory reorganization, the immediate storage of an event memory entry into NVRAM is triggered also if its occurrence counter has exceeded the value defined by DemImmediateNvStorageLimit and even if DemImmediateNvStorage is disabled.</p>



	If the occurrence counter of an event (with DemImmediateNvStorage enabled) exceeds DemImmediateNvStorageLimit, the event memory entry and its event related data will be stored into NVRAM during the shutdown phase. DemImmediateNvStorageLimit should be enabled if DemImmediateNvStorage is enabled for any of the DTC's, otherwise a warning will be generated in DemGeneral
Multiplicity	0..1
Type	INTEGER
Default value	254
Range	<=254 >=1
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemNvStorageEmptyEventMemoryEntries
Label	Initialize EventMemory for immediate storage
Description	Switch to overwrite invalid or uninitialized NVRAM blocks with initialized empty event memory entries during shutdown. ▶ true: Initialized empty entries are written during Dem_Shutdown() ▶ false: Initialized empty entries are NOT written during Dem_Shutdown()
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemUserDefMemoryId
Label	User-defined memory identifier
Description	Defines a memory ID parameter to be used for selecting the Dem internal secondary event memory. If configured, and the requested DTCOrigin matches with the configured user-defined memory ID + 0x100, the DTCOrigin is mapped to the Dem internal secondary event memory. If not configured, all Dcm requests for the impacted services, issued for a DTCOrigin other than DEM_DTC_ORIGIN_PRIMARY_MEMORY, DEM_DTC_ORIGIN_MIRROR_MEMORY, DEM_DTC_ORIGIN_PERMANENT_MEMORY and DEM_DTC_ORIGIN_-



	OBD_RELEVANT_MEMORY will be considered invalid. Similarly, all SW-C/CDD requests for a DTCOrigin other than DEM_DTC_ORIGIN_PRIMARY_MEMORY, DEM_DTC_ORIGIN_SECONDARY_MEMORY, DEM_DTC_ORIGIN_MIRROR_MEMORY and DEM_DTC_ORIGIN_PERMANENT_MEMORY will be considered invalid.
Multiplicity	0..1
Type	INTEGER
Range	<=255 >=0
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemMaxNumberEventEntryMirror
Label	Mirror Event Memory Entries
Description	Maximum number of events which can be stored in the <i>mirror memory</i> .
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemMaxSizeFreezeFrameEntryMirror
Label	Maximum Mirror Freeze Frame Size
Description	<p>This value is only needed for calibration (without event combination).</p> <p>This size (in bytes) will be reserved for each mirror freeze frame entry in RAM/NVRAM. During calibration the total size of a mirror freeze frames must never be larger than this value. If the size gets larger than this configured value the behavior is undefined.</p> <p>The size of a freeze frame includes all common freeze frames, if configured.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DemCalibrationSupport: must be set to <code>true</code> to enable support for <i>Maximum Mirror Freeze Frame Size</i>. ▶ DemEventCombinationSupport: must be disabled (<code>DEM_EVCOMB_DISABLED</code>) to enable support for <i>Maximum Mirror Freeze Frame Size</i>.



	► DemMaxNumberEventEntryMirror: must be > 0 to enable support for <i>Maximum Mirror Freeze Frame Size</i> .
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<=65535 >=1
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemMaxNumberEventEntryPermanent
Label	Permanent Event Memory Entries (0, 4 -> 255)
Description	<p>Maximum number of events which can be stored in the <i>permanent memory</i>.</p> <p>The assignment of an event to this memory type is dynamic and used for emission-related events only.</p> <p>If zero is configured, the permanent memory is deactivated.</p> <p>Note: This configuration parameter is enabled only if OBD support is enabled.</p>
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DemMaxNumberEventEntryPrimary
Label	Primary Event Memory Entries
Description	Maximum number of events which can be stored in the <i>primary memory</i> .
Multiplicity	1..1
Type	INTEGER
Default value	30
Range	<=255 >=1
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU



Parameter Name	DemMaxSizeFreezeFrameEntryPrimary
Label	Maximum Primary Freeze Frame Size
Description	<p>This value is only needed for calibration (without event combination).</p> <p>This size (in bytes) will be reserved for each primary freeze frame entry in RAM/NVRAM. During calibration the total size of a primary freeze frames must never be larger than this value. If the size gets larger than this configured value the behavior is undefined.</p> <p>The size of a freeze frame includes all common freeze frames if configured.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DemCalibrationSupport: must be set to <code>true</code> to enable support for <i>Maximum Primary Freeze Frame Size</i>. ▶ DemEventCombinationSupport: must be disabled (<code>DEM_EVCOMB_DISABLED</code>) to enable support for <i>Maximum Primary Freeze Frame Size</i>. ▶ DemMaxNumberEventEntryPrimary: must be > 0 to enable support for <i>Maximum Primary Freeze Frame Size</i>.
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<p>≤ 65535</p> <p>≥ 1</p>
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemMaxNumberEventEntrySecondary
Label	Secondary Event Memory Entries
Description	Maximum number of events which can be stored in the <i>secondary memory</i> .
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemMaxSizeFreezeFrameEntrySecondary
-----------------------	--



Label	Maximum Secondary Freeze Frame Size
Description	<p>This value is only needed for calibration (without event combination).</p> <p>This size (in bytes) will be reserved for each secondary freeze frame entry in RAM/NVRAM. During calibration the total size of a secondary freeze frames must never be larger than this value. If the size gets larger than this configured value the behavior is undefined.</p> <p>The size of a freeze frame includes all common freeze frames if configured.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DemCalibrationSupport: must be set to <code>true</code> to enable support for <i>Maximum Secondary Freeze Frame Size</i>. ▶ DemEventCombinationSupport: must be disabled (<code>DEM_EVCOMB_DISABLED</code>) to enable support for <i>Maximum Secondary Freeze Frame Size</i>. ▶ DemMaxNumberEventEntrySecondary: must be > 0 to enable support for <i>Maximum Secondary Freeze Frame Size</i>.
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<p>≤ 65535</p> <p>≥ 1</p>
Configuration class	<p>PreCompile:</p> <p style="margin-left: 20px;">VariantPreCompile</p>
Origin	Elektrobit Automotive GmbH

Parameter Name	DemMaxNumberPrestoredFF
Label	Prestored Freeze Frame Entries
Description	<p><i>The functionality related to this parameter is not supported by the current implementation. Nevertheless, Freeze Frame Prestorage can be used for events individually.</i></p> <p>Defines the maximum number for prestored freeze frames.</p> <p>If set to 0, then freeze frame prestorage is not supported by the ECU.</p>
Multiplicity	1..1
Type	INTEGER
Default value	255
Range	≤ 255



	>=0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	DemOBDSupport
Label	OBD Support
Description	<p>This configuration switch defines whether OBD is supported or not.</p> <p>Note: In case DemOBDSupport is disabled, the DEM module will handle DTCExtDataRecordNumber of 0xFE in <code>Dem_GetSizeOfExtendedDataRecordSelection()</code> API with <code>DEM_NO SUCH ELEMENT</code> NRC.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	DemOBDIumprFunctionsEnabled
Label	IUMPR functions enabled
Description	<p>This configuration switch defines whether IUMPR related functions are available or not.</p> <ul style="list-style-type: none"> ▶ true: The functions <code>Dem_ReplIUMPRFaultDetect()</code>, <code>Dem_ReplIUMPRDenLock()</code> and <code>Dem_ReplIUMPRDenRelease()</code> are available. ▶ false: The functions <code>Dem_ReplIUMPRFaultDetect()</code>, <code>Dem_ReplIUMPRDenLock()</code> and <code>Dem_ReplIUMPRDenRelease()</code> are not available. <p>These functions are available only if DemOBDSupport is enabled.</p> <p>Note: The parameter option <code>true</code> will only provide dummy API implementation</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH
Parameter Name	DemCalibrationSupport
Label	Calibration Support



Description	<p>Switch to enable calibration support of specific configuration data.</p> <ul style="list-style-type: none"> ▶ true: Calibration supported. <p>The following parameters can be calibrated by considering the BSWMD data structure descriptions during calibration:</p> <ul style="list-style-type: none"> ▶ DTC (<i>DemDTCCClass</i>) functionality: <ul style="list-style-type: none"> ▶ DTC UDS Value (<i>DemUdsDTC</i>) ▶ DTC Functional Unit (<i>DemDTCFunctionalUnit</i>) ▶ DTC Severity (<i>DemDTCSeverity</i>) ▶ OBD DTC Value (<i>DemObdDTC</i>) ▶ Debounce Counter (<i>DemDebounceCounterBased</i>) functionality: <ul style="list-style-type: none"> ▶ Counter Decrement (<i>DemDebounceCounterDecrementStepSize</i>) ▶ Counter Increment (<i>DemDebounceCounterIncrementStepSize</i>) ▶ Counter Failed Threshold (<i>DemDebounceCounterFailedThreshold</i>) ▶ Counter Passed Threshold (<i>DemDebounceCounterPassedThreshold</i>) ▶ Counter Unconfirmed Threshold (<i>DemDebounceCounterUnconfirmedThreshold</i>) ▶ Jump-Down (<i>DemDebounceCounterJumpDown</i>) ▶ Jump-Down Value (<i>DemDebounceCounterJumpDownValue</i>) ▶ Jump-Up (<i>DemDebounceCounterJumpUp</i>) ▶ Jump-Up Value (<i>DemDebounceCounterJumpUpValue</i>) ▶ Time Debounce (<i>DemDebounceTimeBased</i>) functionality: <ul style="list-style-type: none"> ▶ Event Failed Timeout ((<i>DemDebounceTimeFailedThreshold</i>)) ▶ Event Passed Timeout ((<i>DemDebounceTimePassedThreshold</i>)) ▶ Indicator Attributes (<i>DemIndicatorAttribute</i>) functionality: <ul style="list-style-type: none"> ▶ Indicator Id (<i>DemIndicatorID</i>) ▶ Indicator Behaviour (<i>DemIndicatorBehaviour</i>) ▶ Indicator Failure Cycle Counter Threshold (<i>DemIndicatorFailureCycleCounterThreshold</i>) ▶ Indicator Healing Cycle Counter Threshold (<i>DemIndicatorHealingCycleCounterThreshold</i>) ▶ Indicator Failure Cycle (<i>DemIndicatorFailureCycleRef</i>) ▶ Indicator Healing Cycle (<i>DemIndicatorHealingCycleRef</i>)
--------------------	--



- ▶ **Event Class** (*DemEventClass*) functionality:
 - ▶ Event Failure Cycle Counter Threshold (*DemEventFailureCycleCounterThreshold*)
 - ▶ Event Failure Cycle (*DemEventFailureCycleRef*)
 - ▶ Minimum Ratio Event (*DemMinRatioEvent*)
 - ▶ Event OBD Readiness Group (*DemEventOBDReadinessGroup*)
 - ▶ OBD Aging Cycle Counter Threshold (*DemOBDAgingCycleCounterThreshold*)
 - ▶ Event Available (*DemEventAvailable*)
 - ▶ Aging Allowed (*DemAgingAllowed*)
 - ▶ Aging Cycle Counter Threshold (*DemAgingCycleCounterThreshold*)
 - ▶ Event Priority (*DemEventPriority*)
 - ▶ Operation Cycle Reference (*DemOperationCycleRef*)
- ▶ **Event General** (*DemEventParameter*) functionality:
 - ▶ Event Class (*DemEventClass*)
 - ▶ Enable Condition Group (*DemEnableConditionGroupRef*)
 - ▶ Freeze Frame Class (*DemFreezeFrameClassRef*)
 - ▶ Development Freeze Frame A Class (*DemDevFreezeFrameA-ClassRef*)
 - ▶ Development Freeze Frame B Class (*DemDevFreezeFrameB-ClassRef*)
- ▶ **Service API Parameters** (*DemGeneral*) functionality:
 - ▶ Force ReadinessGroup Comprehensive component Monitoring (*DemOBDForceCCMonitoringReady*)
 - ▶ Force ReadinessGroup Fuel System Monitoring (*DemOBDForceFuelSysMonitoringReady*)
 - ▶ Force ReadinessGroup Misfire Monitoring (*DemOBDForceMisfireMonitoringReady*)
 - ▶ Common part of Freeze Frame Class (*DemCommonFFDataClassRef*)
 - ▶ Common part of Development Freeze Frame A Class (*DemCommonFFDataDevAClassRef*)
 - ▶ Common Operation Cycle (*DemCommonOperationCycleSupport*)



- ▶ **Passive Table** (*DemAdvDisplacementPassiveTableLine*) functionality:
 - ▶ Reported Event Reference (*DemReportedEventRef*)
 - ▶ Passive Event Reference (*DemPassiveEventRef*)
 - ▶ **OBD Support** (*DemOBDSupport*) functionality:
 - ▶ OBD Compliancy (*DemOBDCompliance*)
 - ▶ OBD Engine Type (*DemOBDEngineType*)
 - ▶ **PID List** (*DemPidClass*) functionality:
 - ▶ PID Identifier (*DemPidIdentifier*)
 - ▶ **Freeze Frame Class List** (*DemFreezeFrameClass*) functionality:
 - ▶ Did Class (*DemDidClassRef*)
- If event combination support is enabled (see *DemEventCombinationSupport*) only following parameters can be calibrated:
- ▶ **OBD properties** (*Dem_OBDConfig*) functionality:
 - ▶ OBD DTC Value (*Dem_ObdDTC*)
 - ▶ **Indicator Attributes** (*DemIndicatorAttribute*) functionality;
 - ▶ Indicator Id (*DemIndicatorID*)
 - ▶ Indicator Behaviour (*DemIndicatorBehaviour*)
 - ▶ Indicator Failure Cycle Counter Threshold (*DemIndicatorFailureCycleCounterThreshold*)
 - ▶ Indicator Healing Cycle Counter Threshold (*DemIndicatorHealingCycleCounterThreshold*)
 - ▶ Indicator Failure Cycle (*DemIndicatorFailureCycleRef*)
 - ▶ Indicator Healing Cycle (*DemIndicatorHealingCycleRef*)
 - ▶ **Mapping of indicator group per event** (*DemIndicatorAttribute*).
 - ▶ false: Calibration not supported

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

Parameter Name	DemClearDTCOfDisabledEvents
Label	Clear DTC Of Disabled Events



Description	<p>Switch to enable clearing of DTCs for disabled events from event memory</p> <ul style="list-style-type: none"> ▶ <code>true</code>: all disabled events shall be cleared from the event memory including related actions (esp. status byte reset, callback triggers, etc.) if <code>ClearDTC</code> is called with <code>DTC = DEM_DTC_GROUP_ALL_DTCs</code>, regardless of whether the related <code>DemObdDTC/DemUdsDTC</code> exist, have the value 0 or they are suppressed ▶ <code>false</code>: disabled events shall not be cleared from event memory <p>This feature is only needed when:</p> <ul style="list-style-type: none"> ▶ calibration without event combination is enabled (<code>DemCalibrationSupport</code> is set to <code>true</code> and <code>DemEventCombinationSupport</code> is set to <code>DEM_EVCOMB_DISABLED</code>) OR ▶ dynamic availability is enabled (<code>DemAvailabilitySupport</code> is set to <code>DEM_EVENT_AVAILABILITY</code>) <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ <code>DemCalibrationSupport</code>: must be set to <code>true</code> to enable support for <i>Clear DTC Of Disabled Events</i>. ▶ <code>DemEventCombinationSupport</code>: must be disabled (<code>DEM_EVCOMB_DISABLED</code>) to enable support for <i>Clear DTC Of Disabled Events</i>. ▶ <code>DemAvailabilitySupport</code>: must be set to dynamic availability (<code>DEM_EVENT_AVAILABILITY</code>) to enable support for <i>Clear DTC Of Disabled Events</i>.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemCommonOperationCycleSupport
Label	Common Operation Cycle
Description	<p>This configuration parameter defines whether a common operation cycle is used.</p> <p>If this parameter is enabled, the Dem uses the operation cycle configured in <code>DemOperationCycleRef</code> as common operation cycle for these specific operation cycle references:</p> <ul style="list-style-type: none"> ▶ <code>DemAgingCycleRef</code>



	<ul style="list-style-type: none"> ▶ DemEventFailureCycleRef ▶ DemIndicatorFailureCycleRef ▶ DemIndicatorHealingCycleRef <p>Therefore those specific operation cycle references are disabled and not editable if the common operation cycle is enabled.</p> <p>This value is only needed for calibration (without event combination).</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DemCalibrationSupport: must be set to <code>true</code> to enable support for <i>Common Operation Cycle</i>. ▶ DemEventCombinationSupport: must be disabled (<code>DEM_EVCOMB_DISABLED</code>) to enable support for <i>Common Operation Cycle</i>.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemAgingAllowedSeperateFlag
Label	Aging Allowed Separate Flag
Description	<p>This parameter enables a convenient way for calibrating the value of DemAgingAllowed.</p> <p>If this parameter is enabled the flag AllowAging is used to calibrate the value of DemAgingAllowed. Otherwise the value of DemAgingCycleCounterThreshold is used to disable aging, the threshold must be calibrated to 0 to disable aging.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DemCalibrationSupport: must be set to <code>true</code> to enable support for <i>Aging Allowed Separate Flag</i>. ▶ DemEventCombinationSupport: must be disabled (<code>DEM_EVCOMB_DISABLED</code>) to enable support for <i>Aging Allowed Separate Flag</i>.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile



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

Parameter Name	DemMaxNumberEventClasses
Label	Maximum number of Event Classes
Description	<p>The vendor-specific parameter DemMaxNumberEventClasses specifies how many instances of the class DemEventClass that should be allocated at least.</p> <p>If calibration (without event combination) is enabled the Dem will generate parts of DemEventClass separately and not as part of the DemEventParameter description.</p> <p>Each set of configuration variants in DemEventClass will only have one entry in the data structure Dem_EventClassDesc[]. If less different sets of DemEventClass are configured than DemMaxNumberEventClasses, the Dem will generate additional empty entries in Dem_EventClassDesc[]. These empty entries can be assigned during calibration.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DemCalibrationSupport: must be set to <code>true</code> to enable support for <i>Maximum number of Event Classes</i>. ▶ DemEventCombinationSupport: must be disabled (<code>DEM_EVCOMB_DISABLED</code>) to enable support for <i>Maximum number of Event Classes</i>.
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<p><=2048</p> <p>>=1</p>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemMaxNumberIndicatorClasses
Label	Maximum number of Indicator Classes
Description	<p>The vendor-specific parameter DemMaxNumberIndicatorClasses specifies how many instances of the class DemIndicatorClass should at least be allocated.</p> <p>If calibration is enabled, the Dem will generate one indicator class for every distinct set of indicator attributes of the configured events.</p>



	<p>If the number of indicator classes is less than the value of this parameter, then this value specifies the number of indicator classes. The additional indicator classes are generated with dummy values.</p> <p>If this parameter is configured and has a value greater than 0, then at least one indicator has to be configured in the DemIndicator container and linked to an event.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DemCalibrationSupport: must be set to <code>true</code> to enable support for <i>Maximum number of Indicator Classes</i>.
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemOBDSupportKind	
Label	OBD Support kind	
Description	<p>This configuration switch defines OBD support and the kind of OBD ECU.</p> <p><i>Note:</i> The name of the configuration parameter is not AUTOSAR-compatible.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_OBD_DEP_SEC_ECU: OBD dependent / secondary ECU. (<i>currently not supported</i>) ▶ DEM_OBD_MASTER_ECU: OBD master ECU. ▶ DEM_OBD_PRIMARY_ECU: OBD primary ECU. (<i>currently not supported</i>) 	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	DEM_OBD_MASTER_ECU	
Range	<hr/> DEM_OBD_DEP_SEC_ECU <hr/> DEM_OBD_MASTER_ECU <hr/> DEM_OBD_PRIMARY_ECU	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	



Parameter Name	DemOccurrenceCounterProcessing
Label	Occurrence Counter Processing
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This configuration switch defines the consideration of the fault confirmation process for the occurrence counter. For OBD and mixed systems (OBD/non OBD, refer to DemOBDSupport) the fault confirmation process must not be considered.</p> <p>The current implementation of the occurrence counter processing behaves as if the <code>DEM_PROCESS_OCCCTR_TF</code> would have been selected.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ <code>DEM_PROCESS_OCCCTR_CDT</code>: the occurrence counter is triggered after the fault confirmation was successful. (<i>currently not supported</i>) ▶ <code>DEM_PROCESS_OCCCTR_TF</code>: the occurrence counter is triggered by the TestFailed bit only, but the fault confirmation is not considered.
Multiplicity	1..1
Type	ENUMERATION
Default value	<code>DEM_PROCESS_OCCCTR_TF</code>
Range	<code>DEM_PROCESS_OCCCTR_CDT</code> <code>DEM_PROCESS_OCCCTR_TF</code>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemOperationCycleStatusStorage
Label	Operation Cycle Status Storage
Description	<p>Defines if the operation cycle state is available over the power cycle (stored non-volatile) or not.</p> <p>The NVRAM block <code>DEM_NVM_BLOCK_ID_DEFAULT</code> needs to be configured (refer to <code>DemNvRamBlockId</code>) to store the operation cycle state non-volatile.</p> <ul style="list-style-type: none"> ▶ <code>true</code>: the operation cycle state is stored non-volatile. ▶ <code>false</code>: the operation cycle state is only stored volatile <p>Integration notes:</p>



	<ul style="list-style-type: none"> ▶ In case of activated feature, event-specific fault detection counters (FDC) will be reset to 0 (zero) when Dem_PreInit has been called. ▶ In case of activated feature, ALL operation cycle states will be stored non-volatile. Operations cycles which should not be kept started after a power cycle need to be stopped before Dem_Shutdown(). ▶ In case of deactivated feature, ONLY operation cycles with DemOperationCycleAutomaticEnd enabled will be stopped automatically during Dem_Shutdown().
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemPTOSupport
Label	PTO Support
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This configuration switch defines whether PTO support (and therefore PID \$1E support) is enabled or not.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemStatusBitHandlingTestFailedSinceLastClear
Label	Handling of TestFailedSinceLastClear Status-Bit
Description	<p>This configuration switch defines whether the aging and displacement mechanism shall be applied to the "TestFailedSinceLastClear" status bits</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_STATUS_BIT_AGING_AND_DISPLACEMENT: The "TestFailedSinceLastClear" status bits are reset to 0, if aging or displacement applies (like done for the "ConfirmedDTC" status bits).



	► DEM_STATUS_BIT_NORMAL: Aging and displacement has no impact on the "TestFailedSinceLastClear" status bits
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_STATUS_BIT_NORMAL
Range	DEM_STATUS_BIT_AGING_AND_DISPLACEMENT DEM_STATUS_BIT_NORMAL
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemResetPendingBitOnOverflow
Label	Reset Pending Bit On Overflow
Description	<p>This configuration switch defines, whether the pending bit is reset or not while an event memory entry will be displaced or aged (the bit will not be set without an event memory entry).</p> <p>Thus, it affects the way in which the PDTC bit is read with the help of UDS service <i>ReadDTCInformation (0x19)</i>.</p> <p>Note: This parameter influences the reporting of DTCs as per SWS_Dem_00410 (ASR R.4.2) and this requirement is completely satisfied only when this parameter is enabled.</p> <p>Range:</p> <ul style="list-style-type: none"> ► true: The UDS status bit PDTC is not set without an event memory entry. The UDS status bit PDTC is reset if the event memory entry is displaced or aged. ► false: The PDTC UDS status bit will be set independent of an event memory entry and remains set for displaced or aged event memory entries. Due to this behavior the PDTC bit cannot be used for filtering only events\DTCS that have an entry in the event memory. <p>Note: Independent of this parameter, the bit will always be reset during ClearDTC.</p> <p>Note: The PDTC behavior is decoupled from the R4.1 configuration parameter DemResetConfirmedBitOnOverflow by this separated parameter.</p>
Multiplicity	1..1
Type	BOOLEAN



Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemUpdateOccOrderOnEventEntryUpdate	
Label	Update OccOrder On Event Entry Change	
Description	<p>This configuration switch defines the behaviour for updating the occurrence order (OccOrder).</p> <ul style="list-style-type: none"> ▶ true: update OccOrder with every update of the related event memory entry ▶ false: update OccOrder only once when event memory entry is created 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemResetConfirmedBitOnOverflow	
Label	Reset Confirmed Bit On Overflow	
Description	<p>This configuration switch defines whether the confirmed bit is reset or not while an event memory entry will be displaced (the bit will not be set without an event memory entry).</p> <p>Thus, it affects the way in which the CDTC bit is read with the help of UDS service <i>ReadDTCInformation (0x19)</i>.</p> <p>Note: This parameter influences the reporting of DTCs as per SWS_Dem_00410 (ASR R.4.2) and this requirement is completely satisfied only when this parameter is enabled.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ true: The UDS status bit CDTC is not set without an event memory entry. The UDS status bit CDTC is reset if the event memory entry is displaced. ▶ false: The CDTC UDS status bit will be set independent of an event memory entry and remains set for displaced event memory entries. The following functionalities result: 	



	<ul style="list-style-type: none"> ▶ The CDTC bit cannot be used for filtering only events\DTCS that have an entry in the event memory. All events will be included in the filtering operation. ▶ If at least one event has fault confirmation enabled, meaning Event Failure Cycle Counter Threshold (DemEventFailureCycleCounterThreshold) must be configured, it allows for a behavior where the <i>fault confirmation</i> is available <i>without an event memory entry</i>. <p>Note: Independent of this parameter, the bit will always be reset during aging and ClearDTC.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Reset Pending Bit On Overflow: must be disabled, to allow disabling of "Reset Confirmed Bit On Overflow".
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemStatusBitStorageTestFailed	
Label	Status Bit Storage Test Failed	
Description	Activate/Deactivate the permanent storage of the "TestFailed" status bits. <ul style="list-style-type: none"> ▶ true: storage activated ▶ false: storage deactivated When DemStatusBitStorageTestFailedPerEvent is enabled it will overwrite this parameter for individual events.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemStorageConditionSupport	
Label	Storage Condition Support	



Description	<i>The functionality related to this parameter is not supported by the current implementation.</i>
	This configuration switch defines whether support for storage conditions is enabled or not. <ul style="list-style-type: none"> ▶ true: support for storage conditions is enabled ▶ false: support for storage conditions is disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DemImmediateStorageInternalEvents
Label	Immediate Storage For Internal Events
Description	This configuration switch defines if internal events are stored immediately. <ul style="list-style-type: none"> ▶ true: internal events are stored immediately ▶ false: internal events are not stored immediately <p>Warning: Take into consideration that only configured internal events will be affected by this configuration parameter. For the events that are re-calibrated to be internal, the old Immediate Storage configuration will be taken into consideration.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemIntermediateNvStorageOfDTCStatus
Label	Intermediate DTC status storage
Description	Enables the intermediate non-volatile storage of the DTC status. <ul style="list-style-type: none"> ▶ true: Intermediate non-volatile storage of the DTC status is enabled ▶ false: Intermediate non-volatile storage of the DTC status is disabled



	<p>If enabled, the non-volatile storage of the DTC status is triggered at intermediate points in time, whenever the immediate storage of the event related data is triggered.</p> <p>For combined DTCs, the event status of all events contributing to the combined DTC status is stored.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ <code>DemImmediateNvStorage</code>: must be set to <code>true</code> for a DTC to enable support for the intermediate non-volatile storage of the DTC status. ▶ <code>DemImmediateStorageInternalEvents</code>: when set to <code>true</code>, it activates the intermediate non-volatile storage of the DTC status for internal events. ▶ <code>DemStatusBitStorageTestFailed</code>: when set to <code>false</code>, it deactivates the non-volatile storage of the TestFailed (TF) status bit.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	<code>PreCompile:</code> <code>VariantPreCompile</code>
Origin	Elektrobit Automotive GmbH

Parameter Name	DemTaskTime
Label	Task Time (s)
Description	<p>Allows to configure the time for the periodic cyclic task (in seconds).</p> <p><i>Note:</i> This configuration value shall be equal to the value in the Basic Software Scheduler configuration of the Rte module.</p> <p>The AUTOSAR configuration standard is to use SI units, so this parameter is defined as float value in seconds.</p> <p><i>min:</i> A negative value is not allowed.</p> <p><i>max:</i> After the event status was reported, processing shall be completed within 100ms in order to have the fault entry status information updated as soon as possible (e.g. for PID \$01).</p> <p><i>upperMultiplicity:</i> Exactly one <code>TaskTime</code> must be specified per configuration.</p> <p><i>lowerMultiplicity:</i> Exactly one <code>TaskTime</code> must be specified per configuration.</p>
Multiplicity	1..1
Type	FLOAT



Default value	0.02
Range	<=0.1 >=0.001
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU V1.0.0

Parameter Name	DemTriggerDcmReports
Label	Trigger Dcm Reports
Description	Activate/Deactivate the notification to the Diagnostic Communication Manager for ROE processing. ▶ true: Dcm ROE notification activated ▶ false: Dcm ROE notification deactivated
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DemTriggerDltReports
Label	Trigger Dlt Reports
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> Activate/Deactivate the notification to the Diagnostic Log and Trace. ▶ true: Dlt notification activated ▶ false: Dlt notification deactivated
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DemTriggerFimReports
Label	Trigger Fim Reports



Description	Activate/Deactivate the notification to the Function Inhibition Manager. It is used by FiM only if FiMEventUpdateTriggeredByDem is enabled in FiM.	
	<ul style="list-style-type: none"> ▶ true: FiM notification activated ▶ false: FiM notification deactivated 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DemSupportFiMFDCThreshold	
Label	Support FiM FDC Threshold	
Description	<p>Activate/Deactivate the notification to the Function Inhibition Manager on unconfirmed FDC threshold. It is used by FiM only if is enabled in FiM.</p> <ul style="list-style-type: none"> ▶ true: FiM notification activated and interfaces are available ▶ false: FiM notification deactivated and interfaces are absent <p>Note: If FiM notification is enabled, then DemExtendedDataCapture configuration parameter has to be set to "DEM_TRIGGER_ON_FDC_THRESHOLD".</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	VCC	

Parameter Name	DemTriggerMonitorInitBeforeClearOk	
Label	Trigger Monitor Init Before Clear Ok	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Defines if the monitor re-initialization has to be triggered before or after the Dem module returns DEM_CLEAR_OK.</p> <ul style="list-style-type: none"> ▶ true: trigger re-initialization before DEM_CLEAR_OK ▶ false: trigger re-initialization after DEM_CLEAR_OK 	
Multiplicity	1..1	



Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemCallbackInitMForEReenabledSupport	
Label	Init Monitor for Event Reenabled Support	
Description	<p>This configuration parameter specifies whether support of <code>InitMonitorForEvent</code> callback function with <code>InitMonitorReason DEM_INIT_MONITOR_REENABLED</code> is enabled or not.</p> <ul style="list-style-type: none"> ▶ true: enable support of <code>InitMonitorForEvent</code> callback function with <code>InitMonitorReason DEM_INIT_MONITOR_REENABLED</code> ▶ false: disable support of <code>InitMonitorForEvent</code> callback function with <code>InitMonitorReason DEM_INIT_MONITOR_REENABLED</code> 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemMaxHandledInitMonitorReenabledPerScheduling	
Label	Maximum number of <code>InitMonitorForEvent</code> callbacks triggered per scheduling	
Description	<p>This parameter defines the maximum number of <code>InitMonitorForEvent</code> callbacks to be triggered per invocation of <code>Dem_MainFunction</code>, when DTC setting is re-enabled. This configuration item has effect only if the value is less than the number of configured <code>DemCallbackInitMForE/DemCallbackInitMForEFnc</code> callbacks.</p> <p>If a large value is configured for <code>DemMaxHandledInitMonitorReenabledPerScheduling</code>, the run time of <code>Dem_MainFunction</code> will correspondingly increase when DTC setting is re-enabled.</p> <p>On the other hand, more time will be required until all configured <code>InitMonitorForEvent</code> callbacks are triggered, if a small value is configured for <code>DemMaxHandledInitMonitorReenabledPerScheduling</code>.</p> <p>Dependency on parameter(s):</p>	



	<ul style="list-style-type: none"> ▶ DemCallbackInitMForEReenabledSupport: must be set to <code>true</code> to enable support of <code>InitMonitorForEvent</code> callback function with <code>InitMonitorReason DEM_INIT_MONITOR_REENABLED</code>.
Multiplicity	1..1
Type	INTEGER
Default value	50
Range	<p><=1023</p> <p>>=1</p>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemTypeOfDTCSupported
Label	Supported DTC Type
Description	<p>This parameter defines the format returned by <code>Dem_GetTranslationType</code> and does not relate to/influence the supported Dem functionality.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ <code>DEM_DTC_TRANSLATION_ISO11992_4</code>: ISO11992-4 DTC format. ▶ <code>DEM_DTC_TRANSLATION_ISO14229_1</code>: ISO14229-1 DTC format (3 byte format). ▶ <code>DEM_DTC_TRANSLATION_ISO15031_6</code>: IISO15031-6 DTC format (2 byte format). ▶ <code>DEM_DTC_TRANSLATION_SAEJ1939_73</code>: SAEJ1939-73 DTC format. ▶ <code>DEM_DTC_TRANSLATION_J2012DA_FORMAT_04</code>: SAE_J2012-DA_DTC-Format_00 (3 byte format).
Multiplicity	1..1
Type	ENUMERATION
Default value	<code>DEM_DTC_TRANSLATION_ISO14229_1</code>
Range	<p><code>DEM_DTC_TRANSLATION_ISO11992_4</code></p> <p><code>DEM_DTC_TRANSLATION_ISO14229_1</code></p> <p><code>DEM_DTC_TRANSLATION_ISO15031_6</code></p> <p><code>DEM_DTC_TRANSLATION_SAEJ1939_73</code></p> <p><code>DEM_DTC_TRANSLATION_J2012DA_FORMAT_04</code></p>
Configuration class	VariantPreCompile: VariantPreCompile



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

Parameter Name	DemTypeOfFreezeFrameRecordNumeration	
Label	Type Of Freeze Frame Record Numeration	
Description	<p>This parameter defines the type of assigning freeze frame record numbers for event-specific freeze frame records.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_FF_RECNUM_CALCULATED: freeze frame records will be numbered consecutively starting with 1 in their chronological order. ▶ DEM_FF_RECNUM_CONFIGURED: freeze frame records will be numbered based on the given configuration in their chronological order. <p>Constraint: If DEM_FF_RECNUM_CONFIGURED is selected, DemFreezeFrameRecNumClass list shall not be empty.</p>	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	DEM_FF_RECNUM_CALCULATED	
Range	DEM_FF_RECNUM_CALCULATED DEM_FF_RECNUM_CONFIGURED	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemCommonFFDataClassRef	
Label	Common part of Freeze Frame Class	
Description	<p>Parameter used for pointing out the common part for the ordinary AUTOSAR freeze frame.</p> <ul style="list-style-type: none"> ▶ If this parameter is enabled, all regular freeze frames configured via DemFreezeFrameClassRef will always contain the data given by this reference as common part in front of them, even if no individual freeze frame is configured by DemFreezeFrameClassRef. In this case, the regular freeze frame cannot be deactivated completely for any event. ▶ If this parameter is disabled, there will be no common data attached in front of the regular freeze frames. <p>This mechanism can be used for saving configuration data in ROM and to lower the effort to configure/calibrate the freeze frames.</p>	



Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemDevFreezeFrameSupport
Label	Development Freeze Frame Support
Description	<p>This parameter configures whether the Development Freeze Frames are supported or not:</p> <ul style="list-style-type: none"> ▶ TRUE: Development Freeze Frame A (refer to <i>DemDevFreezeFrameAClassRef</i>) and Development Freeze Frame B (<i>DemDevFreezeFrameBClassRef</i>) are supported. ▶ FALSE: Development Freeze Frames are not supported. <p>Development freeze frames are:</p> <ul style="list-style-type: none"> ▶ not supported if event combination is enabled (see <i>DemEventCombinationSupport</i>) ▶ not immediately stored (see <i>DemIMmediateNvStorage</i>) ▶ not prestored (see <i>DemFFPrestorageSupported</i>) i.e., it is not supported by the API <i>Dem_PrestoreFreezeFrame</i> ▶ accessible only to the Dcm e.g., it cannot be read by the API <i>Dem_GetEventFreezeFrameData</i>
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemCommonFFDataDevAClassRef
Label	Common part of Development Freeze Frame A Class
Description	<p>This reference defines the link to the common part for the Development Freeze Frame A Class.</p> <ul style="list-style-type: none"> ▶ If this parameter is enabled, all development freeze frames of type A configured via <i>DemDevFreezeFrameAClassRef</i> will always contain the data given by this reference as common part in front of them, even if no individual



	<p>freeze frame is configured by DemDevFreezeFrameAClassRef. In this case, the development freeze frame A cannot be deactivated completely for any event.</p> <ul style="list-style-type: none"> ▶ If this parameter is disabled, there will be no common data attached in front of the development freeze frames of type A. <p>This mechanism can be used for saving configuration data in ROM and to lower the effort to configure/calibrate the freeze frames.</p>
Multiplicity	0..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemMaxNumberDevFreezeFrameAEntry
Label	Development Freeze Frame A Entries
Description	<p>Maximum number of Development Freeze Frame A events.</p> <p>The size of Development Freeze Frame A has to fit to the size of Primary memory.</p>
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemMaxSizeDevFreezeFrameAEntry
Label	Maximum size of Development Freeze Frame A Class
Description	<p>This value is only needed for calibration (without event combination).</p> <p>This size (in bytes) will be reserved for each development freeze frame A entry in RAM/NVRAM.</p> <p>During calibration the total size of the development freeze frame A including the common freeze frame must never be larger than this value. If the size gets larger than this configured value, the behavior is undefined.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DemCalibrationSupport: must be set to <code>true</code> to enable support for <i>Maximum size of Development Freeze Frame A Class</i>.



	<ul style="list-style-type: none"> ▶ DemEventCombinationSupport: must be disabled (<code>DEM_EVCOMB_DISABLED</code>) to enable support for <i>Maximum size of Development Freeze Frame A Class</i>. ▶ DemMaxNumberDevFreezeFrameAEntry: must be > 0 to enable support for <i>Maximum size of Development Freeze Frame A Class</i>.
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<p><=65545</p> <hr/> <p>>=1</p>
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemMaxNumberDevFreezeFrameBEntry
Label	Development Freeze Frame B Entries
Description	<p>Maximum number of Development Freeze Frame B events.</p> <p>The size of Development Freeze Frame B has to fit to the size of Primary memory.</p>
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemMaxSizeDevFreezeFrameBEntry
Label	Maximum size of Development Freeze Frame B Class
Description	<p>This value is only needed for calibration (without event combination).</p> <p>This size (in bytes) will be reserved for each development freeze frame B entry in RAM/NVRAM.</p> <p>During calibration the total size of the development freeze frame B must never be larger than this value. If the size gets larger than this configured value, the behavior is undefined.</p> <p>Dependency on parameter(s):</p>



	<ul style="list-style-type: none"> ▶ DemCalibrationSupport: must be set to <code>true</code> to enable support for <i>Maximum size of Development Freeze Frame B Class</i>. ▶ DemEventCombinationSupport: must be disabled (<code>DEM_EVCOMB_DISABLED</code>) to enable support for <i>Maximum size of Development Freeze Frame B Class</i>. ▶ DemMaxNumberDevFreezeFrameBEntry: must be > 0 to enable support for <i>Maximum size of Development Freeze Frame B Class</i>.
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<=65545 >=1
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemStoreInternalEvents	
Label	Dem Store Internal Events	
Description	<p>This configuration parameter defines whether storage of internal events (DemDTCClassRef is disabled), including freeze frames and extended data, is performed or only the event status is updated:</p> <ul style="list-style-type: none"> ▶ TRUE: Event-related data of internal events will be stored. (With calibration support, an event can be re-calibrated to an internal event by calibrating the DTC value to 0.) ▶ FALSE: Event-related data of internal events will not be stored. <p>Note: UDS Status-Byte will be updated regardless whether this parameter is enabled or disabled.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemVersionInfoApi	
Label	Enable Version Info API	
Description	Activate/Deactivate the version information API.	



	<ul style="list-style-type: none"> ▶ <code>true</code>: version information activated ▶ <code>false</code>: version information deactivated
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU
Parameter Name	DemMILIndicatorRef
Label	MIL Indicator
Description	<p>This parameter defines the indicator representing the MIL.</p> <p>This parameter is mandatory for ECUs supporting OBD (refer to <code>DemOBDSupport</code>).</p>
Multiplicity	0..1
Type	REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU
Parameter Name	DemMaxNumEventsPerReadinessGroup
Label	Events per Readiness Group
Description	<p>Maximum number of events per OBD Readiness Group.</p> <p>This parameter controls the maximum number of events which can be assigned to an OBD Readiness Group.</p> <p>It is enabled only if <code>DemOBDSupport</code> and <code>DemCalibrationSupport</code> are both enabled and set to <code>true</code> and <code>DemEventCombinationSupport</code> is disabled (set to <code>DEM_EVCOMB_DISABLED</code>)</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ <code>DemOBDSupport</code>: must be set to <code>true</code> to enable support for <i>Events per Readiness Group</i>. ▶ <code>DemCalibrationSupport</code>: must be set to <code>true</code> to enable support for <i>Events per Readiness Group</i>. ▶ <code>DemEventCombinationSupport</code>: must be disabled (<code>DEM_EVCOMB_DISABLED</code>) to enable support for <i>Events per Readiness Group</i>.
Multiplicity	1..1



Type	INTEGER	
Default value	1	
Range	<=40 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemJ1939Support	
Label	J1939 Support	
Description	<p>This configuration switch defines whether J1939 is supported or not.</p> <p>If OBD is disabled, the following DMs are unavailable:</p> <ul style="list-style-type: none"> ▶ DM5: Diagnostic Readiness 1 ▶ DM6: Emission-Related Pending Diagnostic Trouble Codes ▶ DM12: Emission-Related MIL-On Diagnostic Trouble Codes ▶ DM23: Emission-Related Previously MIL-On Diagnostic Trouble Codes ▶ DM28: Emission-Related Permanent Diagnostic Trouble Codes 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC R4.2.2	

Parameter Name	DemAmberWarningLampIndicatorRef	
Label	Amber Warning Indicator	
Description	<p>This parameter defines the indicator representing the Amber Warning.</p> <p>This parameter is optional for ECUs supporting J1939 (refer to DemJ1939Support).</p> <p>If this parameter is enabled and calibration support is OFF, it is expected that at least one event references it. If this parameter is enabled and calibration support is ON, it is expected that at least one indicator attribute is defined so that later on the reference can be filled during calibration.</p>	
Multiplicity	0..1	
Type	REFERENCE	



Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC R4.2.2	

Parameter Name	DemProtectLampIndicatorRef	
Label	Protect Indicator	
Description	<p>This parameter defines the indicator representing the Protect.</p> <p>This parameter is optional for ECUs supporting J1939 (refer to DemJ1939Support).</p> <p>If this parameter is enabled and calibration support is OFF, it is expected that at least one event references it. If this parameter is enabled and calibration support is ON, it is expected that at least one indicator attribute is defined so that later on the reference can be filled during calibration.</p>	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC R4.2.2	

Parameter Name	DemRedStopLampIndicatorRef	
Label	Red Stop Indicator	
Description	<p>This parameter defines the indicator representing the Red Stop.</p> <p>This parameter is optional for ECUs supporting J1939 (refer to DemJ1939Support).</p> <p>If this parameter is enabled and calibration support is OFF, it is expected that at least one event references it. If this parameter is enabled and calibration support is ON, it is expected that at least one indicator attribute is defined so that later on the reference can be filled during calibration.</p>	
Multiplicity	0..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC R4.2.2	

Parameter Name	DemOBDForceMisfireMonitoringReady	
Label	Force ReadinessGroup Misfire Monitoring	
Description	When this parameter is set to true PID\$01 and PID\$41 byte B bit 4 (Complete bit for ReadinessGroup Misfire Monitoring) will be forced to complete.	



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

Parameter Name	DemOBDForceFuelSysMonitoringReady	
Label	Force ReadinessGroup Fuel System Monitoring	
Description	When this parameter is set to true PID\$01 and PID\$41 byte B bit 5 (Complete bit for ReadinessGroup Fuel System Monitoring) will be forced to complete.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemOBDForceCCMonitoringReady	
Label	Force ReadinessGroup Comprehensive component Monitoring	
Description	When this parameter is set to true PID\$01 and PID\$41 byte B bit 6 (Complete bit for ReadinessGroup Comprehensive component Monitoring) will be forced to complete.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemNvDataConfigSignatureUsed	
Label	Nv Data configuration signature	
Description	<p>This configuration parameter enables a module-internal mechanism for the validation of the non-volatile data used by the DEM module.</p> <p>This mechanism calculates a signature of the current DEM configuration to store it non volatile.</p> <p>Once the calculated signature differs from the stored one, all non-volatile data used by the DEM are deleted-initialized during <code>Dem_Init()</code>.</p>	



	<ul style="list-style-type: none"> ▶ true: Validation of non-volatile data is enabled ▶ false: Validation of non-volatile data is disabled <p>Note: If this parameter is changed from disabled to enabled, the NvM data is corrupted. This implies a signature mismatch and the Dem performs a reset of the NvM data.</p> <p>If this parameter is changed from enabled to disabled, then NvM data is corrupted as well. But this could not be identified by the Dem module. The NvM data have to be adjusted manually as described below.</p> <p>Note: Devices without a shutdown phase shall NOT use this feature. It is not ensured that the actual valid signature is successfully stored in the default NvM block.</p> <p>Note: Be aware that the content of the permanent memory (OBD-relevant events) is also deleted in case of a signature mismatch. Evaluate the relevant impact to all legislative requirements!</p> <p>Note: If this feature can NOT be used for various reasons, you have to process a manual adjustment of the non-volatile data as follows:</p> <ul style="list-style-type: none"> ▶ Lock event memories ▶ Clear event memories ▶ Update Dem Configuration ▶ Update related block size configurations in NVRAM, if necessary ▶ Update Block IDs of all Dem NVRAM blocks (by using the "Static Block ID Check", see NvMStaticBlockIDCheck)
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemNvDataConfigSignatureCalcSet
Label	Nv Data configuration signature for all parameters
Description	<p>This configuration parameter defines how the signature used in DemNvDataConfigSignatureUsed is calculated.</p> <ul style="list-style-type: none"> ▶ DEM_NVM_SIGNATURE_MIN: Calculation over all configuration parameters which affects the size and the structure of the NvM data



- ▶ `DEM_NVM_SIGNATURE_ALL`: Calculation over all existing DEM configuration parameters

Following configuration parameters affect the size and the structure of the NvM data:

- ▶ `DemFreezeFrameCapture`
- ▶ `DemExtendedDataCapture`
- ▶ `DemNvRamBlockIdRef`
- ▶ `DemResetConfirmedBitOnOverflow`
- ▶ `DemGetDTCByOccurrenceTimeSupport`
- ▶ `DemMaxNumberEventEntryPrimary`
- ▶ `DemMaxNumberEventEntrySecondary`
- ▶ `DemMaxNumberEventEntryMirror`
- ▶ `DemMaxNumberEventEntryPermanent`
- ▶ `DemClearDTCBehavior`
- ▶ `DemImmediateStorageInternalEvents`
- ▶ `DemOperationCycleStatusStorage`
- ▶ `DemIndicator`
- ▶ `DemMaxNumberDevFreezeFrameAEntry`
- ▶ `DemMaxNumberDevFreezeFrameBEntry`
- ▶ `DemEventCombinationSupport`
- ▶ `DemOBDSupport`
- ▶ `DemOperationCycleType`
- ▶ `DemOBDCentralizedPID21Handling`
- ▶ `DemOBDCentralizedPID31Handling`
- ▶ `DemPidClass`
- ▶ `DemEventParameter`
- ▶ `DemEventId`
- ▶ `DemExtendedDataClassRef`
- ▶ `DemExtendedDataRecordClassRef`
- ▶ `DemDataElementClassRef`
- ▶ `DemDataElementReadFnc`
- ▶ `DemDataElementUsePort`



	<ul style="list-style-type: none"> ▶ DemInternalDataElement ▶ DemFreezeFrameClassRef ▶ DemDidClassRef ▶ DemDidIdentifier ▶ DemDidDataElementClassRef ▶ DemDataElementDataSize ▶ DemDevFreezeFrameAClassRef ▶ DemDevFreezeFrameBClassRef ▶ DemDTCClassRef ▶ DemImmediateNvStorage ▶ DemEventFailureCycleCounterThreshold ▶ DemIndicatorAttribute ▶ DemEventAvailable ▶ DemEventDestination ▶ DemEvtCmbCommonParamMaster ▶ DemDebounceCounterStorage <p>In case of <code>DEM_NVM_SIGNATURE_MIN</code>, any change to these parameters leads to an update of the configuration signature and consequently, to the reset of the NvM data.</p> <p>In case of <code>DEM_NVM_SIGNATURE_ALL</code>, each change of the configuration would lead to a reset of the NvM data, also a change of SwMajorVersion, SwMinorVersion or SwPatchVersion which is caused by a module update.</p>
Multiplicity	1..1
Type	ENUMERATION
Default value	<code>DEM_NVM_SIGNATURE_MIN</code>
Range	<code>DEM_NVM_SIGNATURE_MIN</code> <code>DEM_NVM_SIGNATURE_ALL</code>
Configuration class	PreCompile: <input style="width: 100px;" type="text"/> VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemCallbackMemStackMainFuncTrigger
Label	'Main function trigger' Function Name



Description	<p>User-defined name of the Memory stack main function triggering callout function. If this parameter is enabled, the Dem indirectly triggers the Memory stack by calling this callout function. This is required as <code>Dem_Init()</code> and <code>Dem_Shutdown()</code> are synchronous.</p> <p>The function must have a prototype compatible to <code>extern void Appl_TriggerMemoryStack(NvM_BlockIdType BlockId);</code>, which is declared in <code>Dem_Cfg.h</code>.</p> <p>An integration code example can be found in the template file <code>Dem_Callouts.c</code> together with a description of the recommended integration approach.</p> <p>Integration note:</p> <p>Dependent on the integration approach, related measures for <i>concurrency protection</i> and other aspects (like project-specific integration requirements, Watchdog integration requirements, etc.) have to be applied! The implementation of this callout has always to be checked and adapted against all integration aspects and requirements, which <i>cannot be known/provided by a generic BSW module implementation</i>.</p> <p>Consult the related documentation (such as your project integration manual, the <i>AutoCore_Generic_Memory_Stack_documentation</i>, etc.) for correct integration of the memory stack main functions (according to reentrancy characteristics, possible trigger contexts, etc.).</p>
Multiplicity	1..1
Type	STRING
Default value	<code>Appl_TriggerMemoryStack</code>
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemUserControlledWIRBitSupport
Label	User controlled WarningIndicatorRequested(WIR)-bit
Description	<p><i>This configuration parameter defines if user controlled WarningIndicatorRequested-bit is supported.</i></p> <p>This feature allows the parallel Dem internal handling and the external handling (e.g. by an especial fail-safe SW-C) of the WIR-bit of an event in Dem, by offering of the API <code>Dem_SetWIRStatus()</code>.</p> <ul style="list-style-type: none"> ▶ true: Dem provides the API <code>Dem_SetWIRStatus()</code> to control (set/reset) the WarningIndicatorRequested-bit of a configured event (in Dem) regarding to e.g. fail-safe state.



	<ul style="list-style-type: none"> ▶ <code>false</code>: The setting of the WIR-bit of an event is exclusively controlled by the Dem internal WIR-bit handling. ▶ The user controlled WarningIndicatorRequested-bit is only stored volatile, therefore after a restart of the Dem module, the SW-C is responsible to re-inform the Dem about the requested failsafe status, if this is the case. If the former requested WIR bit is no longer needed after the restart, the failsafe SW-C must trigger the reset of the WIR-bit by calling the <i>Dem_SetWIRStatus()</i> API. <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ <code>DemRteUsage</code>: must be set to <code>true</code> to enable support for user controlled WarningIndicatorRequested-bit control by <i>fail-safe</i> SW-C.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemDataElementDefaultEndianness	
Label	Type Of Default Data Element Endianness	
Description	<p>This parameter defines the default endianness of the data belonging to a data element which is applicable if the <code>DemExternalSRDataElementClass</code> does not define a endianness.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ <code>BIG_ENDIAN</code>: Most significant byte shall come at the lowest address. ▶ <code>LITTLE_ENDIAN</code>: Most significant byte shall come highest address. ▶ <code>OPAQUE</code>: opaque data endianness (copy bytes in the order they are represented). 	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	OPAQUE	
Range	BIG_ENDIAN LITTLE_ENDIAN OPAQUE	
Configuration class	PreCompile:	VariantPreCompile



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

Parameter Name	DemIncludeProjectSpecificCustomizationSupport	
Label	Project Specific Customization Support	
Description	<p>This configuration switch defines whether support for project-specific customization is enabled or not.</p> <ul style="list-style-type: none"> ▶ true: support for project-specific customization is enabled ▶ false: support for project-specific customization is disabled <p>When this parameter is enabled, the template header-file Dem_Custonization.h will be included in the Dem-internal header files structure. This template file can be edited by the user for the inclusion of project-specific customization. Note: The project-specific APIs must not access any Dem-internal data structures and functions because of concurrence limitations.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.3.1.23. DemClient

Parameters included		
Parameter name		Multiplicity
DemClientId		1..1

Parameter Name	DemClientId	
Label	Client Identifier	
Description	Defines a unique identifier for a Dem client. This number is used by this client in the ClientId parameter in all API with this parameter.	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	



5.3.1.24. DemServiceAPI

Parameters included	
Parameter name	Multiplicity
DemEnableASR40ServiceAPI	1..1
DemEnableASR42ServiceAPI	1..1
DemDefaultASRServiceAPI	1..1

Parameter Name	DemEnableASR40ServiceAPI	
Label	Enable AUTOSAR 4.0 service API	
Description	If enabled, the AUTOSAR 4.0.3 service interface package ASR40 is provided in addition to the configured default service interface package.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemEnableASR42ServiceAPI	
Label	Enable AUTOSAR 4.2 service API	
Description	If enabled, the AUTOSAR 4.2.1 service interface package ASR42 is provided in addition to the configured default service interface package. The Cddlf service interface for ClearDTC is not included in the ASR42 package. The additional ASR42 APIs GetDebouncingOfEvent(), GetEventFreezeFrameDataEx() and GetEventExtendedDataRecordEx() are available as stub only.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemDefaultASRServiceAPI	
Label	Default AUTOSAR service API	
Description	Defines the AUTOSAR version used as base for creating the service API: ► AUTOSAR_40: AUTOSAR 4.0.3 is used as base.	



- ▶ **AUTOSAR_42:** AUTOSAR 4.2.1 is used as base.
Main differences to AUTOSAR_40 selection:
 - Additional operation: ClearDTC
 - Additional operations: GetDebouncingOfEvent, GetEventFreezeFrameDataEx and GetEventExtendedDataRecordEx. Functionality not supported. APIs available as stubs only.

- ▶ **AUTOSAR_43:** AUTOSAR 4.3.1 is used as base.
Main differences to AUTOSAR_42 selection:
 - Removed operations: GetEventFreezeFrameData, GetEventExtendedDataRecord.

See the table below for additional info.

Note: **Availability** column shows existence of operation with respect to the AUTOSAR version selected by DemDefaultASRServiceAPI parameter.

Note: The **Prototype** column specifies the AUTOSAR version to which the operation has a matching prototype.

Client-Server Interface	Client-Server Operation	Prototype	Availability	Comments
Diagnostic-Monitor	SetEventStatus	4.0.3	all	-
Diagnostic-Monitor	ResetEventStatus	4.0.3	all	-
Diagnostic-Monitor	ResetEvent-DebounceStatus	4.2.1	all	-
Diagnostic-Monitor	Prestore-FreezeFrame	4.0.3	all	-
Diagnostic-Monitor	ClearPre-storedFreeze-Frame	4.0.3	all	-
Diagnostic-Monitor	SetEventDisabled	4.0.3	all	Generated if parameter DemOBDSupport = 'true'.
DiagnosticInfo	GetEventStatus	4.0.3	all	-
DiagnosticInfo	GetEventFailed	4.0.3	all	-



	DiagnosticInfo	GetEventTest- ed	4.0.3	all	-
	DiagnosticInfo	GetDT- COfEvent	4.0.3	all	-
	DiagnosticInfo	GetFaultDe- tectionCounter	4.0.3 4.3.1	all	With AU- TOSAR_43 operation also returns DEM_- E_NO_FDC_- AVAILABLE.
	DiagnosticInfo	GetEvent- FreezeFrame- Data	4.0.3 4.2.1	AUTOSAR_40 AUTOSAR_42	Additional pos- sible error val- ues for AU- TOSAR 4.2.1. Removed from AU- TOSAR_43.
	DiagnosticInfo	GetEven- tExtended- DataRecord	4.0.3	AUTOSAR_40 AUTOSAR_42	Removed from AU- TOSAR_43.
	DiagnosticInfo	GetDebounce- gOfEvent	4.2.2	>=AU- TOSAR_42	Only Stub im- plementation.
	DiagnosticInfo	GetEven- tExtended- DataRecordEx	4.2.2	>=AU- TOSAR_42	Only Stub im- plementation.
	DiagnosticInfo	GetEvent- FreezeFrame- DataEx	4.2.2	>=AU- TOSAR_42	Only Stub im- plementation.
	GeneralDiag- nosticInfo	GetEventSta- tus	4.0.3	all	-
	GeneralDiag- nosticInfo	GetEvent- Failed	4.0.3	all	-
	GeneralDiag- nosticInfo	GetEventTest- ed	4.0.3	all	-
	GeneralDiag- nosticInfo	GetDT- COfEvent	4.0.3	all	-
	GeneralDiag- nosticInfo	GetFaultDe- tectionCounter	4.0.3 4.3.1	all	With AU- TOSAR_43 operation also



				returns DEM_- E_NO_FDC_- AVAILABLE.
GeneralDiagnosticInfo	GetEvent- FreezeFrame- Data	4.0.3 4.2.1	AUTOSAR_40 AUTOSAR_42	Additional pos- sible error val- ues for AU- TOSAR 4.2.1. Removed from AU- TOSAR_43.
GeneralDiagnosticInfo	GetEven- tExtended- DataRecord	4.0.3	AUTOSAR_40 AUTOSAR_42	Removed from AU- TOSAR_43.
GeneralDiagnosticInfo	GetDebouncein- gOfEvent	4.2.2	>=AU- TOSAR_42	Only Stub im- plementation.
GeneralDiagnosticInfo	GetEven- tExtended- DataRecordEx	4.2.2	>=AU- TOSAR_42	Only Stub im- plementation.
GeneralDiagnosticInfo	GetEvent- FreezeFrame- DataEx	4.2.2	>=AU- TOSAR_42	Only Stub im- plementation.
OperationCycle	SetOpera- tionCycleState	4.0.3	all	Generated if parame- ter DemOp- erationCy- cleProcessing = 'DEM_PRO- CESS_OP- CYC_STATE'.
OperationCycle	SetOpera- tionCycleCnt- Value	4.0.3	all	Generated if parame- ter DemOp- erationCy- cleProcessing = 'DEM_PRO- CESS_OP- CYC_COUNTER'.
OperationCycle	SetCycleQual- ified	4.3.1	all	-



	AgingCycle	SetAgingCycleState	4.0.3	all	-
	ExternalAgingCycle	SetAgingCycleCounterValue	4.0.3	all	-
	EnableCondition	SetEnableCondition	4.0.3	all	-
	StorageCondition	SetStorageCondition	4.0.3	all	-
	IndicatorStatus	GetIndicatorStatus	4.0.3	all	-
	EvMemOverflowIndication	GetEventMemoryOverflow	4.0.3	all	-
	EvMemOverflowIndication	GetNumberOfEventMemoryEntries	4.2.1	all	-
	DTCSuppression	SetDTCSuppression	4.0.3	all	Generated if parameter DemSuppressionSupport = 'DEM_DTC_SUPPRESSION'.
	EventAvailable	SetEventAvailable	4.2.1	all	Generated if parameter DemAvailabilitySupport = 'DEM_EVENT_AVAILABILITY'.
	SetClearDTC	SetClearDTC	4.2.1	all	Operation generated only if parameter DemOBDSupportKind = 'DEM_OBD_MASTER_ECU'.



	CddIf	ClearDTC	4.2.2	>=AU-TOSAR_42	Available only in the default service interface package.
	IUMPRNumerator	ReplIUMPRFault Detect	4.0.3	all	-
	IUMPRDenominator	ReplIUMPRDen Lock	4.0.3	all	-
	IUMPRDenominator	ReplIUMPRDen Release	4.0.3	all	-
	PfcCyclePfc-CycleQualified	SetPfcCycleSetPfcCycleQualified	4.1.3	all	Generated if parameter DemOBDSupport = 'true'.
	PfcCyclePfc-CycleQualified	GetPfcCycle-Qualified	4.1.3	all	Generated if parameter DemOBDSupport = 'true'.
	ResetReadiness	ResetReadiness	-	all	Non AU-TOSAR. Generated if DemOBDSupport = 'true'.
	PowerTakeOff	SetPtoStatus	4.0.3	all	Generated if parameter DemOBDSupport = 'true'.
	Callback-InitMonitor-ForEvent	InitMonitor-ForEvent	4.0.3	all	Available only in the default service interface package.
	CallbackInit-MonitorFor-Function	InitMonitorFor- Function	4.0.3	all	Available only in the default service interface package.
	CallbackEven-tStatusChange	EventStatusChanged	4.0.3 4.2.1	all	Selecting AU-TOSAR_40, prototype is ASR 4.0.3,



				otherwise 4.- 2.1. Available only in the de- fault service in- terface pack- age.
	GeneralCall- backEventSta- tusChange	EventSta- tusChanged	4.0.3 4.2.1	all
	Callback- DTCSta- tusChange	DTCSta- tusChanged	4.0.3	all
	CallbackEvent- DataChanged	EventData- aChanged	4.0.3 4.2.1	all
	GeneralCall- backEventDat- aChanged	EventData- aChanged	4.0.3 4.2.1	all



	Callback- ClearEventAI- lowed lowed	ClearEventAI- lowed	4.0.3	all	Available only in the default service inter- face package.
	CSDaSer- vices_[Sync- DataElement]	ReadData	4.0.3	AUTOSAR_40	Available only in the default service inter- face package.
	DataSer- vices_[Sync- DataElement]	ReadData	4.2.1	>=AU- TOSAR_42	Available only in the default service inter- face package.
	CallbackGet- FaultDetect- Counter	GetFaultDe- tectionCounter	4.0.3	all	Available only in the default service inter- face package.
	EventStatus	SetWIRStatus	4.2.1	all	Generated if parameter De- mUserCon- trolledWIR- BitSupport = 'true'.
Multiplicity	1..1				
Type	ENUMERATION				
Default value	AUTOSAR_40				
Range	AUTOSAR_40 AUTOSAR_42 AUTOSAR_43				
Configuration class	VariantPreCompile:		VariantPreCompile		
Origin	Elektrobit Automotive GmbH				

5.3.1.25. DemBSWDistribution

Containers included		
Container name	Multiplicity	Description



Containers included

DemSatelliteInstanceConfig	0..n	<p>This container defines the Dem satellite cores and their specific data related to the pre-initialization phase of the Dem module.</p> <p>Each Os core (which is not selected as Dem master instance) shall configure a Dem satellite instance.</p>
--	------	---

Parameters included

Parameter name	Multiplicity
DemDistributedBSWEventReporting	1..1
DemMasterCoreId	1..1
DemSRQueueLengthDistributedBSWEventReporting	1..1

Parameter Name	DemDistributedBSWEventReporting
Label	Distributed BSW Event Reporting
Description	<p>This configuration switch defines whether support for distributed BSW event reporting via <code>Dem_ReportErrorStatus()</code> from multiple cores is enabled or not.</p> <p>The Dem satellite instances will only collect the reports from their related cores and send these reports to the Dem master instance, which executes the actual event processing.</p> <ul style="list-style-type: none"> ▶ TRUE: <code>Dem_ReportErrorStatus()</code> can be used from BSW modules that are integrated on cores with a Dem satellite instance. All other Dem APIs must only be called from modules of the Dem master instance core. ▶ FALSE: Any Dem API must only be called from modules of the core on which the Dem module is integrated.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemMasterCoreId
Label	Dem Master Core ID
Description	This parameter defines the OsCoreID associated with the <i>Dem master/main instance</i> .



	On multi-core platforms with DemDistributedBSWEventReporting disabled there will only be the Dem master instance which must be integrated to the core which also integrates all other modules reporting BSW errors to the Dem.
Multiplicity	1..1
Type	INTEGER
Default value	0
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemSRQueueLengthDistributedBSWEventReporting	
Label	Dem SR Queue Length for BSW event reporting	
Description	This parameter defines the Sender / Receiver queue length for BSW event reports received on the Dem master instance core. This queue is necessary in case multiple reports are triggered from the Dem satellite instances in order to buffer those on the Dem master instance and process them sequentially.	
Multiplicity	1..1	
Type	INTEGER	
Default value	20	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

5.3.1.26. DemSatelliteInstanceConfig

Parameters included		
Parameter name		Multiplicity
DemSatelliteCoreId		1..1
DemSatelliteInstanceLocalBswEventBufferSize		1..1

Parameter Name	DemSatelliteCoreId
Label	Satellite Core Id
Description	This parameter defines the OsCoreID associated with the related Dem satellite instance.
Multiplicity	1..1
Type	INTEGER



Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemSatelliteInstanceLocalBswEventBufferSize	
Label	Local BSW Event Queue Size	
Description	<p>Maximum number of local event-queue for handling of BSW event reports for the dedicated Dem satellite instance.</p> <p>This local buffer on a Dem satellite instance is used for the multi-core functionality, to store Dem BSW events reported when the Dem is not yet fully initialized, i.-e. pre-initialized.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	10	
Range	<p><=255</p> <p>>=1</p>	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

5.3.1.27. DemCallbackDTCStatusChanged

Parameters included		
Parameter name	Multiplicity	
DemCallbackDTCStatusChangedFnc	0..1	

Parameter Name	DemCallbackDTCStatusChangedFnc
Label	DTCStatusChanged Callback C-Function
Description	<p>For a description of the callback behavior, refer to the parent container.</p> <p>The expected C callback prototype is:</p> <pre>Std_ReturnType &lt;Mod&gt;_DemTriggerOnDTCStatus(uint32 DTC, uint8 DTCStatusOld, uint8 DTCStatusNew)</pre> <p>To enable usage of this C callback function, proceed as follows:</p> <ol style="list-style-type: none"> 1. Add callback function to the callback function list DTCStatusChanged Callback List



	<p>2. Activate this parameter</p> <p>3. Configure the related C function name, e.g. <code>Cdd_DemTriggerOnDTCStatus</code></p> <p>4. Configure the related C header file inclusion using the parameter <code>DemHeaderFileInclusion</code></p> <p>Note: If the parameter <code>DemTriggerDcmReports</code> is enabled, this parameter shall not be "Dcm_DemTriggerOnDTCStatus".</p>
Multiplicity	0..1
Type	FUNCTION-NAME
Default value	Module_DemTriggerOnDTCStatus
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.3.1.28. DemDataElementClass

Containers included		
Container name	Multiplicity	Description
<u>DemExternalCSDataElement-Class</u>	1..1	<p>This container contains the configuration (parameters) for an external client/server based data element class.</p> <p>It defines, how the Dem can obtain the value of the data element from either a SW-C or another BSW module. Whether a client/server or a C function-call is used, is defined by port <code>DemDataElementUsePort</code>.</p>
<u>DemExternalSRDataElement-Class</u>	1..1	<p>This container contains the configuration (parameters) for an external sender/receiver based data element class. It defines, how the Dem can obtain the value of the data element from a SW-C, by using a sender/receiver port.</p>
<u>DemInternalDataElement-Class</u>	1..1	<p>This container contains the configuration (parameters) for an internal data element class.</p>

5.3.1.29. DemExternalCSDataElementClass

Parameters included	
Parameter name	Multiplicity

**Parameters included**

DemDataElementDataSize	1..1
DemDataElementReadFnc	0..1
DemDataElementUsePort	1..1
DemDataElementInterfaceWithEventParameter	1..1

Parameter Name**DemDataElementDataSize****Label**

Data Element Data Size

Description

Defines the size of the data element in bytes.

Multiplicity

1..1

Type

INTEGER

Default value

1

Range

<=255

>=1

Configuration class**VariantPreCompile:**

VariantPreCompile

Origin

AUTOSAR_ECUC

Parameter Name**DemDataElementReadFnc****Label**

DataElementRead Callback C-Function

Description

For a description of the callback behavior, refer to the DemExternalCSDataElementClass description.

The expected C callback prototype depends on the parameter DemDataElementInterfaceWithEventParameter

To enable usage of this C callback function, proceed as follows:

1. Configure a Data Element Class of type DemExternalCSDataElementClass in the DemDataElementClass tab
2. Disable Data Element use Port for this callback using the parameter DemDataElementUsePort
3. Activate this parameter
4. Configure the related C function name, e.g. Cdd_EbTest_DemReadDataElement_0
5. Configure the related C header file inclusion using the parameter DemHeaderFileInclusion

Multiplicity

0..1



Type	FUNCTION-NAME	
Default value	Module_DataElementRead	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemDataElementUsePort	
Label	Data Element use Port	
Description	<p>If the parameter is set to <code>True</code>, a R-Port is generated, to obtain the data element (interface <code>DataServices < SyncDataElement ></code>).</p> <p>If the parameter is set to <code>False</code>, the information is obtained by C function-call on another BSW module specified by the parameter <code>DemDataElementReadFnc</code>.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemDataElementInterfaceWithEventParameter	
Label	Read Function requires EventId parameter	
Description	<p>Parameter used for accessing the EventId during the capture of environment data.</p> <p>If the parameter is set to <code>False</code>, the expected callback prototype is the AUTOSAR compatible prototype: <code>Std_ReturnType < Mod > _ - DemRead < DataElement > (uint8* Buffer)</code></p> <p>If the parameter is set to <code>True</code>, the callback used to obtain the data element will use EventId as an extra parameter, <code>Std_ReturnType < Mod > _ - DemRead < DataElement > (uint8* Buffer, Dem_EventIdType EventId)</code></p> <p>Note: It is recommended that the symbolic names provided by the Dem module are used for the EventId, and not the numeric values since the Ids depend on the current Dem configuration. The callback shall avoid interpreting the EventId in any way, as only a hand-through mechanism is allowed.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

5.3.1.30. DemExternalSRDataElementClass

Parameters included	
Parameter name	Multiplicity
DemDataElementDataSize	1..1
DemDataElementDataType	1..1
DemDataElementEndianness	0..1

Parameter Name	DemDataElementDataSize
Label	Data Element Data Size
Description	Defines the size of the data element in bytes.
Multiplicity	1..1
Type	INTEGER
Default value	1
Configuration class	VariantPreCompile:
Origin	AUTOSAR_ECUC

Parameter Name	DemDataElementDataType
Label	Data Element Data Type
Description	Defines the type of the data element. Note: The other data types (boolean or signed integers) could still be connected via RTE Data Conversion feature.
Multiplicity	1..1
Type	ENUMERATION
Default value	UINT8
Range	UINT8 UINT16 UINT32
Configuration class	VariantPreCompile:
Origin	AUTOSAR_ECUC



Parameter Name	DemDataElementEndianness	
Label	Data Element Endianness	
Description	<p>Defines the endianness of the data belonging to an external data. If no DemDataElementEndianness is defined the value of DemDataElementDefaultEndianness is applicable.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ BIG_ENDIAN: Most significant byte shall come at the lowest address. ▶ LITTLE_ENDIAN: Most significant byte shall come highest address. ▶ OPAQUE: opaque data endianness (copy bytes in the order they are represented). 	
Multiplicity	0..1	
Type	ENUMERATION	
Default value	OPAQUE	
Range	BIG_ENDIAN LITTLE_ENDIAN OPAQUE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.3.1.31. DemInternalDataElementClass

Parameters included	
Parameter name	Multiplicity
DemDataElementDataSize	1..1
DemInternalDataElement	1..1

Parameter Name	DemDataElementDataSize	
Label	Data Element Data Size	
Description	Defines the size of the data element in bytes.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Configuration class	VariantPreCompile:	VariantPreCompile



Origin	AUTOSAR_ECUC
Parameter Name	DemInternalDataElement
Label	Internal Data Element
Description	<p>This parameter defines the Dem-internal data value, which is mapped to the data element.</p> <p>To allow the mapping of DEM_SI30, DEM_FDC12, and OCC values the parameter DemExtendedDataCapture needs to be set to DEM_TRIGGER_ON_FDC_THRESHOLD. The event using the related extended data class has to use counter-based or monitor internal debouncing - otherwise the counters will not be supported.</p> <p>To allow the mapping of DEM_CURRENT_FDC, the event using the related extended data class has to configure the FDC callback function if monitor internal debouncing is used.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_AGINGCTR: map Dem-internal aging counter count-up. ▶ DEM_AGINGCTR_UPCNT: map Dem-internal aging counter count-up. ▶ DEM_AGINGCTR_DOWNCNT: map Dem-internal aging counter count-down. ▶ DEM_OCCCTR: map Dem-internal occurrence counter: In case the parameter DemFreezeFrameCapture is set to DEM_TRIGGER_ON_FDC_THRESHOLD, this counter will count the number of times the freeze frame is captured. (This is different to Dem523 and Dem524.) ▶ DEM_OVFLIND: map Dem-internal overflow indication. ▶ DEM_SIGNIFICANCE: map (static) Dem-internal event significance. ▶ DEM_SI30 (vendor-specific extension): map Dem-internal DTC status indicators. ▶ DEM_CURRENT_FDC (vendor-specific extension): map Dem-internal fault detection counter: The singed ranges of the FDC are mapped as follows: [0..127] -> [0x00..0x7F], [-1..128] -> [0xFF..0x80]. ▶ DEM_FDC12 (vendor-specific extension): map Dem-internal maximum fault detection counter since last clear: The maximum value, which is equal to or greater than unconfirmedDTCLimit, that the internal counter has reached since the last time DTC information was cleared. ▶ DEM_OCC1 (vendor-specific extension): map Dem-internal operation cycles since last unconfirmedDTC counter: This counter represents the number of operation cycles since unconfirmedDTC threshold was reached last time.



If the unconfirmedDTC threshold equals the failed threshold, the counter counts operation cycles since last qualified failed result.

- ▶ **DEM_OCC2** (vendor-specific extension): map Dem-internal unconfirmed-DTC aging counter: This counter represents the number of operation cycles having only qualified passed results since unconfirmedDTC threshold was reached last time. If the unconfirmedDTC threshold equals the failed threshold, the behavior is similar as behavior of the 'aging counter' (DEM_AGINGCTR) with the difference that it is update on qualified passed result instead on end of operation cycle.
- ▶ **DEM_OCC3** (vendor-specific extension): map Dem-internal operation cycles since first unconfirmedDTC counter: This counter represents the number of operation cycles since unconfirmedDTC threshold was reached first time. If the unconfirmedDTC threshold equals the failed threshold, the counter counts operation cycles since first qualified failed result.
- ▶ **DEM_OCC4** (vendor-specific extension): map Dem-internal unconfirmedDTC operation cycle counter: This counter represents the number of operation cycles having unconfirmedDTC threshold reached. If the unconfirmedDTC threshold equals the failed threshold, the counter counts operation cycles having failed threshold reached.
- ▶ **DEM_OCC5** (vendor-specific extension): map Dem-internal unconfirmedDTC warm-up cycle counter: This represents the number of warm-up cycles (WUC) of emission related events, that are reported as failed until they become confirmed.
- ▶ **DEM_OCC6** (vendor-specific extension): map Dem-internal consecutive failed operation cycle counter: This counter represents the number of consecutive operation cycles having a qualified failed result.
- ▶ **DEM_OCC7** (vendor-specific extension): map Dem-internal qualified/unconfirmedDTC operation cycle counter: This counter represents the number of operation cycles having a qualified result or having unconfirmedDTC threshold reached (since unconfirmedDTC threshold was reached first time). If the unconfirmedDTC threshold equals the failed threshold, the counter counts operation cycles having qualified result.
- ▶ **DEM_PRIORITY**: map Dem-internal event priority.
- ▶ **DEM_EVENT_ID**: map Dem-internal eventId: EventId of the event which causes the event entry or the last update of it. This data is not static for combined events.
- ▶ **DEM_ROOT_EVENT_ID**: map Dem-internal root event Id: EventId of the root event which caused the event entry or the last update of it. This data is on-



	<p>ly useful and could differ from DEM_EVENT_ID in case of the entry is triggered indirectly by a DemMultiEventTriggeringSlaveEventRef.</p> <ul style="list-style-type: none"> ▶ DEM_CYCLES_SINCE_LAST_FAILED: represents the number of operation cycles since the DTC fault detection counter last reached its maximum value +127 (since DTC information was last cleared). The count includes also operation cycles in which the test has not been completed. <i>Note:</i> the lower limit 0x00U - event is not yet qualified OR no operation cycle is completed since last occurrence (it is qualified in the current operation cycle) ▶ Note: Depending on the value of configuration parameter DemStatusBitStorageTestFailedPerEvent (or general configuration parameter DemStatusBitStorageTestFailed, if not configured per event), the functionality of the CSLF counter is impacted as follows:<ul style="list-style-type: none"> * ENABLED: the counter will continue to increment itself if the qualification of the event is FAILED in a new power-cycle. Only an explicit PASSED inbetween FAILED qualifications will cause a transition of the TF bit from 0 to 1 and reset the counter. * DISABLED: the value of TF bit is not stored inbetween power cycles, a FAILED is considered a new occurrence causing the counter CSLF to reset on the new power cycle. ▶ DEM_CYCLES_SINCE_FIRST_FAILED: represents the number of operation cycles since the DTC fault detection counter first reached its maximum value of +127 (since DTC information was last cleared). The count includes also operation cycles in which the test has not been completed. <i>Note:</i> the lower limit: 0x00U - event is not yet qualified OR no operation cycle is completed since then (it is qualified in the current operation cycle). <p><i>Note:</i> Internal data elements bigger than 1 byte (e.g. event id - 2 bytes) are serialized as little endian format (least significant byte first).</p> <p><i>Note:</i> Due to the fact that these counters described above are specified as one byte, the maximum value which can be reached is 255 (0xFFU), even if more cycles are elapsed.</p>
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_AGINGCTR
Range	DEM_AGINGCTR DEM_AGINGCTR_UPCNT DEM_AGINGCTR_DOWNCNT DEM_OCCCTR



	DEM_OVFLIND
	DEM_SIGNIFICANCE
	DEM_SI30
	DEM_CURRENT_FDC
	DEM_FDC12
	DEM_OCC1
	DEM_OCC2
	DEM_OCC3
	DEM_OCC4
	DEM_OCC5
	DEM_OCC6
	DEM_OCC7
	DEM_PRIORITY
	DEM_EVENT_ID
	DEM_ROOT_EVENT_ID
	DEM_CYCLES_SINCE_LAST_FAILED
	DEM_CYCLES_SINCE_FIRST_FAILED
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.3.1.32. DemDidClass

Parameters included	
Parameter name	Multiplicity
DemDidIdentifier	1..1
DemDidDataElementClassRef	1..255

Parameter Name	DemDidIdentifier
Label	DID Identifier
Description	Identifier of the Data ID.
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile



Origin	AUTOSAR_ECUC
Parameter Name	DemDidDataElementClassRef
Label	Did Data Element Class
Description	This reference contains the link to a data element class. Internal data elements can not be referenced in the current implementation.
Multiplicity	1..255
Type	CHOICE-REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.3.1.33. DemEnableCondition

Parameters included	
Parameter name	Multiplicity
DemEnableConditionId	1..1
DemEnableConditionStatus	1..1
Parameter Name	
Parameter Name	DemEnableConditionId
Label	Enable Condition Id
Description	<p>Defines a unique enable condition Id. This parameter should not be changeable by user, because the Id should be generated by Dem itself to prevent gaps and multiple use of an Id.</p> <p>The enable condition Ids shall be sequentially ordered beginning with 0 and no gaps in between.</p> <p>The Handle-Id Wizard can be used to set this value automatically.</p>
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	
Parameter Name	DemEnableConditionStatus
Label	Initial Enable Condition Status
Description	Defines the initial status for enable or disable of acceptance of event reports of a diagnostic event.



	The value is the initialization after power up (before this condition is reported the first time). <ul style="list-style-type: none"> ▶ true: acceptance of a diagnostic event enabled ▶ false: acceptance of a diagnostic event disabled
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

5.3.1.34. DemEnableConditionGroup

Parameters included	
Parameter name	Multiplicity
DemEnableConditionRef	1..255
Parameter Name	
Label	Enable Condition
Description	Defines a Enable Condition, which allows for switching on/off the acceptance of event status reports. This parameter is optional and depends on manufacturer.
Multiplicity	1..255
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU V1.0.0

5.3.1.35. DemMultiEventTriggering

Parameters included	
Parameter name	Multiplicity
DemMultiEventTriggeringAlternativeBehavior	1..1
DemMultiEventTriggeringMasterEventRef	1..1

**Parameters included**

DemMultiEventTriggeringSlaveEventRef	1..255
--	--------

Parameter Name	DemMultiEventTriggeringAlternativeBehavior
Label	Alternative Behavior
Description	<p>This parameter enables exclusive replacement of the master event by one of the slave events, i.e. the slave event is processed instead of the master event.</p> <p>If none slave event is enabled then only the master event is processed as it would be done if no DemMultiEventTriggering is configured for the master.</p> <p>For the exclusive replacement of the master event is necessary that, all referenced slave events have a valid <i>DemEnableConditionGroupRef</i>.</p> <p>For an easier handling each of this groups, the <i>DemEnableConditionGroupRef</i> shall have exactly one enable condition defined.</p> <p>Furthermore for maximum one slave event the enable condition group has to be enabled by <i>DemEnableConditionStatus</i> for maximum one of the slave events. It should be ensured also during run-time that, maximum one of the slaves are enabled. Otherwise it is not predictable which slave event replaces the master event.</p> <ul style="list-style-type: none"> ▶ DEM_MULTIEVENTTRIGGERING_ALT_NONE: default behavior is enabled ▶ DEM_MULTIEVENTTRIGGERING_ALT_EXCLUSIVE_REPLACE: exclusive replacement is enabled
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_MULTIEVENTTRIGGERING_ALT_NONE
Range	DEM_MULTIEVENTTRIGGERING_ALT_NONE DEM_MULTIEVENTTRIGGERING_ALT_EXCLUSIVE_REPLACE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemMultiEventTriggeringMasterEventRef
Label	Master Event
Description	Reference to the event that will trigger other events upon reception of this event.
Multiplicity	1..1
Type	REFERENCE



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU R4.4.0	

Parameter Name	DemMultiEventTriggeringSlaveEventRef	
Label	Slave Event	
Description	Reference to the event that is triggered upon triggering the master event.	
Multiplicity	1..255	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU R4.4.0	

5.3.1.36. DemExtendedDataClass

Parameters included		
Parameter name		Multiplicity
DemExtendedDataRecordClassRef		1..253
Parameter Name		
Parameter Name	DemExtendedDataRecordClassRef	
Label	Extended Data Record	
Description	This reference contains the link to an extended data class record.	
Multiplicity	1..253	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

5.3.1.37. DemExtendedDataRecordClass

Parameters included		
Parameter name		Multiplicity
DemExtendedDataRecordNumber		1..1
DemExtendedDataRecordUpdate		0..1
DemDataElementClassRef		1..255
DemExtendedDataRecordAlwaysAvailable		1..1



Parameter Name	DemExtendedDataRecordNumber	
Label	Extended Data Record Number	
Description	<p>This configuration parameter specifies an unique identifier for an extended data record.</p> <p>One or more extended data records can be assigned to one diagnostic event/DTC.</p> <p>0x00 and 0xF0 – 0xFF are reserved by ISO (therefore the maximal value equals 0xEF).</p>	
Multiplicity	1..1	
Type	INTEGER	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemExtendedDataRecordUpdate	
Label	Extended Data Record Update	
Description	<p>This parameter defines the case, when the extended data record is stored/updated.</p> <p><i>If the parameter is disabled, the value is taken as DEM_UPDATE_RECORD_YES by default.</i></p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_UPDATE_RECORD_NO: This extended data record is only captured for new event memory entries. ▶ DEM_UPDATE_RECORD_YES: This extended data record is captured every time. 	
Multiplicity	0..1	
Type	ENUMERATION	
Default value	DEM_UPDATE_RECORD_YES	
Range	DEM_UPDATE_RECORD_NO DEM_UPDATE_RECORD_YES	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemDataElementClassRef	
-----------------------	-------------------------------	--



Label	Data Element Class	
Description	This reference contains the link to a data element class.	
Multiplicity	1..255	
Type	CHOICE-REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	
Parameter Name	DemExtendedDataRecordAlwaysAvailable	
Label	Data Record always available	
Description	<p>Defines if the extended data record is available without an event memory entry.</p> <p>The value can be set to true only if all referenced data elements are available without an event memory entry.</p> <p>The following internal data elements are always available:</p> <ul style="list-style-type: none"> ▶ DEM_CURRENT_FDC ▶ DEM_OVFLIND ▶ DEM_PRIORITY ▶ DEM_SIGNIFICANCE <p>Note: Configuring the extended data record as always available has impact on the following APIs:</p> <ul style="list-style-type: none"> ▶ Dem_GetEventExtendedDataRecord() ▶ Dem_GetNextExtendedDataRecord() ▶ DemGetSizeOfExtendedDataRecordSelection() 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

5.3.1.38. DemFreezeFrameClass

Parameters included	
Parameter name	Multiplicity



Parameters included

DemDidClassRef	1..255
--------------------------------	--------

Parameter Name	DemDidClassRef	
Label	Did Class	
Description	For OBD-relevant data. Multiple PIDs can be relevant per freeze frame.	
Multiplicity	1..255	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.3.1.39. DemFreezeFrameRecNumClass

Parameters included

Parameter name	Multiplicity
DemFreezeFrameRecordNumber	1..254

Parameter Name	DemFreezeFrameRecordNumber	
Label	Freeze Frame Record	
Description	This parameter defines a record number for a freeze frame record. This record number is unique per freeze frame record number class. The range of this value is defined by ISO 14229-1 (0x01 .. 0xFE).	
Multiplicity	1..254	
Type	INTEGER	
Default value	1	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.3.1.40. DemSPNClass

Parameters included

Parameter name	Multiplicity
----------------	--------------



Parameters included

DemSPNId	1..1
DemSPNDataElementClassRef	1..1

Parameter Name	DemSPNId
Label	Suspect Parameter Number
Description	Suspect parameter number
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC R4.2.2

Parameter Name	DemSPNDataElementClassRef
Label	SPN Data Element
Description	This reference contains the link to a data element class. Internal data elements can not be referenced in the current implementation.
Multiplicity	1..1
Type	CHOICE-REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC R4.2.2

5.3.1.41. DemJ1939FreezeFrameClass

Parameters included

Parameter name	Multiplicity
DemSPNClassRef	1..255

Parameter Name	DemSPNClassRef
Label	SPN Class
Description	This container contains the combinations of SPNs for a J1939 relevant freeze frame.
Multiplicity	1..255
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile



Origin	AUTOSAR_ECUC R4.2.2
--------	---------------------

5.3.1.42. DemGeneralOBD

Parameters included	
Parameter name	Multiplicity
DemOBDEventDisplacement	1..1
DemOBDCentralizedPID21Handling	1..1
DemOBDCentralizedPID31Handling	1..1
DemOBDCompliancy	1..1
DemOBDDelayedDCYConfirmedAndMIL	1..1
DemOBDInputAcceleratorPaddleInformation	1..1
DemOBDInputAmbientPressure	1..1
DemOBDInputAmbientTemperature	1..1
DemOBDInputDistanceInformation	1..1
DemOBDInputEngineSpeed	1..1
DemOBDInputEngineTemperature	1..1
DemOBDInputProgrammingEvent	1..1
DemOBDInputVehicleSpeed	1..1
DemOBDTIMESinceEngineStart	0..1
DemOBDEngineType	1..1

Parameter Name	DemOBDEventDisplacement
Label	OBD Event Displacement
Description	<p>This configuration switch activates/deactivates a particular displacement behavior for OBD events.</p> <p>If enabled, an OBD event that</p> <ul style="list-style-type: none"> ▶ triggers the MIL indicator or ▶ holds the legislative Freeze Frame or ▶ has the Pending-DTC status bit set <p>will not be considered during the displacement mechanism. Potential memory entries of such events will not be displaced if the new reported event has the same priority.</p>



	<p>The OBD event condition will be ignored if the new reported event has a higher priority.</p> <p>If the event displacement strategy is configured to <code>DEM_DISPLACEMENT_PRIO_OCC</code>, the activation of the OBD event displacement behavior has no effect if <code>DemEventDisplacementCriterionTNCTOC</code> is disabled, because in this case same prioritized event entries are ignored anyway.</p> <ul style="list-style-type: none"> ▶ <code>true</code>: OBD events with above listed special conditions are protected from being displaced by reported events with the same priority. ▶ <code>false</code>: The displacement mechanism will not take into account the special OBD conditions i.e. OBD events might be considered and displaced by reported events with the same (or higher) priority. <p>If the vendor-specific extension <code>DEM_TRIGGER_ON_FDC_THRESHOLD</code> is selected for <code>DemExtendedDataCapture</code> and <code>DemFreezeFrameCapture</code>, a non-standard/vendor-specific displacement algorithm is activated. It is not recommended to enable <code>DemOBDEventDisplacement</code> in this case.</p> <p>The overall support for event memory displacement is enabled/disabled by the configuration parameter <code>DemEventDisplacementSupport</code>. There is an overview of all configuration parameters available which affect the displacement behavior.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	DemOBDCentralizedPID21Handling
Label	Centralized PID21 Handling
Description	<p>Enables centralized calculation of PID \$21 retrieved from the OBD Master ECU.</p> <p>Note: The parameter option <code>true</code> (according to R4.2) is not supported in the current implementation.</p> <p><code>true</code>: Centralized calculation: <code>Dem_SetDataOfPID21()</code> and <code>Dem_GetDataOfPID21()</code> are not supported (according to R4.2). <code>Dem_DcmReadDataOfPID21()</code> will always return 0.</p> <p><code>false</code>: Local calculation: PID \$21 will be calculated Dem internally independent of <code>DemOBDSupportKind</code>.</p> <p>The interface for the <code>Dem_DcmReadDataOfPID21()</code> API is not exported to RTE.</p>



	<p>To enable the Dem internal calculation, proceed as follows:</p> <ol style="list-style-type: none"> 1. Set the parameter DemOBDDInputVehicleSpeed as DemExternalCSDDataElementClass or DemExternalSRDataElementClass and provide the application SWC that calculates the values for Vehicle Input Speed (PID \$0D). 2. Enable the parameter DemOBDTIMESinceEngineStart, set it as DemExternalCSDDataElementClass or DemExternalSRDataElementClass and provide the application SWC that calculates the values for Time Since Engine Start (PID \$1F). 3. Enable the parameter DemMILIndicatorRef. <p>The PID \$21 information can be cleared using service \$04, independent of the ClearEventAllowed setting.</p> <p>Note: This parameter was introduced with AUTOSAR R4.2.2.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	true
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU

Parameter Name	DemOBDCentralizedPID31Handling
Label	Centralized PID31 Handling
Description	<p>Enables centralized calculation of PID \$31 retrieved from the OBD Master ECU.</p> <p>Note: The parameter option <code>true</code> (according to R4.2) is not supported in the current implementation.</p> <p><code>true</code>: Centralized calculation: <code>Dem_SetDataOfPID31()</code> is not supported (according to R4.2). <code>Dem_DcmReadDataOfPID31()</code> will always return 0.</p> <p><code>false</code>: Local calculation: PID \$31 will be calculated Dem internally independent of DemOBDSupportKind.</p> <p>To enable the Dem internal calculation, proceed as follows:</p> <ol style="list-style-type: none"> 1. Set the parameter DemOBDDInputVehicleSpeed as DemExternalCSDDataElementClass and provide the application SWC that calculates the values for Vehicle Input Speed (PID \$0D). 2. Enable the parameter DemOBDTIMESinceEngineStart, set it as DemExternalCSDDataElementClass and provide the application SWC that calculates the values for Time Since Engine Start (PID \$1F).



	Note: This parameter was introduced with AUTOSAR R4.2.1.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	PreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemOBDCompliance	
Label	OBD Compliancy	
Description	<p>Configuration value to define the appropriate value to PID\$1C "OBD requirements to which vehicle or engine is certified." according to the respective standards, e.g. OBD, OBDII, JOBD etc.</p> <p>Notice as well J1979 or the "DiagnosticReadiness 1" DM5 message of J1939-73</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	5	
Range	<p><=255</p> <p>>=0</p>	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemOBDDelayedDCYConfirmedAndMIL	
Label	Enable Delayed Event Confirmation	
Description	<p>This configuration parameter controls the delayed calculation of the confirmed status for the OBD driving cycle.</p> <p>If this parameter is enabled, events having the driving cycle configured as operation/failure cycle will only be confirmed, if the driving cycle was qualified via Dem_SetCycleQualified().</p> <p>Note: This configuration parameter is not editable. To disable the delayed confirmation, the driving cycle can be qualified directly after each start of the driving cycle.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	DemOBDInputAcceleratorPaddleInformation	
Label	Input Accelerator Paddle Information	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Input variable for the accelerator paddle information, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.</p>	
Multiplicity	1..1	
Type	CHOICE-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemOBDInputAmbientPressure	
Label	Input Ambient Pressure	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Input variable for the ambient pressure, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.</p>	
Multiplicity	1..1	
Type	CHOICE-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	DemOBDInputAmbientTemperature	
Label	Input Ambient Temperature	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Input variable for the ambient temperature, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.</p>	
Multiplicity	1..1	
Type	CHOICE-REFERENCE	



Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	DemOBDInputDistanceInformation
Label	Input Distance Information
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> Input variable for the distance information, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.
Multiplicity	1..1
Type	CHOICE-REFERENCE
Configuration class	VariantPreCompile:
Origin	AUTOSAR_ECU

Parameter Name	DemOBDInputEngineSpeed
Label	Input Engine Speed
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> Input variable for the engine speed, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.
Multiplicity	1..1
Type	CHOICE-REFERENCE
Configuration class	VariantPreCompile:
Origin	AUTOSAR_ECU

Parameter Name	DemOBDInputEngineTemperature
Label	Input Engine Temperature
Description	<i>The functionality related to this parameter is not supported by the current implementation.</i> Input variable for the engine temperature, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.
Multiplicity	1..1
Type	CHOICE-REFERENCE
Configuration class	VariantPreCompile:



Origin	AUTOSAR_ECUC
Parameter Name	DemOBDInputProgrammingEvent
Label	Input Programming Event
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Input variable for the programming event, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.</p>
Multiplicity	1..1
Type	CHOICE-REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	DemOBDInputVehicleSpeed
Label	Input Vehicle Speed
Description	<p>Input variable for the vehicle speed, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.</p> <p>This parameter is required for Dem internal PID \$31 and PID \$21 calculation.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ DemOBDCentralizedPID31Handling: must be set to <code>false</code> to enable the Dem internal PID \$31 calculation. ▶ DemOBDCentralizedPID21Handling: must be set to <code>false</code> to enable the Dem internal PID \$21 calculation.
Multiplicity	1..1
Type	CHOICE-REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC
Parameter Name	DemOBDTImeSinceEngineStart
Label	Time Since Engine Start
Description	<p>Input variable for the Time Since Engine Start information, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.</p> <p>This parameter is required for Dem internal PID \$31 and PID \$21 calculation.</p> <p>Dependency on parameter(s):</p>



	<ul style="list-style-type: none"> ▶ DemOBDCentralizedPID31Handling: must be set to <code>false</code> to enable the Dem internal PID \$31 calculation. ▶ DemOBDCentralizedPID21Handling: must be set to <code>false</code> to enable the Dem internal PID \$21 calculation. <p>Note: This parameter was introduced with AUTOSAR R4.2.1.</p>
Multiplicity	0..1
Type	CHOICE-REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemOBDEngineType	
Label	OBD Engine Type	
Description	Switch to provide either Gasoline or Diesel parameters. Range: <ul style="list-style-type: none"> ▶ DEM_IGNITION_COMPRESSION: Diesel engine type. ▶ DEM_IGNITION_SPARK: Gasoline engine type. 	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	DEM_IGNITION_COMPRESSION	
Range	DEM_IGNITION_COMPRESSION DEM_IGNITION_SPARK	
Configuration class	PreCompile: VariantPreCompile	
Origin	AUTOSAR_ECUC	

5.3.1.43. DemGeneralJ1939

Parameters included	
Parameter name	Multiplicity
DemJ1939ReadingDtcSupport	1..1
DemJ1939RatioSupport	1..1
DemJ1939Readiness1Support	1..1
DemJ1939Readiness2Support	1..1

**Parameters included**

DemJ1939Readiness3Support	1..1
DemJ1939ClearDTCsupport	1..1
DemJ1939FreezeFrameSupport	1..1
DemJ1939Dm31Support	1..1
DemJ1939ExpandedFreezeFrameSupport	1..1

Parameter Name	DemJ1939ReadingDtcSupport	
Label	J1939 Reading DTC Support	
Description	<p>This configuration switch defines whether J1939 DTC readout is supported or not.</p> <p>This switches on and off the APIs Dem_J1939DcmSetDTCFilter, Dem_J1939DcmGetNumberOfFilteredDTC and Dem_J1939DcmGetNextFilteredDTC.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC R4.2.2	

Parameter Name	DemJ1939RatioSupport	
Label	J1939 Ratio Support	
Description	<p>This configuration switch defines whether J1939 ratio is supported or not.</p> <p>This switches on and off the APIs Dem_J1939DcmSetRatioFilter and Dem_J1939DcmGetNextFilteredRatio.</p> <p>Note: This configuration parameter is enabled only if OBD support is enabled.</p> <p>Note: The functionality related to this parameter is not supported by the current implementation. The corresponding APIs are implemented as stubs only.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC R4.2.2	
Parameter Name	DemJ1939Readiness1Support	



Label	J1939 Readiness 1 Support	
Description	<p>This configuration switch defines whether J1939 diagnostic readiness 1 is supported or not.</p> <p>This switches on and off the API Dem_J1939DcmReadDiagnosticReadiness1.</p> <p>Note: This configuration parameter is enabled only if OBD support is enabled.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC R4.2.2	

Parameter Name	DemJ1939Readiness2Support	
Label	J1939 Readiness 2 Support	
Description	<p>This configuration switch defines whether J1939 diagnostic readiness 2 is supported or not.</p> <p>This switches on and off the API Dem_J1939DcmReadDiagnosticReadiness2.</p> <p>Note: This API is implemented only as stub.</p> <p>Note: This configuration parameter is enabled only if OBD support is enabled.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC R4.2.2	

Parameter Name	DemJ1939Readiness3Support	
Label	J1939 Readiness 3 Support	
Description	<p>This configuration switch defines whether J1939 diagnostic readiness 3 is supported or not.</p> <p>This switches on and off the API Dem_J1939DcmReadDiagnosticReadiness3.</p> <p>Note: This API is implemented only as stub.</p> <p>Note: This configuration parameter is enabled only if OBD support is enabled.</p>	
Multiplicity	1..1	



Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU C R4.2.2	

Parameter Name	DemJ1939ClearDTCsupport	
Label	J1939 Clear DTC Support	
Description	<p>This configuration switch defines whether J1939 Clear DTC is supported or not.</p> <p>This switches on and off the APIs Dem_J1939DcmClearDTC</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU C R4.2.2	

Parameter Name	DemJ1939FreezeFrameSupport	
Label	J1939 Freeze Frame Support	
Description	<p>This configuration switch defines whether J1939 freeze frames are supported or not.</p> <p>This switches on and off the following APIs:</p> <ul style="list-style-type: none"> ▶ Dem_J1939DcmSetFreezeFrameFilter ▶ Dem_J1939DcmGetNextFreezeFrame 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU C R4.2.2	

Parameter Name	DemJ1939Dm31Support	
Label	J1939 Dm31 Support	
Description	<p>This configuration switch defines whether J1939 DM31 is supported or not.</p> <p>This switches on and off the APIs</p>	



	<ul style="list-style-type: none"> ▶ Dem_J1939DcmFirstDTCwithLampStatus ▶ Dem_J1939DcmGetNextDTCwithLampStatus <p>The functionality related to this parameter is not supported by the current implementation.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU R4.2.2

Parameter Name	DemJ1939ExpandedFreezeFrameSupport	
Label	J1939 Expanded Freeze Frame Support	
Description	<p>This configuration switch defines whether J1939 expanded freeze frames are supported or not.</p> <p>This switches on and off the APIs</p> <ul style="list-style-type: none"> ▶ Dem_J1939DcmSetFreezeFrameFilter ▶ Dem_J1939DcmGetNextFreezeFrame ▶ Dem_J1939DcmGetNextSPNInFreezeFrame <p>The functionality related to this parameter is not supported by the current implementation.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU R4.2.2	

5.3.1.44. DemGroupOfDTC

Parameters included	
Parameter name	Multiplicity
DemGroupDTCs	1..1
Parameter Name	DemGroupDTCs



Label	Group DTC Value
Description	<p>DTC value of the selected group of DTC.</p> <p>(Range: 3 byte, 0x000000 is only available for the emission-related DTC group, 0xFFFFFFF is reserved for 'all DTCs', according to ISO14229-1 Annex D.1,)</p> <p>The DTC group 'all DTCs' is always available and will not be configured.</p>
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.3.1.45. DemIndicator

Parameters included	
Parameter name	Multiplicity
DemIndicatorID	1..1

Parameter Name	DemIndicatorID
Label	Indicator Id
Description	<p>Unique identifier of an indicator.</p> <p>The indicator Ids shall be sequentially ordered beginning with 0 and no gaps in between.</p> <p>The Handle-Id Wizard can be used to set this value automatically.</p>
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.3.1.46. DemNvRamBlockId

Parameters included	
Parameter name	Multiplicity

**Parameters included**

DemNvRamBlockIdRef	1..1
------------------------------------	------

Parameter Name	DemNvRamBlockIdRef	
Label	NVRAM Block Reference	
Description	<p>This parameter references to the NvmBlockDescriptor for NVRAM Blocks.</p> <p>Index of the NvmBlockDescriptor will be used by the DEM module for the identification of the NVM block</p>	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC V1.0.0	

5.3.1.47. DemOperationCycle**Parameters included**

Parameter name	Multiplicity
DemOperationCycleType	1..1
DemOperationCycleAutomaticEnd	1..1

Parameter Name	DemOperationCycleType
Label	Operation Cycle Type
Description	<p>Operation cycles types for the Dem to be supported by cycle-state APIs.</p> <p>Further cycle types can be specified as part of the Dem delivery.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_OPCYC_IGNITION: Ignition ON / OFF cycle. ▶ DEM_OPCYC_OBD_DCY: OBD Driving cycle. ▶ DEM_OPCYC_OTHER: further operation cycle. ▶ DEM_OPCYC_POWER: Power ON / OFF cycle. ▶ DEM_OPCYC_TIME: Time-based operation cycle. ▶ DEM_OPCYC_WARMUP: OBD Warm up cycle.



Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_OPCYC_POWER
Range	DEM_OPCYC_IGNITION DEM_OPCYC_OBD_DCY DEM_OPCYC_OTHER DEM_OPCYC_POWER DEM_OPCYC_TIME DEM_OPCYC_WARMUP
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemOperationCycleAutomaticEnd	
Label	Automatic ending	
Description	If this configuration parameter is set to TRUE, the dedicated operation cycle will be ended automatically at system shutdown.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.3.1.48. DemAgingCycle

5.3.1.49. DemRatiold

Parameters included	
Parameter name	Multiplicity
DemIUMPRGroup	1..1
DemRatioldType	1..1

**Parameters included**

DemDiagnosticEventRef	1..1
DemFunctionIdRef	1..1
DemSecondaryFunctionIdRef	0..n

Parameter Name	DemIUMPRGroup
Label	IUMPR Group
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This parameter specifies the assigned IUMPR group of the ratio Id.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_IUMPR_BOOSTPRS: Boost Pressure ▶ DEM_IUMPR_CAT1: Catalyst 1 ▶ DEM_IUMPR_CAT2: Catalyst 2 ▶ DEM_IUMPR_EGR: exhaust gas recirculation ▶ DEM_IUMPR_EGSENSOR: exhaust gas sensor ▶ DEM_IUMPR_EVAP: System Vapor Pressure ▶ DEM_IUMPR_NMHCCAT: non-methane hydrocarbons catalyst ▶ DEM_IUMPR_NOXADSORB: NOx adsorber ▶ DEM_IUMPR_NOXCAT: NOx catalyst ▶ DEM_IUMPR_OXS1: Oxygen sensor 1 ▶ DEM_IUMPR_OXS2: Oxygen sensor 2 ▶ DEM_IUMPR_PMFILTER: Particulate matter filter ▶ DEM_IUMPR_PRIVATE: ▶ DEM_IUMPR_SAIR: Secondary air system ▶ DEM_IUMPR_SECOXS1: secondary oxygen sensor 1 ▶ DEM_IUMPR_SECOXS2: secondary oxygen sensor 2
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_IUMPR_BOOSTPRS
Range	DEM_IUMPR_BOOSTPRS DEM_IUMPR_CAT1



	DEM_IUMPR_CAT2
	DEM_IUMPR_EGR
	DEM_IUMPR_EGSENSOR
	DEM_IUMPR_EVAP
	DEM_IUMPR_NMHCCAT
	DEM_IUMPR_NOXADSORB
	DEM_IUMPR_NOXCAT
	DEM_IUMPR_OXS1
	DEM_IUMPR_OXS2
	DEM_IUMPR_PMFILTER
	DEM_IUMPR_PRIVATE
	DEM_IUMPR_SAIR
	DEM_IUMPR_SECOXS1
	DEM_IUMPR_SECOXS2
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemRatioIdType
Label	Ratio Id Type
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This parameter defines whether the ratio Id will be calculated API or observer based.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ DEM_RATIO_API: API based ratio Id. ▶ DEM_RATIO_OBSERVER: Observer based ratio Id.
Multiplicity	1..1
Type	ENUMERATION
Default value	DEM_RATIO_API
Range	<ul style="list-style-type: none"> DEM_RATIO_API DEM_RATIO_OBSERVER
Configuration class	VariantPreCompile: VariantPreCompile



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

Parameter Name	DemDiagnosticEventRef
Label	Diagnostic Event
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This reference contains the link to a diagnostic event.</p>
Multiplicity	1..1
Type	REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemFunctionIdRef
Label	Primary Function Id
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This reference contains the link to a function identifier within the FiM which is used as a primary FID.</p>
Multiplicity	1..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemSecondaryFunctionIdRef
Label	Secondary Function Id
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This reference contains the link to a function identifier within the FiM which is used as a secondary FID.</p>
Multiplicity	0..n
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC



5.3.1.50. DemEventStatusPort

Parameters included	
Parameter name	Multiplicity
DemCSQueueLengthSetWIRStatusRteCalculated	1..1
DemCSQueueLengthSetWIRStatus	1..1

Parameter Name	DemCSQueueLengthSetWIRStatusRteCalculated	
Label	RTE calculated Queue Length for all operations of EventStatus Port	
Description	Configuration parameter that allows to set the queue length to all operations of the EventStatus Port to values calculated by RTE. This parameter is available only if DemRteUsage and DemUserControlledWIR-BitSupport parameters are enabled.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthSetWIRStatus	
Label	Dem_SetWIRStatus Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation</i> Dem_SetWIRStatus. This parameter is available only if DemRteUsage and DemUserControlledWIR-BitSupport parameters are enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	



5.3.1.51. DemOpCyclePort

Parameters included	
Parameter name	Multiplicity
DemCSQueueLengthSetOperationCycleStateRteCalculated	1..1
DemCSQueueLengthSetOperationCycleState	1..1
DemCSQueueLengthSetCycleQualified	1..1

Parameter Name	DemCSQueueLengthSetOperationCycleStateRteCalculated	
Label	RTE calculated Queue Length for the operation of DemOpCycle Port	
Description	Configuration parameter that allows to set the queue length to all operations of the OpCycle Port to values calculated by RTE. This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthSetOperationCycleState	
Label	Dem_SetOperationCycleState Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operations</i> Dem_SetOperationCycleState. This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthSetCycleQualified	
Label	Dem_SetCycleQualified Queue Length	



Description	Configuration parameter which allows defining the queue length of the <i>Client / Server ComSpec Operations</i> Dem_SetCycleQualified.	
	This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

5.3.1.52. DemAgingCyclePort

Parameters included	
Parameter name	Multiplicity
DemCSQueueLengthSetAgingCycleRteCalculated	1..1
DemCSQueueLengthSetAgingCycleState	1..1
DemCSQueueLengthSetAgingCycleCounterValue	1..1

Parameter Name	DemCSQueueLengthSetAgingCycleRteCalculated	
Label	RTE calculated Queue Length for all operations of EventInfo Port	
Description	Configuration parameter that allows to set the queue length to all operations of the AgingCycle Port to values calculated by RTE. This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthSetAgingCycleState	
Label	Dem_SetAgingCycleState Queue Length	



Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_SetAgingCycleState</i> . This parameter is available only if DemRteUsage is enabled and DemAgingCycleCounterProcessing is set to DEM_PROCESS_AGINGCTR_INTERN.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthSetAgingCycleCounterValue	
Label	Dem_SetAgingCycleCounterValue Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_SetAgingCycleCounterValue</i> . This parameter is available only if DemRteUsage is enabled and DemAgingCycleCounterProcessing is set to DEM_PROCESS_AGINGCTR_EXTERN.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

5.3.1.53. DemEnableCondPort

Parameters included	
Parameter name	Multiplicity
DemCSQueueLengthSetEnableConditionRteCalculated	1..1
DemCSQueueLengthSetEnableCondition	1..1



Parameter Name	DemCSQueueLengthSetEnableConditionRteCalculated	
Label	RTE calculated Queue Length for the operation of EnableCond Port	
Description	Configuration parameter that allows to set the queue length to all operations of the StorageCond Port to values calculated by RTE. This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthSetEnableCondition	
Label	Dem_SetEnableCondition Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation</i> Dem_SetEnableCondition. This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 _____ >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

5.3.1.54. DemStorageCondPort

Parameters included		
Parameter name	Multiplicity	
DemCSQueueLengthSetStorageConditionRteCalculated	1..1	
DemCSQueueLengthSetStorageCondition	1..1	

Parameter Name	DemCSQueueLengthSetStorageConditionRteCalculated
-----------------------	---



Label	RTE calculated Queue Length for the operation of StorageCond Port	
Description	<p>Configuration parameter that allows to set the queue length to all operations of the StorageCond Port to values calculated by RTE.</p> <p>This parameter is available only if DemRteUsage is enabled.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthSetStorageCondition	
Label	Dem_SetStorageCondition Queue Length	
Description	<p>Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation</i> Dem_SetStorageCondition.</p> <p>This parameter is available only if DemRteUsage is enabled.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<p><=65535</p> <p>>=1</p>	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

5.3.1.55. DemIndStatusPort

Parameters included		
Parameter name		Multiplicity
DemCSQueueLengthGetIndicatorStatusRteCalculated		1..1
DemCSQueueLengthGetIndicatorStatus		1..1

Parameter Name	DemCSQueueLengthGetIndicatorStatusRteCalculated
Label	RTE calculated Queue Length for the operation of ControlDTCSuppression Port



Description	Configuration parameter that allows to set the queue length to all operations of the IndStatus Port to values calculated by RTE. This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthGetIndicatorStatus	
Label	Dem_GetIndicatorStatus Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation</i> Dem_GetIndicatorStatus. This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

5.3.1.56. DemControlDTCSuppressionPort

Parameters included		
Parameter name		Multiplicity
DemCSQueueLengthSetDTCSuppressionRteCalculated		1..1
DemCSQueueLengthSetDTCSuppression		1..1

Parameter Name	DemCSQueueLengthSetDTCSuppressionRteCalculated
Label	RTE calculated Queue Length for the operation of ControlDTCSuppression Port
Description	Configuration parameter that allows to set the queue length to all operations of the ControlDTCSuppression Port to values calculated by RTE.



	This parameter is available only if DemRteUsage is enabled.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PreCompile: VariantPreCompile
Origin	EB

Parameter Name	DemCSQueueLengthSetDTCSuppression	
Label	Dem_SetDTCSuppression Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation</i> Dem_SetDTCSuppression. This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

5.3.1.57. DemCddIfPort

Parameters included	
Parameter name	Multiplicity
DemCSQueueLengthClearDTCRteCalculated	1..1
DemCSQueueLengthClearDTC	1..1

Parameter Name	DemCSQueueLengthClearDTCRteCalculated
Label	RTE calculated Queue Length for the operation of CddIf Port
Description	Configuration parameter that allows to set the queue length to all operations of the CddIf Port to values calculated by RTE. This parameter is available only if DemRteUsage is enabled and DemDefault-ASRServiceAPI is set to AUTOSAR_42 or AUTOSAR_43.



Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PreCompile: VariantPreCompile
Origin	EB

Parameter Name	DemCSQueueLengthClearDTC	
Label	Dem_ClearDTC Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_ClearDTC</i> . This parameter is available only if DemRteUsage is enabled and DemDefaultASRServiceAPI is set to AUTOSAR_42 or AUTOSAR_43.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

5.3.1.58. DemResetReadinessPort

Parameters included	
Parameter name	Multiplicity
DemCSQueueLengthResetReadinessRteCalculated	1..1
DemCSQueueLengthResetReadiness	1..1

Parameter Name	DemCSQueueLengthResetReadinessRteCalculated
Label	RTE calculated Queue Length for the operation of ResetReadiness Port
Description	Configuration parameter that allows to set the queue length to all operations of the ResetReadiness Port to values calculated by RTE. This parameter is available only if DemRteUsage and DemOBDSupport are enabled.



Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthResetReadiness	
Label	Dem_ResetReadiness Queue Length	
Description	<p>Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_ResetReadiness</i>.</p> <p>This parameter is available only if DemRteUsage and DemOBDSupport are enabled.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<p><=65535</p> <p>>=1</p>	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

5.3.1.59. DemPowerTakeOffStatusPort

Parameters included		
Parameter name		Multiplicity
DemCSQueueLengthSetPtoStatusRteCalculated		1..1
DemCSQueueLengthSetPtoStatus		1..1

Parameter Name	DemCSQueueLengthSetPtoStatusRteCalculated
Label	RTE calculated Queue Length for the operation of PowerTakeOffStatus Port
Description	<p>Configuration parameter that allows to set the queue length to all operations of the PowerTakeOffStatus Port to values calculated by RTE.</p> <p>This parameter is available only if DemRteUsage and DemOBDSupport are enabled.</p>



Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthSetPtoStatus	
Label	Dem_SetPtoStatus Queue Length	
Description	<p>Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_SetPtoStatus</i>.</p> <p>This parameter is available only if DemRteUsage and DemOBDSupport are enabled.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<p><=65535</p> <p>>=1</p>	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

5.3.1.60. DemEventPort

Parameters included	
Parameter name	Multiplicity
DemCSEventGroupRteCalculated	1..1
DemCSQueueLengthSetEventStatus	1..1
DemCSQueueLengthResetEventDebounceStatus	1..1
DemCSQueueLengthResetEventStatus	1..1
DemCSQueueLengthRestoreFreezeFrame	1..1
DemCSQueueLengthClearPrestoredFreezeFrame	1..1
DemCSQueueLengthSetEventDisabled	1..1

Parameter Name	DemCSEventGroupRteCalculated
-----------------------	--



Label	RTE calculated Queue Length for all operations of Event Port	
Description	Configuration parameter that allows to set the queue length to all operations of the EventInfo Port to values calculated by RTE. This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthSetEventStatus	
Label	Dem_SetEventStatus Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation</i> Dem_SetEventStatus. This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthResetEventDebounceStatus	
Label	Dem_ResetEventDebounceStatus Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation</i> Dem_ResetEventDebounceStatus. This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	



Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthResetEventStatus	
Label	Dem_ResetEventStatus Queue Length	
Description	<p>Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_ResetEventStatus</i>.</p> <p>This parameter is available only if DemRteUsage is enabled.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<p><=65535</p> <p>>=1</p>	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthPrestoreFreezeFrame	
Label	Dem_PrestoreFreezeFrame Queue Length	
Description	<p>Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_PrestoreFreezeFrame</i>.</p> <p>This parameter is available only if DemRteUsage is enabled.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<p><=65535</p> <p>>=1</p>	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthClearPrestoredFreezeFrame	
Label	Dem_ClearPrestoredFreezeFrame Queue Length	
Description	<p>Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_ClearPrestoredFreezeFrame</i>.</p>	



	This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthSetEventDisabled	
Label	Dem_SetEventDisabled Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation</i> Dem_SetEventDisabled. This parameter is available only if DemRteUsage and DemOBDSupport are enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

5.3.1.61. DemEventInfoPort

Parameters included		
Parameter name	Multiplicity	
DemCSEventInfoGroupRteCalculated	1..1	
DemCSQueueLengthGetEventStatus	1..1	
DemCSQueueLengthGetEventFailed	1..1	
DemCSQueueLengthGetEventTested	1..1	
DemCSQueueLengthGetDTCOfEvent	1..1	



Parameters included

DemCSQueueLengthGetFaultDetectionCounter	1..1
DemCSQueueLengthGetEventFreezeFrameData	1..1
DemCSQueueLengthGetEventExtendedDataRecord	1..1
DemCSQueueLengthGetEventFreezeFrameDataEx	1..1
DemCSQueueLengthGetEventExtendedDataRecordEx	1..1
DemCSQueueLengthGetDebouncingOfEvent	1..1

Parameter Name	DemCSEventInfoGroupRteCalculated	
Label	RTE calculated Queue Length for all operations of EventInfo Port	
Description	Configuration parameter that allows to set the queue length to all operations of the EventInfo Port to values calculated by RTE. This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthGetEventStatus	
Label	Dem_GetEventStatus Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_GetEventStatus</i> . This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthGetEventFailed	
Label	Dem_GetEventFailed Queue Length	



Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_GetEventFailed</i> . This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthGetEventTested	
Label	Dem_GetEventFailed Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_GetEventFailed</i> . This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthGetDTCOfEvent	
Label	Dem_GetDTCOfEvent Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_GetDTCOfEvent</i> . This parameter is available only if DemRteUsage is enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535	



	>=1
Configuration class	PreCompile: VariantPreCompile
Origin	EB

Parameter Name	DemCSQueueLengthGetFaultDetectionCounter	
Label	Dem_GetFaultDetectionCounter Queue Length	
Description	<p>Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_GetFaultDetectionCounter</i>.</p> <p>This parameter is available only if DemRteUsage is enabled.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<p><=65535</p> <hr/> <p>>=1</p>	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthGetEventFreezeFrameData	
Label	Dem_GetEventFreezeFrameData Queue Length	
Description	<p>Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_GetEventFreezeFrameData</i>.</p> <p>This parameter is available only if DemRteUsage is enabled and DemDefaultASRServiceAPI is not set to AUTOSAR_43.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<p><=65535</p> <hr/> <p>>=1</p>	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthGetEventExtendedDataRecord	
Label	Dem_GetEventExtendedDataRecord Queue Length	



Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_GetEventExtendedDataRecord</i> . This parameter is available only if DemRteUsage is enabled and DemDefaultASRServiceAPI is not set to AUTOSAR_43.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthGetEventFreezeFrameDataEx	
Label	Dem_GetEventFreezeFrameDataEx Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_GetEventFreezeFrameDataEx</i> . This parameter is available only if DemRteUsage is enabled and DemDefaultASRServiceAPI is set to AUTOSAR_43 or AUTOSAR_42 or DemEnableASR42ServiceAPI is enabled.	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthGetEventExtendedDataRecordEx	
Label	Dem_GetEventExtendedDataRecordEx Queue Length	
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_GetEventExtendedDataRecordEx</i> . This parameter is available only if DemRteUsage is enabled DemDefaultASRServiceAPI is set to AUTOSAR_42 or AUTOSAR_43 or DemEnableASR42ServiceAPI is enabled.	



Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthGetDebouncingOfEvent	
Label	Dem_GetDebouncingOfEvent Queue Length	
Description	<p>Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation</i> Dem_GetDebouncingOfEvent.</p> <p>This parameter is available only if DemRteUsage is enabled DemDefaultASRServiceAPI is set to AUTOSAR_42 or AUTOSAR_43 or DemEnableASR42ServiceAPI is enabled.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<=65535 >=1	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

5.3.1.62. DemPfcCycleQualifiedPort

Parameters included		
Parameter name	Multiplicity	
DemCSPfcCycleQualifiedGroupRteCalculated	1..1	
DemCSQueueLengthSetPfcCycleSetPfcCycleQualified	1..1	
DemCSQueueLengthGetPfcCycleQualified	1..1	

Parameter Name	DemCSPfcCycleQualifiedGroupRteCalculated
-----------------------	---



Label	RTE calculated Queue Length for all operations of PfcCycleQualified Port	
Description	<p>Configuration parameter that allows to set the queue length to all operations of the PfcCycleQualified Port to values calculated by RTE.</p> <p>This parameter is available only if DemRteUsage and DemOBDSupport are enabled.</p>	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthSetPfcCycleSetPfcCycleQualified	
Label	Dem_SetPfcCycleQualified Queue Length	
Description	<p>Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation</i> Dem_SetPfcCycleQualified.</p> <p>This parameter is available only if DemRteUsage and DemOBDSupport are enabled.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	1	
Range	<p><=65535</p> <p>>=1</p>	
Configuration class	PreCompile:	VariantPreCompile
Origin	EB	

Parameter Name	DemCSQueueLengthGetPfcCycleQualified	
Label	Dem_GetPfcCycleQualified Queue Length	
Description	<p>Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation</i> Dem_GetPfcCycleQualified.</p> <p>This parameter is available only if DemRteUsage and DemOBDSupport are enabled.</p>	
Multiplicity	1..1	
Type	INTEGER	



Default value	1
Range	<=65535 >=1
Configuration class	PreCompile: VariantPreCompile
Origin	EB

5.3.1.63. DemOverflowIndPort

Parameters included	
Parameter name	Multiplicity
DemCSOverflowIndicationGroupRteCalculated	1..1
DemCSQueueLengthGetEventMemoryOverflow	1..1
DemCSQueueLengthGetNumberOfEventMemoryEntries	1..1

Parameter Name	DemCSOverflowIndicationGroupRteCalculated
Label	RTE calculated Queue Length for all operations of OverflowIndication Port
Description	Configuration parameter that allows to set the queue length to all operations of the OverflowIndication Port to values calculated by RTE. This parameter is available only if DemRteUsage is enabled.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PreCompile: VariantPreCompile
Origin	EB

Parameter Name	DemCSQueueLengthGetEventMemoryOverflow
Label	Dem_GetEventMemoryOverflow Queue Length
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_GetEventMemoryOverflow</i> . This parameter is available only if DemRteUsage is enabled.
Multiplicity	1..1
Type	INTEGER



Default value	1
Range	<=65535 >=1
Configuration class	PreCompile: VariantPreCompile
Origin	EB

Parameter Name	DemCSQueueLengthGetNumberOfEventMemoryEntries
Label	Dem_GetNumberOfEventMemoryEntries Queue Length
Description	Configuration parameter that allows defining the queue length of the <i>Client / Server ComSpec Operation Dem_GetNumberOfEventMemoryEntries</i> . This parameter is available only if DemRteUsage is enabled.
Multiplicity	1..1
Type	INTEGER
Default value	1
Range	<=65535 >=1
Configuration class	PreCompile: VariantPreCompile
Origin	EB

5.3.1.64. DemStorageCondition

Parameters included	
Parameter name	Multiplicity
DemStorageConditionId	1..1
DemStorageConditionStatus	1..1

Parameter Name	DemStorageConditionId
Label	Storage Condition Id
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Defines a unique storage condition Id. This parameter should not be changeable by user, because the Id should be generated by Dem itself to prevent gaps and multiple use of an Id.</p>



	The storage condition Ids should be sequentially ordered beginning with 0 and no gaps in between.
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	DemStorageConditionStatus	
Label	Initial Storage Condition Status	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Defines the initial status for enable or disable of storage of a diagnostic event.</p> <p>The value is the initialization after power up (before this condition is reported the first time).</p> <ul style="list-style-type: none"> ▶ true: storage of a diagnostic event enabled ▶ false: storage of a diagnostic event disabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.3.1.65. DemStorageConditionGroup

Parameters included		
Parameter name		Multiplicity
DemStorageConditionRef		1..255

Parameter Name	DemStorageConditionRef
Label	Dem Storage Condition
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>References a storage condition.</p>



Multiplicity	1..255
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECU V1.0.0

5.3.1.66. DemDefensiveProgramming

Parameters included	
Parameter name	Multiplicity
DemDefProgEnabled	1..1
DemPrecondAssertEnabled	1..1
DemPostcondAssertEnabled	1..1
DemStaticAssertEnabled	1..1
DemUnreachAssertEnabled	1..1
DemInvariantAssertEnabled	1..1

Parameter Name	DemDefProgEnabled
Label	Enable Defensive Programming
Description	<p>Enables or disables the defensive programming feature for the module Dem.</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	DemPrecondAssertEnabled
Label	Enable Precondition Assertions



Description	Enables handling of precondition assertion checks reported from the module Dem. Dependency on parameter(s): <ul style="list-style-type: none">▶ Enable Development Error Detection (<code>DemDevErrorDetect</code>): must be enabled▶ Enable Defensive Programming (<code>DemDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemPostcondAssertEnabled
Label	Enable Postcondition Assertions
Description	Enables handling of postcondition assertion checks reported from the module Dem. Dependency on parameter(s): <ul style="list-style-type: none">▶ Enable Development Error Detection (<code>DemDevErrorDetect</code>): must be enabled▶ Enable Defensive Programming (<code>DemDefProgEnabled</code>): must be enabled
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	VariantPreCompile: VariantPreCompile
Origin	Elektrobit Automotive GmbH

Parameter Name	DemStaticAssertEnabled
Label	Enable Static Assertions
Description	Enables handling of static assertion checks reported from the module Dem. Dependency on parameter(s): <ul style="list-style-type: none">▶ Enable Development Error Detection (<code>DemDevErrorDetect</code>): must be enabled▶ Enable Defensive Programming (<code>DemDefProgEnabled</code>): must be enabled



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

Parameter Name	DemUnreachAssertEnabled	
Label	Enable Unreachable Code Assertions	
Description	<p>Enables handling of unreachable code assertion checks reported from the module Dem.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Enable Development Error Detection (<code>DemDevErrorDetect</code>): must be enabled ▶ Enable Defensive Programming (<code>DemDefProgEnabled</code>): must be enabled 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

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



5.3.1.67. PublishedInformation

Parameters included	
Parameter name	Multiplicity
PbcfgMSupport	1..1

Parameter Name	PbcfgMSupport
Label	PbcfgM support
Description	Specifies whether or not the Dem can use the PbcfgM module for post-build support.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

5.3.2. Recommended configurations

5.3.2.1. DemRecConfigurationDtcGroupsOpCycle

Containers included	
Container name	Container definition
DemGeneral	DemGeneral

Parameters included	
Parameter name	Value

5.3.2.1.1. DemGeneral

Containers included	
Container name	Container definition
DEM_DTC_GROUP_EMISSION_REL_DTCS	DemGroupOfDTC



Containers included

DEM_DTC_GROUP_POWERTRAIN_DTCS	DemGroupOfDTC
DEM_DTC_GROUP_CHASSIS_DTCS	DemGroupOfDTC
DEM_DTC_GROUP_BODY_DTCS	DemGroupOfDTC
DEM_DTC_GROUP_NETWORK_COM_DTCS	DemGroupOfDTC
DEM_OPCYC_POWER	DemOperationCycle

Parameters included

Parameter name	Value

5.3.2.1.2. DEM_DTC_GROUP_EMISSION_REL_DTCS

Parameters included

Parameter name	Value
DemGroupDTCs	0

5.3.2.1.3. DEM_DTC_GROUP_POWERTRAIN_DTCS

Parameters included

Parameter name	Value
DemGroupDTCs	2

5.3.2.1.4. DEM_DTC_GROUP_CHASSIS_DTCS

Parameters included

Parameter name	Value
DemGroupDTCs	4

5.3.2.1.5. DEM_DTC_GROUP_BODY_DTCS

Parameters included

Parameter name	Value
DemGroupDTCs	6



5.3.2.1.6. DEM_DTC_GROUP_NETWORK_COM_DTCS

Parameters included	
Parameter name	Value
DemGroupDTCs	8

5.3.2.1.7. DEM_OPCYC_POWER

Parameters included	
Parameter name	Value
DemOperationCycleType	DEM_OPCYC_POWER

5.3.3. Application programming interface (API)

5.3.3.1. Type definitions

5.3.3.1.1. Dem_DTCFormatType

Purpose	Selects/specifies the format of the DTC value.
Type	uint8

5.3.3.1.2. Dem_DTCGroupType

Purpose	Used to define the group of DTCs.
Type	uint32
Description	The user can add further groups. Unused bytes shall be filled with 00.

5.3.3.1.3. Dem_DTCKindType

Purpose	Used to define the DTC kind (type).
Type	uint8



5.3.3.1.4. Dem_DTCOriginType

Purpose	Enum used to define the location of the events/DTC.
Type	uint16
Description	The Dem_DTCOriginType is used to differ between the different event memories. The definition and use of the different memory types is OEM specific.

5.3.3.1.5. Dem_DTCRequestType

Purpose	Request type for Dem_GetDTCByOccurrenceTime() .
Type	uint8

5.3.3.1.6. Dem_DTCSeverityType

Purpose	Defines the type of a DTCSeverityMask according to ISO14229-1.
Type	uint8

5.3.3.1.7. Dem_DTCStatusMaskType

Purpose	Used to set the current status.
Type	uint8
Description	value: - 0x00 ... 0xFF: Match DTCStatusMask as defined in ISO14229-1

5.3.3.1.8. Dem_DTCTranslationFormatType

Purpose	DTC translation format as defined in ISO14229-1 Service 0x19 returned by Dem_GetTranslationType() .
Type	uint8

5.3.3.1.9. Dem_DataElementDataType

Purpose	Used to specify the data type used for External Data Elements.
Type	uint8



5.3.3.1.10. Dem_DataKindType

Purpose	Used to define the data kind (type).
Type	uint8

5.3.3.1.11. Dem_DebounceCounterStatusType

Purpose	Internal counter value for counter based debouncing algorithm.
Type	sint16

5.3.3.1.12. Dem_DebounceResetStatusType

Purpose	This type contains all monitor test result values, which can be reported via Dem_ResetEventDebounceStatus() .
Type	uint8
Description	bits 0x02..0xFF: reserved

5.3.3.1.13. Dem_DebouncingStateType

Purpose	Type for debouncing state.
Type	uint8

5.3.3.1.14. Dem_EntryDataType

Purpose	Entry data type, dynamic part of the error memory.
Type	uint8

5.3.3.1.15. Dem_EntryStatusType

Purpose	Type concerning immediate storage.	
Type	enum	
Constants	DEM_ENTRY_UNCHANGED	entry was not changed and needs not to be stored persistently



	DEM_ENTRY_CHANGED	entry was changed and has to be stored persistently during shutdown
	DEM_ENTRY_STORE_IMMEDIATELY	entry was changed and has to be stored persistent immediately
	DEM_ENTRY_CLEAR_IMMEDIATELY	entry was cleared and has to be stored persistent immediately

5.3.3.1.16. Dem_EvSignificanceType

Purpose	Type concerning the significance level per event which can be mapped as an external data.
Type	uint8

5.3.3.1.17. Dem_EventMemoryEntryType

Purpose	Event memory entry type, static part of the error memory.	
Type	struct	
Members	Dem_EventIdType EventId	Event Id of the entry
	Dem_EventIdType RootId	Root Id of the entry
	Dem_FFIIdxType OccCtr	Occurrence of event also used for NumFF
	Dem_OccOrderType OccOrder	Occurrence order of the event in the event memory
	uint8 SI30	DTC status indicators
	uint8 FDC12	Maximum fault detection counter since last clear
	uint8 OCC1	operation cycles since last unconfirmed-DTC counter
	uint8 OCC2	unconfirmedDTC aging counter
	uint8 OCC3	operation cycles since first unconfirmed-DTC counter
	uint8 OCC4	unconfirmedDTC operation cycle counter
	uint8 OCC5	number of WCUs cycles where event is reported as active counter
	uint8 OCC6	consecutive failed operation cycle counter



	uint8 OCC7	qualified/unconfirmedDTC operation cycle counter
	Dem_FailureCycleCounterType EventFailureCycleCounter	Failure cycle counter of Event
	uint8 CSLF	operation cycles since FDC last reached +127 counter
	uint8 CSFF	operation cycles since FDC first reached +127 counter
	Dem_SizeEntryDataType EntryData-aPos	Start of extended and freeze frame data
	Dem_EntryStatusType EntryStatus	Entry status for immediate storage Start of development freeze frame data
	uint8 DevFFEntryIdx	

5.3.3.1.18. Dem_EventStatusExtendedType

Purpose	In this data-type each bit has an individual meaning. The bit is set to 1 when the condition holds. For example, if the 2nd bit (0x02) is set to 1, this means that the test failed this operation cycle. If the bit is set to 0, it has not yet failed this cycle.
Type	uint8
Description	<p>Used in Dem_GetEventStatus() and Xxx_TriggerOnEventStatus().</p> <ul style="list-style-type: none"> ▶ Bit0: DEM_UDS_STATUS_TF ▶ Bit1: DEM_UDS_STATUS_TFTOC ▶ Bit2: DEM_UDS_STATUS_PDTC ▶ Bit3: DEM_UDS_STATUS_CDT ▶ Bit4: DEM_UDS_STATUS_TNCSLC ▶ Bit5: DEM_UDS_STATUS_TFSLC ▶ Bit6: DEM_UDS_STATUS_TNCTOC ▶ Bit7: DEM_UDS_STATUS_WIR

5.3.3.1.19. Dem_EventStatusType

Purpose	This type contains all monitor test result values, which can be reported via Dem_ReportErrorStatus() and Dem_SetEventStatus() .
---------	---



Type	uint8
Description	bits 0x04..0xFF: reserved

5.3.3.1.20. Dem_FFIIdxType

Purpose	Number of freeze frames type.
Type	uint8
Description	<p>Note: The size of this type is defined by ISO14229-1. (FF RecordNumber = 0x01 ... 0xFE)</p> <p>NumFF and MaxNumFF variables are also of this type.</p>

5.3.3.1.21. Dem_FailureCycleCounterType

Purpose	Type for value of Event failure cycles.
Type	uint8

5.3.3.1.22. Dem_GateEntryBaseFragmentType

Purpose	gate-entry base fragment structure. This type definition represents a "fragment-type" of the full type Dem_GateEntryBaseType . This type definition shall only be used locally to define full-types. This type definition shall not be used in source code as defining a full-type variable definition.	
Type	struct	
Members	Dem_EventIdType EventId	Event Id of the entry
	Dem_EventIdType RootId	Root Id of the entry
	Dem_FFIIdxType OccCtr	Occurrence of Event also used for NumFF
	Dem_OccOrderType OccOrder	Occurrence order of the event in the event memory
	uint8 SI30	DTC status indicators
	uint8 FDC12	Maximum fault detection counter since last clear
	uint8 OCC1	operation cycles since last unconfirmed-DTC counter
	uint8 OCC2	unconfirmedDTC aging counter



	uint8 OCC3	operation cycles since first unconfirmed-DTC counter
	uint8 OCC4	unconfirmedDTC operation cycle counter
	uint8 OCC5	number of WCUs cycles where event is reported as active counter
	uint8 OCC6	consecutive failed operation cycle counter
	uint8 OCC7	qualified/unconfirmedDTC operation cycle counter
	Dem_FailureCycleCounterType EventFailureCycleCounter	Failure cycle counter of Event
	uint8 CSLF	operation cycles since FDC last reached +127 counter
	uint8 CSFF	operation cycles since FDC first reached +127 counter
	boolean CDTCFlag	Confirmation status

5.3.3.1.23. Dem_GateEntryBaseType

Purpose	gate-entry base type - used for basic access	
Type	struct	
Members	Dem_GateEntryBaseFragmentType EntryBase	Basic information

5.3.3.1.24. Dem_GateEntryDataPtrVoidType

Purpose	gate-entry data void pointer type
Type	Dem_GateEntryDataVoidType *const

5.3.3.1.25. Dem_GateEntryDataVoidType

Purpose	gate-entry data void type - used for pointers to gate-entry data. This type definition represents a "fragment-type" of the full type Dem_GateEntryDataPtrVoidType. This type definition shall only be used locally to define full-types. This type definition shall not be used in source code as defining a full-type variable definition.
Type	void



5.3.3.1.26. Dem_GateEntryMirrorFragmentType

Purpose	gate-entry mirror data fragment structure. This type definition represents a "fragment-type" of the full type Dem_NvGateEntryMirrorDataType . This type definition shall only be used locally to define full-types. This type definition shall not be used in source code as defining a full-type variable definition.	
Type	struct	
Members	Dem_EntryDataType EntryData	Entry data for gate-entry consisting of Int-Val area, extended data , freeze frame data and J1939 Freeze frame StorageOrderId and DTC/event status value(s)

5.3.3.1.27. Dem_GateEntryOBDFPrimaryFragmentType

Purpose	gate-entry OBD freeze frame primary fragment structure. This type definition represents a "fragment-type" of the full type Dem_NvGateEntryPrimaryDataType . This type definition shall only be used locally to define full-types. This type definition shall not be used in source code as defining a full-type variable definition.	
Type	struct	
Members	Dem_MultipleOBDEventMemoryEntryType OBDF	OBD Freeze Frame entry structure

5.3.3.1.28. Dem_GateEntryPrimaryFragmentType

Purpose	gate-entry primary data fragment structure. This type definition represents a "fragment-type" of the full type Dem_NvGateEntryPrimaryDataType . This type definition shall only be used locally to define full-types. This type definition shall not be used in source code as defining a full-type variable definition.	
Type	struct	
Members	Dem_EntryDataType EntryData	Entry data for gate-entry consisting of Int-Val area, extended data , freeze frame data and J1939 Freeze frame StorageOrderId and DTC/event status value(s)

5.3.3.1.29. Dem_GateEntrySecondaryFragmentType

Purpose	gate-entry secondary data fragment structure. This type definition represents a "fragment-type" of the full type Dem_NvGateEntrySecondaryDataType . This type definition
----------------	--



	shall only be used locally to define full-types. This type definition shall not be used in source code as defining a full-type variable definition.	
Type	struct	
Members	Dem_EntryDataType EntryData	Entry data for gate-entry consisting of Int-Val area, extended data , freeze frame data and J1939 Freeze frame StorageOrderId and DTC/event status value(s)

5.3.3.1.30. Dem_IndicatorCycleCounterType

Purpose	Indicator cycle counter Type.
Type	uint8

5.3.3.1.31. Dem_IndicatorIdType

Purpose	Used to request a indicator type.
Type	uint8
Description	value: indicator lamp ID

5.3.3.1.32. Dem_IndicatorStatusType

Purpose	Used to return the status of Dem_GetIndicatorStatus() .
Type	uint8

5.3.3.1.33. Dem_InitMonitorReasonType

Purpose	(Re-)Initialization reason returned by the callback <Module>_DemInitMonitorFor<EventName>()
Type	uint8

5.3.3.1.34. Dem_J1939DcmDTCStatusFilterType

Purpose	The type to distinguish which DTCs should be filtered.
----------------	--



Type	uint8
------	-------

5.3.3.1.35. Dem_J1939DcmDiagnosticReadiness1Type

Purpose	This structure represents all data elements of the DM05 message.	
Type	struct	
Members	uint16 NonContinuouslyMonitoredSystemsSupport	Identifies the noncontinuously monitored systems support
	uint16 NonContinuouslyMonitoredSystemsStatus	Identifies the noncontinuously monitored systems status
	uint8 ActiveTroubleCodes	Number of active DTCs
	uint8 PreviouslyActiveDiagnosticTroubleCodes	Number of previously active DTCs
	uint8 OBDCompliance	OBD Compliance
	uint8 ContinuouslyMonitoredSystemsSupport_Status	Identifies the continuously monitored system support and status

5.3.3.1.36. Dem_J1939DcmDiagnosticReadiness2Type

Purpose	This structure represents all data elements of the DM05 message.	
Type	struct	
Members	uint16 DistanceTraveledWhileMILisActivated	The kilometers accumulated while the MIL is activated
	uint16 DistanceSinceDTCsCleared	Distance accumulated since emission related DTCs were cleared
	uint16 MinutesRunbyEngineWhileMILisActivated	Accumulated count (in minutes) while the MIL is activated (on)
	uint16 TimeSinceDiagnosticTroubleCodesCleared	Engine running time accumulated since emission related DTCs were cleared

5.3.3.1.37. Dem_J1939DcmDiagnosticReadiness3Type

Purpose	This structure represents all data elements of the DM05 message.	
Type	struct	



Members	uint16 TimeSinceEngineStart	Time since key-on that the engine has been running
	uint16 NonContinuouslyMonitoredSystemsEnableStatus	Enable status of noncontinuous monitors this monitoring cycle
	uint16 NonContinuouslyMonitoredSystems	Completion status of noncontinuous monitors this monitoring cycle
	uint8 NumberOfWarmupsSinceDTCsCleared	Number of OBD warmup cycles since all DTCs were cleared
	uint8 ContinuouslyMonitoredSystemsEnableCompletedStatus	Identifies the continuously monitored system enable/completed support and status

5.3.3.1.38. Dem_J1939DcmLampStatusType

Purpose	The J1939 status of the 4 lamps MIL, RSI, AWL and protect.
Type	uint16

5.3.3.1.39. Dem_J1939DcmSetClearFilterType

Purpose	The type to distinguish which DTCs gets cleared.
Type	uint8

5.3.3.1.40. Dem_J1939DcmSetFreezeFrameFilterType

Purpose	The type to distinguish the J1939 freeze frame kinds.
Type	uint8

5.3.3.1.41. Dem_MultipleOBDEventMemoryEntryType

Purpose	Multiple Event memory entry type, static part of the error memory.	
Type	struct	
Members	Dem_OccOrderType OBDOccOrder	occurrence order of the OBD event in the primary event memory
	Dem_EntryDataType OBDFFData	size in bytes of all data elements contained in the OBD FF



5.3.3.1.42. Dem_NvDataType

Purpose	Structure containing all non-volatile data.	
Type	struct	
Members	uint32 NvRamBlockSignature	Signature value of the configuration of the non-volatile data. This value is compared with the signature value of the configuration of the execution code stored in the macro DEM_CONFIG_NVRAM_SIGNATURE
	Dem_DTCStatusMaskType DTCStatusMask	Status of each DTC. In case Nv storage of TF bit is disabled TF bit 0 shall not be used
	uint8 EvMemOvfIndFlags	Bit-field to store the overflow indication status of all event memorys (primary, secondary, mirror, permanent). One bit will be used per memory.
	uint8 PfcStatus	Permanent fault code
	Dem_IndicatorCycleCounterType IndicatorCycleCounter	Indicator cycle counter
	uint8 IndicatorCycleCounterRole	Flags to indicate indicator cycle counter role
	uint8 IndicatorCycleCounterWIRStatus	Flags to indicate indicator cycle counter WIR status
	Dem_OperationCycleStateType OpCycleState	Operation Cycle states
	Dem_DebounceCounterStatusType DebounceCounterStatus	Debouncing info for persistently stored counter debounced events
	Dem_EventMemoryEntryType EventMemoryPrimary	Primary event memory
	Dem_EventMemoryEntryType EventMemorySecondary	Secondary event memory
	Dem_EventMemoryEntryType EventMemoryMirror	Mirror event memory
	Dem_EntryDataType EntryDataPrimary	Primary entry data area
	Dem_SizeEntryDataType EntryDataFreePosPrimary	Primary entry data free position



<code>Dem_EntryDataType EntryDataSecondary</code>	Secondary entry data area
<code>Dem_SizeEntryDataType EntryDataFreePosSecondary</code>	Secondary entry data free position
<code>Dem_EntryDataType EntryDataMirror</code>	Mirror entry data area
<code>Dem_SizeEntryDataType EntryDataFreePosMirror</code>	Mirror entry data free position
<code>uint8 StorageOrderId</code>	Id to determine the most recent DTC status values
<code>Dem_SingleOBDEventMemoryEntryType SingleOBDEventMemoryEntry</code>	
<code>uint32 TicksTraveledWhileMILIsActivated</code>	distance traveled while MIL is activated
<code>uint8 NrWUCsWhileMILInactive</code>	number of warm-up cycles that happened since the MIL was deactivated
<code>Dem_IndicatorStatusType MILPrevStatusOnlyOnOffRelevance</code>	previous status of the MIL
<code>uint32 TicksTraveledSinceDTCsCleared</code>	Distance traveled since all DTCs were cleared (used by PID \$31)
<code>uint8 OBDWarmupCycleCtr</code>	Number of OBD warm-up cycles since all DTCs were cleared
<code>Dem_ReadinessEnableStatusType ReadinessEnableStatus</code>	Readiness status of the Readiness Groups
<code>Dem_EventIdType DevFFAEventId</code>	EventIds of stored Development Freeze Frames A
<code>uint8 DevFFAData</code>	Data of stored Development Freeze Frames A
<code>Dem_EventIdType DevFFBEventId</code>	EventIds of stored Development Freeze Frames B
<code>uint8 DevFFBData</code>	Data of stored Development Freeze Frames B
<code>Dem_FailureCycleCounterType CmbEvFailureCycleCounter</code>	FailureCycle counters for all combined events
<code>Dem_StatusCounterType StatusCounter</code>	Failure/aging counters for events



	<code>Dem_EventIdType OccurKindEventId</code>	Stored EventIds of DTC by Occurrence-Time
	<code>boolean Dem_OpCycleDCYIsQualified</code>	qualification status of driving cycle
Description	<p>If no immediate storage is used, NvData consists of DTCStatusMask and event memory entries. If immediate storage is used, NvData consists of DTCStatusMask, but no event memory entries. The event memory data will be stored separately via gate-entry if immediate storage is enabled for any event.</p> <p>Immediate storage = OFF:</p> <ul style="list-style-type: none"> ▶ NvM block DEM_NVM_BLOCK_ID_DEFAULT stores all non-volatile data <p>Immediate storage = ON:</p> <ul style="list-style-type: none"> ▶ NvM block DEM_NVM_BLOCK_ID_PRIMARY stores DTCStatusMask ▶ NvM block DEM_NVM_BLOCK_ID_SECONDARY stores entry-related data of primary memory ▶ NvM block DEM_NVM_BLOCK_ID_MIRROR stores entry-related data of secondary memory ▶ NvM block DEM_NVM_BLOCK_ID_PERMANENT stores entry-related data of mirror memory ▶ NvM block DEM_NVM_BLOCK_ID_PERMANENT stores entry-related data of permanent memory 	

5.3.3.1.43. `Dem_NvGateEntryDataPtrVoidType`

Purpose	gate-entry void pointer type
Type	Dem_NvGateEntryDataVoidType *const

5.3.3.1.44. `Dem_NvGateEntryDataVoidType`

Purpose	gate-entry void type - used for pointers to gate entries
Type	void

5.3.3.1.45. `Dem_NvGateEntryMirrorDataType`

Purpose	Structure containing all non-volatile mirror event memory data for one gate-entry.
----------------	--



Type	struct	
Members	Dem_GateEntryBaseFragmentType EntryBase	Basic information
	Dem_GateEntryMirrorFragmentType EntryMirror	Mirror entry data

5.3.3.1.46. Dem_NvGateEntryPrimaryDataType

Purpose	Structure containing all non-volatile primary event memory data for one gate-entry.	
Type	struct	
Members	Dem_GateEntryBaseFragmentType EntryBase	Basic information
	Dem_GateEntryPrimaryFragment- Type EntryPrimary	Primary entry data
	Dem_GateEntryOBDFPrimaryFrag- mentType OBDFEntry	OBDF information

5.3.3.1.47. Dem_NvGateEntrySecondaryDataType

Purpose	Structure containing all non-volatile secondary event memory data for one gate-entry.	
Type	struct	
Members	Dem_GateEntryBaseFragmentType EntryBase	Basic information
	Dem_GateEntrySecondaryFragment- Type EntrySecondary	Secondary entry data

5.3.3.1.48. Dem_OccOrderType

Purpose	Type concerning occurrence order which is used for event displacement.
Type	uint16

5.3.3.1.49. Dem_OperationCycleIdType

Purpose	Used to select the individual operation cycle in the API Dem_SetOperationCycleState() .
---------	---



Type	uint8
Description	value: The OperationCycleId, e.g. the ignition cycle is assigned to an ID

5.3.3.1.50. Dem_OperationCycleStateType

Purpose	This type contains operation cycle state values, which can be reported via Dem_SetOperationCycleState() .
Type	uint8

5.3.3.1.51. Dem_RatioidType

Purpose	OBD specific ratio Id (related to a specific event, a FID, and an IUMPR group). This type depends on the Dem configuration.
Type	uint8
Description	Note: The size of this type is implementation specific and can be uint8 or uint16.

5.3.3.1.52. Dem_ReadinessEnableStatusType

Purpose	Data type for state of the readiness group.
Type	uint32

5.3.3.1.53. Dem_ReturnClearDTCType

Purpose	Used to return the status of Dem_ClearDTC() .
Type	uint8

5.3.3.1.54. Dem_ReturnGetNextFilteredElementType

Purpose	Used to return the status of the Dem_J1939DcmGetNextFilteredDTC<...> interfaces.
Type	uint8



5.3.3.1.55. Dem_ReturnNumberOfFilteredDTCType

Purpose	Used to return the status of Dem_GetNumberOfFilteredDTC() .
Type	uint8

5.3.3.1.56. Dem_ReturnSetFilterType

Purpose	Used to return the status of (re-)setting a specific filter.
Type	uint8

5.3.3.1.57. Dem_SRDataElementEndiannessType

Purpose	Used to specify the endianness used for SR interfaces.
Type	uint8

5.3.3.1.58. Dem_SingleOBDEventMemoryEntryType

Purpose	Single Event memory entry type, static part of the error memory.	
Type	struct	
Members	Dem_EventIdType EventId	event Id of the entry
	Dem_EntryDataType OBDFData	

5.3.3.1.59. Dem_SizeEntryDataType

Purpose	Entry data size type.
Type	uint16

5.3.3.1.60. Dem_StatusCounterType

Purpose	Type for value of Status counters.
Type	uint8



5.3.3.1.61. Dem_UdsStatusByteType

Purpose	Used to set the current status.
Type	uint8
Description	<p>Lower limit 0x00 Elements Lower limit 0xFF Elements</p> <p>In this data-type each bit has an individual meaning. The bit is set to 1 when the condition holds. For example, if the 2nd bit (0x02) is set to 1, this means that the test failed this operation cycle. If the bit is set to 0, it has not yet failed this cycle. Used in Dem_GetDTCStatusAvailabilityMask().</p> <ul style="list-style-type: none">▶ Bit0: DEM_UDS_STATUS_TF▶ Bit1: DEM_UDS_STATUS_TFTOC▶ Bit2: DEM_UDS_STATUS_PDTC▶ Bit3: DEM_UDS_STATUS_CDTC▶ Bit4: DEM_UDS_STATUS_TNCSLC▶ Bit5: DEM_UDS_STATUS_TFSLC▶ Bit6: DEM_UDS_STATUS_TNCTOC▶ Bit7: DEM_UDS_STATUS_WIR

5.3.3.2. Macro constants

5.3.3.2.1. DEM_AR_RELEASE_MAJOR_VERSION

Purpose	AUTOSAR release major version.
Value	4U

5.3.3.2.2. DEM_AR_RELEASE_MINOR_VERSION

Purpose	AUTOSAR release minor version.
Value	0U

5.3.3.2.3. DEM_AR_RELEASE_REVISION_VERSION

Purpose	AUTOSAR release revision version.
----------------	-----------------------------------



Value	3U
--------------	----

5.3.3.2.4. DEM_ASR42_CLEAR_BUSY

Purpose	Another client is currently clearing DTCs. The requested operation will not be started.
Value	0x05U

5.3.3.2.5. DEM_ASR42_CLEAR_FAILED

Purpose	DTC clearing failed.
Value	0x03U

5.3.3.2.6. DEM_ASR42_CLEAR_MEMORY_ERROR

Purpose	An error occurred during erasing a memory location.
Value	0x06U

5.3.3.2.7. DEM_BIG_ENDIAN

Purpose	big endian endianness type used for SR interfaces
Value	0x00U

5.3.3.2.8. DEM_BUFFER_TOO_SMALL

Purpose	The provided buffer size is too small.
Value	0x15U

5.3.3.2.9. DEM_BUSY

Purpose	Another Dem_SelectDTC or Dem_SelectDTC dependent operation of this client is currently in progress.
----------------	---



Value	0x16U
--------------	-------

5.3.3.2.10. DEM_CLEAR_BUSY

Purpose	Another client is currently clearing DTCs. The requested operation will not be started.
Value	0x05U

5.3.3.2.11. DEM_CLEAR_FAILED

Purpose	DTC not cleared.
Value	0x07U

5.3.3.2.12. DEM_CLEAR_MEMORY_ERROR

Purpose	An error occurred during erasing a memory location.
Value	0x06U

5.3.3.2.13. DEM_CLEAR_OK

Purpose	DTC successfully cleared.
Value	0x00U

5.3.3.2.14. DEM_CLEAR_PENDING

Purpose	Clearing of DTC is pending.
Value	0x04U

5.3.3.2.15. DEM_CLEAR_WRONG_DTC

Purpose	DTC value not existing (in this format).
Value	0x01U



5.3.3.2.16. DEM_CLEAR_WRONG_DTCORIGIN

Purpose	Wrong DTC origin.
Value	0x02U

5.3.3.2.17. DEM_CYCLE_STATE_END

Purpose	End of operation cycle.
Value	0x01U

5.3.3.2.18. DEM_CYCLE_STATE_START

Purpose	Start/restart the operation cycle.
Value	0x00U

5.3.3.2.19. DEM_DATAKIND_EXTENDEDDATA

Purpose	Select Extended Data kind.
Value	0x01U

5.3.3.2.20. DEM_DATAKIND_FREEZEFRAME

Purpose	Select Freeze Frame data kind.
Value	0x00U

5.3.3.2.21. DEM_DEBOUNCE_STATUS_FREEZE

Purpose	Freeze status value for event debouncing.
Value	0x00U

5.3.3.2.22. DEM_DEBOUNCE_STATUS_RESET

Purpose	Reset status value for event debouncing.
----------------	--



Value	0x01U
--------------	-------

5.3.3.2.23. DEM_DTC_FORMAT_J1939

Purpose	selects the merged SPN + FMI to 3-byte J1939 DTC format
Value	0x02U

5.3.3.2.24. DEM_DTC_FORMAT_OBD

Purpose	selects the 2-byte OBD DTC format (refer to configuration parameter DemObdDTC)
Value	0x00U

5.3.3.2.25. DEM_DTC_FORMAT_UDS

Purpose	selects the 3-byte UDS DTC format (refer to configuration parameter DemUdsDTC)
Value	0x01U

5.3.3.2.26. DEM_DTC_GROUP_ALL_DTCS

Purpose	All DTCs.
Value	0xFFFFFFFU

5.3.3.2.27. DEM_DTC_GROUP_EMISSION_REL_DTCS

Purpose	Emission related DTCs.
Value	0x0000000U

5.3.3.2.28. DEM_DTC_KIND_ALL_DTCS

Purpose	Select all DTCs.
Value	0x01U



5.3.3.2.29. DEM_DTC_KIND_EMISSION_REL_DTCS

Purpose	Select OBD-relevant DTCs.
Value	0x02U

5.3.3.2.30. DEM_DTC_ORIGIN_MIRROR_MEMORY

Purpose	Event information located in the mirror memory.
Value	0x02U

5.3.3.2.31. DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY

Purpose	Event information located in the OBD relevant memory.
Value	0x04U

5.3.3.2.32. DEM_DTC_ORIGIN_PERMANENT_MEMORY

Purpose	Event information located in the permanent memory.
Value	0x03U

5.3.3.2.33. DEM_DTC_ORIGIN_PRIMARY_MEMORY

Purpose	Event information located in the primary memory.
Value	0x01U

5.3.3.2.34. DEM_DTC_ORIGIN_SECONDARY_MEMORY

Purpose	Event information located in the secondary memory.
Value	0x04U

5.3.3.2.35. DEM_DTC_TRANSLATION_ISO11992_4

Purpose	DTCs of ISO11992_4.
----------------	---------------------



Value	0x03U
--------------	-------

5.3.3.2.36. DEM_DTC_TRANSLATION_ISO14229_1

Purpose	DTCs of ISO14229-1.
Value	0x01U

5.3.3.2.37. DEM_DTC_TRANSLATION_ISO15031_6

Purpose	DTCs of ISO15031-6.
Value	0x00U

5.3.3.2.38. DEM_DTC_TRANSLATION_J2012DA_FORMAT_04

Purpose	DTCs of SAE_J2012-DA_DTCFormat_04.
Value	0x04U

5.3.3.2.39. DEM_DTC_TRANSLATION_SAEJ1939_73

Purpose	DTCs of SAEJ1939-73.
Value	0x02U

5.3.3.2.40. DEM_EVENT_ID_IGNORE

Purpose	Symbolic name of event ID that must be ignored during processing of the queue since it got invalidated.
Value	0x1FFFU

5.3.3.2.41. DEM_EVENT_ID_INVALID

Purpose	Symbolic name of invalid event ID.
Value	0U



5.3.3.2.42. DEM_EVENT_SIGNIFICANCE_FAULT

Purpose	Offset of DTC value in 1st element of event desc.
Value	1U

5.3.3.2.43. DEM_EVENT_SIGNIFICANCE_OCCURRENCE

Purpose	Offset of DTC value in 1st element of event desc.
Value	0U

5.3.3.2.44. DEM_EVENT_STATUS_FAILED

Purpose	Monitor reports qualified test result failed.
Value	0x01U

5.3.3.2.45. DEM_EVENT_STATUS_PASSED

Purpose	Monitor reports qualified test result passed.
Value	0x00U

5.3.3.2.46. DEM_EVENT_STATUS_PREFAILED

Purpose	Monitor reports non-qualified test result pre-failed (debounced Dem-internally).
Value	0x03U

5.3.3.2.47. DEM_EVENT_STATUS_PREPASSED

Purpose	Monitor reports non-qualified test result pre-passed (debounced Dem-internally).
Value	0x02U

5.3.3.2.48. DEM_E_INIT_FAILED

Purpose	DET error code for AR 4.3 in case Dem initialization fails.
---------	---



Value	0x14U
--------------	-------

5.3.3.2.49. DEM_E_INTEGRATION

Purpose	DET error code for unfulfilled return values from an SW-C.
Value	0x41U

5.3.3.2.50. DEM_E_INVALID_OBDMID

Purpose	DET error code for AR 4.3 in case Dem_DcmGetAvailableOBDMIDs is called with invalid OBDMID.
Value	0x50U

5.3.3.2.51. DEM_E_NODATAAVAILABLE

Purpose	The requested event data is not currently stored (but the request was valid).
Value	0x30U

5.3.3.2.52. DEM_E_NO_DTC_AVAILABLE

Purpose	There is no DTC configured in the requested format.
Value	0xAU
Description	Dem-specific Std_ReturnType value used by Dem_GetDTCOfEvent()

5.3.3.2.53. DEM_E_NO_FDC_AVAILABLE

Purpose	There is no fault detection counter available for the requested event.
Value	0xEU

5.3.3.2.54. DEM_E_PARAM_CONFIG

Purpose	DET error code for invalid configuration.
----------------	---



Value	0x10U
--------------	-------

5.3.3.2.55. DEM_E_PARAM_DATA

Purpose	DET error code for invalid data.
Value	0x12U

5.3.3.2.56. DEM_E_PARAM_LENGTH

Purpose	DET error code for invalid length.
Value	0x13U

5.3.3.2.57. DEM_E_PARAM_POINTER

Purpose	DET error code for invalid address.
Value	0x11U

5.3.3.2.58. DEM_E_UNINIT

Purpose	DET error code for wrong initialization status.
Value	0x20U

5.3.3.2.59. DEM_E_WRONG_BUFFERSIZE

Purpose	The provided buffer size is too small.
Value	0x15U

5.3.3.2.60. DEM_E_WRONG_CONDITION

Purpose	DET error code for unfulfilled pre-conditions for an API call.
Value	0x40U



5.3.3.2.61. DEM_E_WRONG_CONFIGURATION

Purpose	DET error code for AR 4.3 in case an API function is called with a parameter value, which is not allowed by active configuration.
Value	0x10U

5.3.3.2.62. DEM_E_WRONG_CONTEXT

Purpose	DET error code for wrong call-context in multi-core case.
Value	0x42U

5.3.3.2.63. DEM_E_WRONG_DIDNUMBER

Purpose	The requested DID is not supported by the freeze frame.
Value	0x32U

5.3.3.2.64. DEM_E_WRONG_RECORDNUMBER

Purpose	The requested record number is not supported by the event.
Value	0x31U

5.3.3.2.65. DEM_FILTERED_BUFFER_TOO_SMALL

Purpose	Buffer in the BufSize parameter is not huge enough.
Value	0x03U

5.3.3.2.66. DEM_FILTERED_NO_MATCHING_ELEMENT

Purpose	No further element (matching the filter criteria) found.
Value	0x01U

5.3.3.2.67. DEM_FILTERED_OK

Purpose	Returned next filtered element.
----------------	---------------------------------



Value	0x00U
--------------	-------

5.3.3.2.68. DEM_FILTERED_PENDING

Purpose	The requested value is calculated asynchronously and currently not available. The caller can retry later. Only used by asynchronous interfaces.
Value	0x02U

5.3.3.2.69. DEM_FILTER_ACCEPTED

Purpose	Filter was accepted.
Value	0x00U

5.3.3.2.70. DEM_FIRST_DET_CONFIRMED_DTC

Purpose	First detected confirmed DTC requested.
Value	0x03U

5.3.3.2.71. DEM_FIRST_FAILED_DTC

Purpose	First failed DTC requested.
Value	0x01U

5.3.3.2.72. DEM_INDICATOR_BLINKING

Purpose	Indicator blinking mode.
Value	0x02U

5.3.3.2.73. DEM_INDICATOR_BLINK_CONT

Purpose	Indicator blinking or continuously on mode. The SW-C for indicator is responsible to decide if the indicator is blinking or continuously on.
Value	0x03U



5.3.3.2.74. DEM_INDICATOR_CONTINUOUS

Purpose	Indicator continuously on mode.
Value	0x01U

5.3.3.2.75. DEM_INDICATOR_FAST_FLASH

Purpose	Indicator fast flash mode.
Value	0x05U

5.3.3.2.76. DEM_INDICATOR_OFF

Purpose	Indicator off mode.
Value	0x00U

5.3.3.2.77. DEM_INDICATOR_SLOW_FLASH

Purpose	Indicator slow flash mode.
Value	0x04U

5.3.3.2.78. DEM_INIT_MONITOR_CLEAR

Purpose	Event was cleared and all internal values and states are reset.
Value	0x01U

5.3.3.2.79. DEM_INIT_MONITOR_REENABLED

Purpose	Enable condition or DTC setting re-enabled.
Value	0x03U

5.3.3.2.80. DEM_INIT_MONITOR_RESTART

Purpose	Operation cycle of the event was (re-)started.
----------------	--



Value	0x02U
--------------	-------

5.3.3.2.81. DEM_INIT_MONITOR_STORAGE_REENABLED

Purpose	Storage condition re-enabled.
Value	0x04U

5.3.3.2.82. DEM_J1939AWIL_ACTIVE

Purpose	Amber warning indicator lamp active.
Value	0x0400U

5.3.3.2.83. DEM_J1939AWIL_CONT_UNAVAILABLE

Purpose	Amber warning indicator lamp is continuously lit or unavailable.
Value	0x000CU

5.3.3.2.84. DEM_J1939AWIL_FAST_FLASH

Purpose	Amber warning indicator lamp is fast flashing.
Value	0x0004U

5.3.3.2.85. DEM_J1939DCM_EXPANDED_FREEZEFRAME

Purpose	ExpandedFreezeFrame(DM25).
Value	0x01U

5.3.3.2.86. DEM_J1939DCM_FREEZEFRAME

Purpose	FreezeFrame (DM04).
Value	0x00U



5.3.3.2.87. DEM_J1939DCM_SPNS_IN_EXPANDED_FREEZEFRAME

Purpose	SPNs in Expanded-FreezeFrame (DM24).
Value	0x02U

5.3.3.2.88. DEM_J1939DTC_ACTIVE

Purpose	Active DTCs filter.
Value	0x00U

5.3.3.2.89. DEM_J1939DTC_CLEAR_ALL

Purpose	active DTCs to be cleared
Value	0x00U

5.3.3.2.90. DEM_J1939DTC_CLEAR_PREVIOUSLY_ACTIVE

Purpose	previously active DTCs to be cleared
Value	0x01U

5.3.3.2.91. DEM_J1939DTC_CURRENTLY_ACTIVE

Purpose	Currently active DTCs filter.
Value	0x04U

5.3.3.2.92. DEM_J1939DTC_PENDING

Purpose	Pending DTCs filter.
Value	0x02U

5.3.3.2.93. DEM_J1939DTC_PERMANENT

Purpose	Permanent DTCs filter.
----------------	------------------------



Value	0x03U
--------------	-------

5.3.3.2.94. DEM_J1939DTC_PREVIOUSLY_ACTIVE

Purpose	Previously active DTCs filter.
Value	0x01U

5.3.3.2.95. DEM_J1939MIL_ACTIVE

Purpose	MIL is active.
Value	0x4000U

5.3.3.2.96. DEM_J1939MIL_CONT_UNAVAILABLE

Purpose	MIL is continuously lit or unavailable.
Value	0x00C0U

5.3.3.2.97. DEM_J1939MIL_FAST_FLASH

Purpose	MIL is fast flashing.
Value	0x0040U

5.3.3.2.98. DEM_J1939PIL_ACTIVE

Purpose	Protection indicator lamp active.
Value	0x0100U

5.3.3.2.99. DEM_J1939PIL_CONT_UNAVAILABLE

Purpose	Protection indicator lamp is continuously lit or unavailable.
Value	0x0003U



5.3.3.2.100. DEM_J1939PIL_FAST_FLASH

Purpose	Protection indicator lamp is fast flashing.
Value	0x0001U

5.3.3.2.101. DEM_J1939RSIL_ACTIVE

Purpose	Red stop indicator lamp active.
Value	0x1000U

5.3.3.2.102. DEM_J1939RSIL_CONT_UNAVAILABLE

Purpose	Red stop indicator lamp is continuously lit or unavailable.
Value	0x0030U

5.3.3.2.103. DEM_J1939RSIL_FAST_FLASH

Purpose	Red stop indicator lamp is fast flashing.
Value	0x0010U

5.3.3.2.104. DEM_LITTLE_ENDIAN

Purpose	little endian endianness type used for SR interfaces
Value	0x01U

5.3.3.2.105. DEM_MAXNUM_OCCURRENCE_KINDS

Purpose	Size of array used to store the EventIds corresponding to the occurrence kinds. This macro is not AUTOSAR specific but used for Dem_NvData structure. The macro definition is in this file because it belongs to the Dem_GetDTCByOccurrenceTime() functionality.
Value	0x04U



5.3.3.2.106. DEM_MODULE_ID

Purpose	AUTOSAR module identification.
Value	54U

5.3.3.2.107. DEM_MOST_RECENT_FAILED_DTC

Purpose	Most recent failed DTC requested.
Value	0x02U

5.3.3.2.108. DEM_MOST_REC_DET_CONFIRMED_DTC

Purpose	Most recently detected confirmed DTC requested.
Value	0x04U

5.3.3.2.109. DEM_NO_EVENT_AVAILABLE

Purpose	UDS status of a disabled event.
Value	0x00U

5.3.3.2.110. DEM_NO_SUCH_ELEMENT

Purpose	The requested record number is not supported by the event.
Value	0x30U

5.3.3.2.111. DEM_NUMBER_FAILED

Purpose	Getting number of filtered DTCs failed.
Value	0x01U

5.3.3.2.112. DEM_NUMBER_OK

Purpose	Getting number of filtered DTCs was successful.
---------	---



Value	0x00U
--------------	-------

5.3.3.2.113. DEM_NUMBER_PENDING

Purpose	Getting number of filtered DTCs is pending.
Value	0x02U

5.3.3.2.114. DEM_NUM_OBD_RDY_GROUPS

Purpose	
Value	0x0CU

5.3.3.2.115. DEM_NUM_OF_DEVFF_KIND

Purpose	Maximal number of configurable development freeze frames kinds.
Value	2U

5.3.3.2.116. DEM_OPAQUE

Purpose	opaque endianness type used for SR interfaces
Value	0x02U

5.3.3.2.117. DEM_PENDING

Purpose	The requested operation is currently in progress.
Value	0x04U

5.3.3.2.118. DEM_READ_DATA_CS_PROTOTYPE_BASIC

Purpose	uint8 data type used for CS interfaces
Value	0x00U



5.3.3.2.119. DEM_READ_DATA_CS_PROTOTYPE_WITH_EVENTID

Purpose	data type used for CS interfaces with EventId
Value	0x01U

5.3.3.2.120. DEM_READ_DATA_SR_DATA_TYPE_UINT16

Purpose	uint16 data type used for SR interfaces
Value	0x03U

5.3.3.2.121. DEM_READ_DATA_SR_DATA_TYPE_UINT32

Purpose	uint32 data type used for SR interfaces
Value	0x04U

5.3.3.2.122. DEM_READ_DATA_SR_DATA_TYPE_UINT8

Purpose	uint8 data type used for SR interfaces
Value	0x02U

5.3.3.2.123. DEM_SEVERITY_CHECK_AT_NEXT_HALT

Purpose	Check at next halt.
Value	0x40U

5.3.3.2.124. DEM_SEVERITY_CHECK_IMMEDIATELY

Purpose	Check immediately.
Value	0x80U

5.3.3.2.125. DEM_SEVERITY_MAINTENANCE_ONLY

Purpose	Maintenance required.
----------------	-----------------------



Value	0x20U
--------------	-------

5.3.3.2.126. DEM_SEVERITY_NO_SEVERITY

Purpose	No severity information available.
Value	0x00U

5.3.3.2.127. DEM_SEVERITY_WWHOBD_CLASS_A

Purpose	WWH-OBD Class A.
Value	0x02U

5.3.3.2.128. DEM_SEVERITY_WWHOBD_CLASS_B1

Purpose	WWH-OBD Class B1.
Value	0x04U

5.3.3.2.129. DEM_SEVERITY_WWHOBD_CLASS_B2

Purpose	WWH-OBD Class B2.
Value	0x08U

5.3.3.2.130. DEM_SEVERITY_WWHOBD_CLASS_C

Purpose	WWH-OBD Class C.
Value	0x10U

5.3.3.2.131. DEM_SEVERITY_WWHOBD_CLASS_NO_CLASS

Purpose	No class information.
Value	0x01U



5.3.3.2.132. DEM_SI30_STATUS_AGEDDTTC

Purpose	DTC status indicator aged DTC status-bit.
Value	0x08U

5.3.3.2.133. DEM_SI30_STATUS_EMISSIONRELATEDDTTC

Purpose	DTC status indicator emission related DTC status-bit.
Value	0x40U

5.3.3.2.134. DEM_SI30_STATUS_SYMPTOM_SLC

Purpose	DTC status indicator symptom since last clear status-bit.
Value	0x10U

5.3.3.2.135. DEM_SI30_STATUS_TFSLCAGED

Purpose	DTC status indicator test failed since last clear or aged status-bit.
Value	0x80U

5.3.3.2.136. DEM_SI30_STATUS_UNCONFIRMEDDTTC

Purpose	DTC status indicator unconfirmed DTC status-bit.
Value	0x01U

5.3.3.2.137. DEM_SI30_STATUS_UNCONFIRMEDDTTC_SLC

Purpose	DTC status indicator unconfirmed DTC since last clear status-bit.
Value	0x04U

5.3.3.2.138. DEM_SI30_STATUS_UNCONFIRMEDDTTC_TOC

Purpose	DTC status indicator unconfirmed DTC in this operation cycle status-bit.
----------------	--



Value	0x02U
--------------	-------

5.3.3.2.139. DEM_SI30_STATUS_WIR_SLC

Purpose	DTC status indicator warning indicator request since last clear status-bit.
Value	0x20U

5.3.3.2.140. DEM_SID_AssembleEDOnFDCThresholdData

Purpose	Service Id for Dem_AssembleEDOnFDCThresholdData().
Value	0x87U

5.3.3.2.141. DEM_SID_AssembleEDSegIntData

Purpose	Service Id for Dem_AssembleEDSegIntData().
Value	0x86U

5.3.3.2.142. DEM_SID_CalloutDynamicDTCFnc

Purpose	Service Id for configured dynamic DTC callout function.
Value	0x80U

5.3.3.2.143. DEM_SID_ClearDTC

Purpose	Service Id for Dem_ClearDTC() according with ASR42 or ASR431.
Value	0x23U

5.3.3.2.144. DEM_SID_ClearPrestoredFreezeFrame

Purpose	Service Id for Dem_ClearPrestoredFreezeFrame() .
Value	0x07U



5.3.3.2.145. DEM_SID_DcmGetAvailableOBDMIDs

Purpose	Service Id for Dem_DcmGetAvailableOBDMIDs() .
Value	0xA3U

5.3.3.2.146. DEM_SID_DcmGetDTCOfOBDFreezeFrame

Purpose	Service Id for Dem_DcmGetDTCOfOBDFreezeFrame() .
Value	0x53U

5.3.3.2.147. DEM_SID_DcmGetDTRData

Purpose	Service Id for Dem_DcmGetDTRData() .
Value	0xA5U

5.3.3.2.148. DEM_SID_DcmGetInfoTypeValue08

Purpose	Service Id for Dem_DcmGetInfoTypeValue08() .
Value	0x6BU

5.3.3.2.149. DEM_SID_DcmGetInfoTypeValue0B

Purpose	Service Id for Dem_DcmGetInfoTypeValue0B() .
Value	0x6CU

5.3.3.2.150. DEM_SID_DcmGetNumTIDsOfOBDMID

Purpose	Service Id for Dem_DcmGetNumTIDsOfOBDMID() .
Value	0xA4U

5.3.3.2.151. DEM_SID_DcmReadDataOfOBDFreezeFrame

Purpose	Service Id for Dem_DcmReadDataOfOBDFreezeFrame() .
---------	--



Value	0x52U
--------------	-------

5.3.3.2.152. DEM_SID_DcmReadDataOfPID01

Purpose	Service Id for Dem_DcmReadDataOfPID01() .
Value	0x61U

5.3.3.2.153. DEM_SID_DcmReadDataOfPID1C

Purpose	Service Id for Dem_DcmReadDataOfPID1C() .
Value	0x63U

5.3.3.2.154. DEM_SID_DcmReadDataOfPID21

Purpose	Service Id for Dem_DcmReadDataOfPID21() .
Value	0x64U

5.3.3.2.155. DEM_SID_DcmReadDataOfPID30

Purpose	Service Id for Dem_DcmReadDataOfPID30() .
Value	0x65U

5.3.3.2.156. DEM_SID_DcmReadDataOfPID31

Purpose	Service Id for Dem_DcmReadDataOfPID31() .
Value	0x66U

5.3.3.2.157. DEM_SID_DcmReadDataOfPID41

Purpose	Service Id for Dem_DcmReadDataOfPID41() .
Value	0x67U



5.3.3.2.158. DEM_SID_DcmReadDataOfPID4D

Purpose	Service Id for Dem_DcmReadDataOfPID4D() .
Value	0x68U

5.3.3.2.159. DEM_SID_DcmReadDataOfPID4E

Purpose	Service Id for Dem_DcmReadDataOfPID4E() .
Value	0x69U

5.3.3.2.160. DEM_SID_DcmReadDataOfPID91

Purpose	Service Id for Dem_DcmReadDataOfPID91() .
Value	0x6AU

5.3.3.2.161. DEM_SID_DisableDTCTRecordUpdate

Purpose	Service Id for Dem_DisableDTCTRecordUpdate() .
Value	0x1AU

5.3.3.2.162. DEM_SID_DisableDTCSetting

Purpose	Service Id for Dem_DisableDTCSetting() .
Value	0x24U

5.3.3.2.163. DEM_SID_EnableDTCTRecordUpdate

Purpose	Service Id for Dem_EnableDTCTRecordUpdate() .
Value	0x1BU

5.3.3.2.164. DEM_SID_EnableDTCSetting

Purpose	Service Id for Dem_EnableDTCSetting() .
---------	---



Value	0x25U
--------------	-------

5.3.3.2.165. DEM_SID_GetDTCByOccurrenceTime

Purpose	Service Id for Dem_GetDTCByOccurrenceTime() .
Value	0x19U

5.3.3.2.166. DEM_SID_GetDTCOfEvent

Purpose	Service Id for Dem_GetDTCOfEvent() .
Value	0x0DU

5.3.3.2.167. DEM_SID_GetDTCSelectionResult

Purpose	Service Id for Dem_GetDTCSelectionResult() .
Value	0xB8U

5.3.3.2.168. DEM_SID_GetDTCSelectionResultForClearDTC

Purpose	Service Id for Dem_GetDTCSelectionResultForClearDTC() .
Value	0xBBU

5.3.3.2.169. DEM_SID_GetDTCStatusAvailabilityMask

Purpose	Service Id for Dem_GetDTCStatusAvailabilityMask() .
Value	0x16U

5.3.3.2.170. DEM_SID_GetDebouncingOfEvent

Purpose	Service Id for Dem_GetDebouncingOfEvent() .
Value	0x9FU



5.3.3.2.171. DEM_SID_GetEventExtendedDataRecord

Purpose	Service Id for Dem_GetEventExtendedDataRecord() .
Value	0x30U

5.3.3.2.172. DEM_SID_GetEventExtendedDataRecordEx

Purpose	Service Id for Dem_GetEventExtendedDataRecordEx() .
Value	0x6DU

5.3.3.2.173. DEM_SID_GetEventFailed

Purpose	Service Id for Dem_GetEventFailed() .
Value	0x0BU

5.3.3.2.174. DEM_SID_GetEventFreezeFrameData

Purpose	Service Id for Dem_GetEventFreezeFrameData() .
Value	0x31U

5.3.3.2.175. DEM_SID_GetEventFreezeFrameDataEx

Purpose	Service Id for Dem_GetEventFreezeFrameDataEx() .
Value	0x6EU

5.3.3.2.176. DEM_SID_GetEventMemoryOverflow

Purpose	Service Id for Dem_GetEventMemoryOverflow() .
Value	0x32U

5.3.3.2.177. DEM_SID_GetEventStatus

Purpose	Service Id for Dem_GetEventStatus() .
---------	---



Value	0x0AU
--------------	-------

5.3.3.2.178. DEM_SID_GetEventTested

Purpose	Service Id for Dem_GetEventTested() .
Value	0x0CU

5.3.3.2.179. DEM_SID_GetFaultDetectionCounter

Purpose	Service Id for Dem_GetFaultDetectionCounter() .
Value	0x3EU

5.3.3.2.180. DEM_SID_GetFunctionalUnitOfDTC

Purpose	Service Id for Dem_GetFunctionalUnitOfDTC() .
Value	0x34U

5.3.3.2.181. DEM_SID_GetIndicatorStatus

Purpose	Service Id for Dem_GetIndicatorStatus() .
Value	0x29U

5.3.3.2.182. DEM_SID_GetNextExtendedDataRecord

Purpose	Service Id for Dem_GetNextExtendedDataRecord() .
Value	0x20U

5.3.3.2.183. DEM_SID_GetNextFilteredDTC

Purpose	Service Id for Dem_GetNextFilteredDTC() .
Value	0x18U



5.3.3.2.184. DEM_SID_GetNextFilteredDTCAndFDC

Purpose	Service Id for Dem_GetNextFilteredDTCAndFDC() .
Value	0x3BU

5.3.3.2.185. DEM_SID_GetNextFilteredDTCAndSeverity

Purpose	Service Id for Dem_GetNextFilteredDTCAndSeverity() .
Value	0x3DU

5.3.3.2.186. DEM_SID_GetNextFilteredRecord

Purpose	Service Id for Dem_GetNextFilteredRecord() .
Value	0x3AU

5.3.3.2.187. DEM_SID_GetNextFilteredStoredDTC

Purpose	Service Id for Dem_DcmGetNextFilteredStoredDTC() .
Value	0xD1U

5.3.3.2.188. DEM_SID_GetNextFreezeFrameData

Purpose	Service Id for Dem_GetNextFreezeFrameData() .
Value	0x1DU

5.3.3.2.189. DEM_SID_GetNumberOfEventMemoryEntries

Purpose	Service Id for Dem_GetNumberOfEventMemoryEntries() .
Value	0x35U

5.3.3.2.190. DEM_SID_GetNumberOfFilteredDTC

Purpose	Service Id for Dem_GetNumberOfFilteredDTC() .
---------	---



Value	0x17U
--------------	-------

5.3.3.2.191. DEM_SID_GetPfcCycleQualified

Purpose	Service Id for Dem_GetPfcCycleQualified() .
Value	0xABU

5.3.3.2.192. DEM_SID_GetSI30Status

Purpose	Service Id for Dem_GetSI30Status() .
Value	0x83U

5.3.3.2.193. DEM_SID_GetSeverityOfDTC

Purpose	Service Id for Dem_GetSeverityOfDTC() .
Value	0x0EU

5.3.3.2.194. DEM_SID.GetSizeOfExtendedDataRecordSelection

Purpose	Service Id for Dem.GetSizeOfExtendedDataRecordSelection() .
Value	0x21U

5.3.3.2.195. DEM_SID.GetSizeOfFreezeFrameSelection

Purpose	Service Id for Dem.GetSizeOfFreezeFrameSelection() .
Value	0x1FU

5.3.3.2.196. DEM_SID_GetStatusOfDTC

Purpose	Service Id for Dem.GetStatusOfDTC() .
Value	0x15U



5.3.3.2.197. DEM_SID_GetTranslationType

Purpose	Service Id for Dem_GetTranslationType() .
Value	0x3CU

5.3.3.2.198. DEM_SID_GetVersionInfo

Purpose	Service Id for Dem_GetVersionInfo() .
Value	0x00U

5.3.3.2.199. DEM_SID_Init

Purpose	Service Id for Dem_Init() .
Value	0x02U

5.3.3.2.200. DEM_SID_J1939DcmClearDTC

Purpose	Service Id for DEM_SID_J1939DcmClearDTC() .
Value	0x95U

5.3.3.2.201. DEM_SID_J1939DcmFirstDTCwithLampStatus

Purpose	Service Id for Dem_J1939DcmFirstDTCwithLampStatus() .
Value	0x93U

5.3.3.2.202. DEM_SID_J1939DcmGetNextDTCwithLampStatus

Purpose	Service Id for Dem_J1939DcmGetNextDTCwithLampStatus() .
Value	0x94U

5.3.3.2.203. DEM_SID_J1939DcmGetNextFilteredDTC

Purpose	Service Id for Dem_J1939DcmGetNextFilteredDTC() .
---------	---



Value	0x92U
--------------	-------

5.3.3.2.204. DEM_SID_J1939DcmGetNextFilteredRatio

Purpose	Service Id for DEM_SID_J1939DcmGetNextFilteredRatio() .
Value	0x9AU

5.3.3.2.205. DEM_SID_J1939DcmGetNextFreezeFrame

Purpose	Service Id for DEM_SID_J1939DcmGetNextFreezeFrame() .
Value	0x97U

5.3.3.2.206. DEM_SID_J1939DcmGetNextSPNInFreezeFrame

Purpose	Service Id for Dem_J1939DcmGetNextSPNInFreezeFrame() .
Value	0x98U

5.3.3.2.207. DEM_SID_J1939DcmGetNumberOfFilteredDTC

Purpose	Service Id for Dem_J1939DcmGetNumberOfFilteredDTC() .
Value	0x91U

5.3.3.2.208. DEM_SID_J1939DcmReadDiagnosticReadiness1

Purpose	Service Id for DEM_SID_J1939DcmReadDiagnosticReadiness1() .
Value	0x9BU

5.3.3.2.209. DEM_SID_J1939DcmReadDiagnosticReadiness2

Purpose	Service Id for DEM_SID_J1939DcmReadDiagnosticReadiness2() .
Value	0x9CU



5.3.3.2.210. DEM_SID_J1939DcmReadDiagnosticReadiness3

Purpose	Service Id for DEM_SID_J1939DcmReadDiagnosticReadiness3() .
Value	0x9DU

5.3.3.2.211. DEM_SID_J1939DcmSetDTCFilter

Purpose	Service Id for Dem_J1939DcmSetDTCFilter() .
Value	0x90U

5.3.3.2.212. DEM_SID_J1939DcmSetFreezeFrameFilter

Purpose	Service Id for DEM_SID_J1939DcmSetFreezeFrameFilter() .
Value	0x96U

5.3.3.2.213. DEM_SID_J1939DcmSetRatioFilter

Purpose	Service Id for DEM_SID_J1939DcmSetRatioFilter() .
Value	0x99U

5.3.3.2.214. DEM_SID_MainFunction

Purpose	Service Id for Dem_MainFunction() .
Value	0x55U

5.3.3.2.215. DEM_SID_PreInit

Purpose	Service Id for Dem_PreInit() .
Value	0x01U

5.3.3.2.216. DEM_SID_PrestoreFreezeFrame

Purpose	Service Id for Dem_PrestoreFreezeFrame() .
---------	--



Value	0x06U
--------------	-------

5.3.3.2.217. DEM_SID_ProcessUnconfirmedThreshold

Purpose	Service Id for Dem_DebounceEventCounterBased() .
Value	0x82U

5.3.3.2.218. DEM_SID_ReplUMPRDenLock

Purpose	Service Id for Dem_ReplUMPRDenLock() .
Value	0x71U

5.3.3.2.219. DEM_SID_ReplUMPRDenRelease

Purpose	Service Id for Dem_ReplUMPRDenRelease() .
Value	0x72U

5.3.3.2.220. DEM_SID_ReplUMPRFaultDetect

Purpose	Service Id for Dem_ReplUMPRFaultDetect() .
Value	0x73U

5.3.3.2.221. DEM_SID_ReportErrorStatus

Purpose	Service Id for Dem_ReportErrorStatus() .
Value	0x0FU

5.3.3.2.222. DEM_SID_ResetEventDebounceStatus

Purpose	Service Id for Dem_ResetEventDebounceStatus() .
Value	0x09U



5.3.3.2.223. DEM_SID_ResetEventStatus

Purpose	Service Id for Dem_ResetEventStatus() .
Value	0x05U

5.3.3.2.224. DEM_SID_ResetReadiness

Purpose	Service Id for Dem_ResetReadiness() .
Value	0xACU

5.3.3.2.225. DEM_SID_SelectDTC

Purpose	Service Id for Dem_SelectDTC() .
Value	0xB7U

5.3.3.2.226. DEM_SID_SelectExtendedDataRecord

Purpose	Service Id for Dem_SelectExtendedDataRecord() .
Value	0xBAU

5.3.3.2.227. DEM_SID_SelectFreezeFrameData

Purpose	Service Id for Dem_SelectFreezeFrameData() .
Value	0xB9U

5.3.3.2.228. DEM_SID_SetAgingCycleCounterValue

Purpose	Service Id for Dem_SetAgingCycleCounterValue() .
Value	0x12U

5.3.3.2.229. DEM_SID_SetAgingCycleState

Purpose	Service Id for Dem_SetAgingCycleState() .
---------	---



Value	0x11U
--------------	-------

5.3.3.2.230. DEM_SID_SetCycleQualified

Purpose	Service Id for Dem_SetCycleQualified() .
Value	0x56U

5.3.3.2.231. DEM_SID_SetDTCFilter

Purpose	Service Id for Dem_SetDTCFilter() .
Value	0x13U

5.3.3.2.232. DEM_SID_SetDTCSuppression

Purpose	Service Id for Dem_SetDTCSuppression() .
Value	0x33U

5.3.3.2.233. DEM_SID_SetEnableCondition

Purpose	Service Id for Dem_SetEnableCondition() .
Value	0x39U

5.3.3.2.234. DEM_SID_SetEventAvailable

Purpose	Service Id for Dem_SetEventAvailable() .
Value	0x37U

5.3.3.2.235. DEM_SID_SetEventDisabled

Purpose	Service Id for Dem_SetEventDisabled() .
Value	0x51U



5.3.3.2.236. DEM_SID_SetEventStatus

Purpose	Service Id for Dem_SetEventStatus() .
Value	0x04U

5.3.3.2.237. DEM_SID_SetFreezeFrameRecordFilter

Purpose	Service Id for Dem_SetFreezeFrameRecordFilter() .
Value	0x3FU

5.3.3.2.238. DEM_SID_SetOperationCycleCntValue

Purpose	Service Id for Dem_SetOperationCycleCntValue() .
Value	0x36U

5.3.3.2.239. DEM_SID_SetOperationCycleState

Purpose	Service Id for Dem_SetOperationCycleState() .
Value	0x08U

5.3.3.2.240. DEM_SID_SetPfcCycleQualified

Purpose	Service Id for Dem_SetPfcCycleQualified() .
Value	0xAAU

5.3.3.2.241. DEM_SID_SetPtoStatus

Purpose	Service Id for Dem_SetPtoStatus() .
Value	0x79U

5.3.3.2.242. DEM_SID_SetSI30Notification

Purpose	Service Id for Dem_SetSI30Notification() .
---------	--



Value	0x85U
--------------	-------

5.3.3.2.243. DEM_SID_SetSI30Symptom

Purpose	Service Id for Dem_SetSI30Symptom() .
Value	0x84U

5.3.3.2.244. DEM_SID_SetStorageCondition

Purpose	Service Id for Dem_SetStorageCondition() .
Value	0x38U

5.3.3.2.245. DEM_SID_SetStoredDTCFilter

Purpose	Service Id for Dem_DcmSetStoredDTCFilter() .
Value	0xD0U

5.3.3.2.246. DEM_SID_SetWIRStatus

Purpose	Service Id for Dem_SetWIRStatus() .
Value	0x7AU

5.3.3.2.247. DEM_SID_Shutdown

Purpose	Service Id for Dem_Shutdown() .
Value	0x03U

5.3.3.2.248. DEM_SID_StoreImmediatelyNext

Purpose	Service Id for Dem_StoreImmediatelyNext() .
Value	0x81U



5.3.3.2.249. DEM_SIZE_OF_AGING_COUNTER

Purpose	Size of Aging Counter.
Value	1U

5.3.3.2.250. DEM_SIZE_STORAGE_ORDER_ID

Purpose	Size of the StorageOrderld.
Value	3U

5.3.3.2.251. DEM_SW_MAJOR_VERSION

Purpose	AUTOSAR module major version.
Value	6U

5.3.3.2.252. DEM_SW_MINOR_VERSION

Purpose	AUTOSAR module minor version.
Value	4U

5.3.3.2.253. DEM_SW_PATCH_VERSION

Purpose	AUTOSAR module patch version.
Value	3U

5.3.3.2.254. DEM_UDS_STATUS_CDTC

Purpose	Confirmed DTC status-bit.
Value	0x08U

5.3.3.2.255. DEM_UDS_STATUS_PDTC

Purpose	Pending DTC status-bit.
----------------	-------------------------



Value	0x04U
--------------	-------

5.3.3.2.256. DEM_UDS_STATUS_TF

Purpose	Test Failed status-bit.
Value	0x01U

5.3.3.2.257. DEM_UDS_STATUS_TFSLC

Purpose	Test Failed Since Last Clear status-bit.
Value	0x20U

5.3.3.2.258. DEM_UDS_STATUS_TFTOC

Purpose	Test Failed This Operation Cycle status-bit.
Value	0x02U

5.3.3.2.259. DEM_UDS_STATUS_TNCSLC

Purpose	Test Not Completed Since Last Clear status-bit.
Value	0x10U

5.3.3.2.260. DEM_UDS_STATUS_TNCTOC

Purpose	Test Not Completed This Operation Cycle status-bit.
Value	0x40U

5.3.3.2.261. DEM_UDS_STATUS_WIR

Purpose	Warning Indicator Requested status-bit.
Value	0x80U



5.3.3.2.262. DEM_VENDOR_ID

Purpose	AUTOSAR vendor identification: Elektrobit Automotive GmbH.
Value	1U

5.3.3.2.263. DEM_WRONG_DTC

Purpose	Selected DTC value in selected format does not exist or clearing is restricted.
Value	0x08U

5.3.3.2.264. DEM_WRONG_DTCORIGIN

Purpose	Selected DTCOrigin does not exist.
Value	0x09U

5.3.3.2.265. DEM_WRONG_FILTER

Purpose	Wrong filter selected.
Value	0x01U

5.3.3.3. Objects

5.3.3.3.1. Dem_NvData

Purpose	Struct containing all NVRAM data.
Type	Dem_NvDataType

5.3.3.3.2. Dem_NvGateEntryMirrorData

Purpose	Struct containing gate entry base and mirror data to be read from and written to NVRAM block.
---------	---



Type	Dem_NvGateEntryMirrorDataType
------	---

5.3.3.3.3. Dem_NvGateEntryPrimaryData

Purpose	Struct containing gate entry base and primary data to be read from and written to NVRAM block.
Type	Dem_NvGateEntryPrimaryDataType

5.3.3.3.4. Dem_NvGateEntrySecondaryData

Purpose	Struct containing gate entry base and secondary data to be read from and written to NVRAM block.
Type	Dem_NvGateEntrySecondaryDataType

5.3.3.4. Functions

5.3.3.4.1. Dem_ASR42_GetEventFreezeFrameData

Purpose	Gets the data of a freeze frame by event.	
Synopsis	<code>Std_ReturnType Dem_ASR42_GetEventFreezeFrameData (Dem_EventIdType EventId , uint8 RecordNumber , boolean ReportTotalRecord , uint16 DataId , uint8 * DestBuffer);</code>	
Service ID	DEM_SID_GetEventFreezeFrameData	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
	RecordNumber	This parameter is a unique identifier for a freeze frame record as defined in ISO15031-5 and ISO14229-1.
	ReportTotalRecord	This parameter specifies whether all PIDs/DIDs are requested (TRUE). A dedicated PID/DID is requested by the parameter DataId (FALSE)



	DataId	This parameter specifies the PID/DID that shall be copied to the destination buffer. If ReportTotalRecord is TRUE, the value of DataId is ignored.
Parameters (out)	DestBuffer	This parameter contains a byte pointer that points to the buffer, to which the freeze frame data record shall be written to. The format is raw hexadecimal values and contains no header-information.
Return Value	success of operation	
	E_OK	Operation was successful
	DEM_E_NODATAAVAILABLE	The requested event data is not currently stored (but the request was valid)
	DEM_E_WRONG_RECORDNUMBER	The requested record number is not supported by the event
	DEM_E_WRONG_DIDNUMBER	The requested DID is not supported by the freeze frame

5.3.3.4.2. Dem_ClearDTC

Purpose	Clears single DTCs as well as groups of DTCs.	
Synopsis	Std_ReturnType Dem_ClearDTC (uint8 ClientId);	
Service ID	DEM_SID_ClearDTC	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	DTC successfully cleared.
	E_NOT_OK	No DTC selected.
	DEM_WRONG_DTC	Selected DTC value in selected format does not exist or clearing is limited to group of all DTCs only.
	DEM_WRONG_DTCORIGIN	Selected DTCOrigin does not exist or DTC clearing is prohibited.
	DEM_CLEAR_FAILED	DTC clearing failed.



	DEM_CLEAR_BUSY	Another client is currently clearing DTCs. The requested operation will not be started and the caller shall try again at a later moment.
	DEM_CLEAR_MEMORY_ERROR	An error occurred during erasing a memory location (e.g. if DemClearDTCBehavior is set to DEM_CLRRESP_NON-VOLATILE_FINISH and erasing of non-volatile-block failed).
	DEM_PENDING	Clearing the DTCs is currently in progress. The caller shall call this function again at a later moment.
Description	<p>It shall be used to clear:</p> <ul style="list-style-type: none"> ▶ all the event status related to the specified DTC ▶ all associated event memory entries for these events (extended data and/or freeze frame data, etc.) 	

5.3.3.4.3. Dem_ClearPrestoredFreezeFrame

Purpose	Clears a prestored freeze frame of a specific event.	
Synopsis	Std_ReturnType Dem_ClearPrestoredFreezeFrame (Dem_EventIdType EventId);	
Service ID	<u>DEM_SID_ClearPrestoredFreezeFrame</u>	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
Return Value	success of operation	
	E_OK	ClearPreStoreFreezeFrame was successful
	E_NOT_OK	ClearPreStoreFreezeFrame failed
Description	<p>This API can only be used through the RTE and therefore no declaration is exported via Dem.h.</p> <p>The API shall be called to delete or release the prestored freeze frame for specific Event ID. If the API <u>Dem_SetEventStatus()</u> (passed failed) is called it has the same</p>	



effect - that means it's not necessary to call the API Dem_ClearPrestoredFreezeFrame directly after [Dem_SetEventStatus\(\)](#).

Caveats: Dem configuration during integration of Monitor Functions is system specific.

Configuration: While configuring the Dem the capability of pre-store functionality for the required event has to be defined.

5.3.3.4.4. Dem_DcmGetAvailableOBDMIDs

Purpose	Reports the value of a requested "availability-OBDMID" to the DCM upon a Service \$06 request. Derived from that the tester displays the supported tests a mechanic can select from. API is needed in OBD-relevant ECUs only.	
Synopsis	<code>Std_ReturnType Dem_DcmGetAvailableOBDMIDs (uint8 Obdmid , uint32 * Obdmidvalue);</code>	
Service ID	DEM_SID_DcmGetAvailableOBDMIDs	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	Obdmid	Availability OBDMID (\$00,\$20, \$40...)
Parameters (out)	Obdmidvalue	Bit coded information on the support of OBDMIDs.
Return Value	success of operation	
	E_OK	Report of DTR result successful
	E_NOT_OK	Report of DTR result failed
Description	API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT) Note: Functionality not implemented, only stub available.	

5.3.3.4.5. Dem_DcmGetDTCOfOBDFreezeFrame

Purpose	Gets DTC by freeze frame record number.
Synopsis	<code>Std_ReturnType Dem_DcmGetDTCOfOBDFreezeFrame (uint8 FrameNumber , uint32 * DTC , Dem_DTCFormatType DTCFormat);</code>
Service ID	DEM_SID_DcmGetDTCOfOBDFreezeFrame



Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	FrameNumber	Unique identifier for a freeze frame record as defined in ISO 15031-5. The value 0x00 indicates the complete OBD freeze frame. Other values are reserved for future functionality and are not supported yet.
	DTCFormat	Output format of the DTC value.
Parameters (out)	DTC	Diagnostic Trouble Code. If the return value of the function is other than E_OK this parameter does not contain valid data.
Return Value	success of operation	
	E_OK	operation was successful
	E_NOT_OK	no DTC available
Description	Note: This API must only be used by the Dcm module!	

5.3.3.4.6. Dem_DcmGetDTRData

Purpose	Reports a DTR data along with TID-value, UaSID, test result with lower and upper limit. API is needed in OBD-relevant ECUs only.	
Synopsis	<pre>Std_ReturnType Dem_DcmGetDTRData (uint8 Obdmid , uint8 TIDindex , uint8 * TIDvalue , uint8 * UaSID , uint16 * Testvalue , uint16 * Lowlimvalue , uint16 * Upplimvalue);</pre>	
Service ID	DEM_SID_DcmGetDTRData	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	Obdmid	Identification of a DTR element by assigned DTRId.
	TIDindex	Index of the TID within the DEM. Runs from 0 to "numberOfTIDs" obtained in the call to Dem_DcmGetNumTIDsOfOBDMID()
Parameters (out)	TIDvalue	TID to be put on the tester response
	UaSID	UaSID to be put on the tester response
	Testvalue	Latest test result



	Lowlimvalue	Lower limit value associated to the latest test result
	Upplimvalue	Upper limit value associated to the latest test result
Return Value		success of operation
E_OK		Report of DTR result successful
E_NOT_OK		Report of DTR result failed
Description		<p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOB-DSupport}) != DEM_OBD_NO_OBD_SUPPORT)</p> <p>Note: Functionality not implemented, only stub available.</p>

5.3.3.4.7. Dem_DcmGetInfoTypeValue08

Purpose	Service is used for requesting IUMPR data according to InfoType \$08. This interface is derived from the prototype <Module>_GetInfotypeValueData() defined by the Dcm. Therefore Dcm_OpStatusType and Std_ReturnType are contained. API is needed in OBD-relevant ECUs only.	
Synopsis	<pre>Std_ReturnType Dem_DcmGetInfoTypeValue08 (Dcm_OpStatusType OpStatus , uint8 * Iumprdata08 , uint8 * Iumprdata08BufferSize);</pre>	
Service ID	DEM_SID_DcmGetInfoTypeValue08	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	OpStatus	Only DCM_INITIAL will appear, because this API behaves synchronous.
Parameters (in,out)	Iumprdata08BufferSize	The maximum number of data bytes that can be written to the Iumprdata08 Buffer.
Parameters (out)	Iumprdata08	Buffer containing the number of data elements (as defined in ISO-15031-5) and contents of InfoType \$08. The buffer is provided by the Dcm.
Return Value	Always E_OK is returned.	
Description	<p>API Availability: This API will be available only if ({ecuc(Dem/ DemGeneral.DemOB-DSupport}) != DEM_OBD_NO_OBD_SUPPORT)</p> <p>Note: Functionality not implemented, only stub available.</p>	



Note: This API must only be used by the Dcm module!

5.3.3.4.8. Dem_DcmGetInfoTypeValue0B

Purpose	Service is used for requesting IUMPR data according to InfoType \$0B. This interface is derived from the prototype <Module>_GetInfotypeValueData() defined by the Dcm. Therefore Dcm_OpStatusType and Std_ReturnType are contained. API is needed in OBD-relevant ECUs only.	
Synopsis	Std_ReturnType Dem_DcmGetInfoTypeValue0B (Dcm_OpStatusType OpStatus , uint8 * Iumprdata0B , uint8 * Iumprdata0BBufferSize) ;	
Service ID	DEM_SID_DcmGetInfoTypeValue0B	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	OpStatus	Only DCM_INITIAL will appear, because this API behaves synchronous.
Parameters (in,out)	Iumprdata0BBufferSize	The maximum number of data bytes that can be written to the Iumprdata0B Buffer.
Parameters (out)	Iumprdata0B	Buffer containing the number of data elements (as defined in ISO-15031-5) and contents of InfoType \$0B. The buffer is provided by the Dcm.
Return Value	Always E_OK is returned.	
Description	API Availability: This API will be available only if ({ecuc(Dem/ DemGeneral.DemOB-DSupport)} != DEM_OBD_NO_OBD_SUPPORT) Note: Functionality not implemented, only stub available. Note: This API must only be used by the Dcm module!	

5.3.3.4.9. Dem_DcmGetNextFilteredStoredDTC

Purpose	Gets the next filtered stored DTC.
Synopsis	Std_ReturnType Dem_DcmGetNextFilteredStoredDTC (uint8 ClientId , uint32 * DTC) ;
Service ID	DEM_SID_GetNextFilteredStoredDTC
Sync/Async	Synchronous



Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
Parameters (out)	DTC	Receives the DTC value in UDS format returned by this function. If the return value of the function is other than E_OK this parameter does not contain valid data.
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	Returned next filtered element
	E_NOT_OK	A development error was detected
	DEM_NO_SUCH_ELEMENT	No further element matching the filter criteria found
Description	<p>The API shall return the stored DTC in event memory from the Dem, matching the filter criteria defined by the function call Dem_DcmSetStoredDTCFilter().</p> <p>Note: This API must only be used by the Dcm module!</p> <p>Note: The value DEM_PENDING is never returned, because of the implemented synchronous behavior.</p>	

5.3.3.4.10. Dem_DcmGetNumTIDsOfOBDMID

Purpose	Gets the number of TIDs per (functional) OBDMID. This can be used by the DCM to iteratively request for OBD/TID result data within a loop from 0....numberOfTIDs-1. API is needed in OBD-relevant ECUs only.	
Synopsis	Std_ReturnType Dem_DcmGetNumTIDsOfOBDMID (uint8 Obdmid , uint8 * numberOfTIDs);	
Service ID	DEM_SID_DcmGetNumTIDsOfOBDMID	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	Obdmid	OBDMID subject of the request to identify the number of assigned TIDs.
Parameters (out)	numberOfTIDs	Number of assigned TIDs for the requested OBDMID. Used as loop value for the DCM to retrieve all OBD/TID result data.
Return Value	success of operation	
	E_OK	Get number of TIDs successful



	E_NOT_OK	Get number of TIDs failed
Description	<p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOB-DSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p> <p>Note: Functionality not implemented, only stub available.</p>	

5.3.3.4.11. Dem_DcmReadDataOfOBDFreezeFrame

Purpose	Gets data element per PID and index of the most important freeze frame being selected for the output of service \$02.	
Synopsis	<pre>Std_ReturnType Dem_DcmReadDataOfOBDFreezeFrame (uint8 PID , uint8 DataElementIndexOfPID , uint8 * DestBuffer , uint16 * BufSize);</pre>	
Service ID	DEM_SID_DcmReadDataOfOBDFreezeFrame	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	PID	This parameter is an identifier for a PID as defined in ISO15031-5.
	DataElementIndexOfPID	This parameter is a Data element index of this PID according to the Dcm configuration of service \$02. It is zero-based and consecutive, and ordered by the data element positions.
Parameters (in,out)	DestBuffer	This parameter contains a byte pointer that points to the buffer to which freeze frame data shall be written. The format is raw hexadecimal values and contains no header information.
	BufSize	When the function is called this parameter contains the maximum number of data bytes that can be written to the buffer.
Return Value	success of operation	
	E_OK	Freeze frame data was successfully reported
	E_NOT_OK	Freeze frame data was not successfully reported
Description	The function stores the data in the provided DestBuffer.	



Note: This API must only be used by the Dcm module!

5.3.3.4.12. Dem_DcmReadDataOfPID01

Purpose	Service to report the value of PID \$01 computed by the Dem.	
Synopsis	<code>Std_ReturnType Dem_DcmReadDataOfPID01 (uint8 * PID01value);</code>	
Service ID	DEM_SID_DcmReadDataOfPID01	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	PID01value	Buffer containing the contents of PID\$01 computed by Dem.
Return Value	Always E_OK is returned, as E_NOT_OK will never appear.	
Description	Note: This API must only be used by the Dcm module!	

5.3.3.4.13. Dem_DcmReadDataOfPID1C

Purpose	Service to report the value of PID \$1C computed by the Dem.	
Synopsis	<code>Std_ReturnType Dem_DcmReadDataOfPID1C (uint8 * PID1Cvalue);</code>	
Service ID	DEM_SID_DcmReadDataOfPID1C	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	PID1Cvalue	Buffer containing the contents of PID\$1C computed by Dem.
Return Value	Always E_OK is returned, as E_NOT_OK will never appear.	
Description	Note: This API must only be used by the Dcm module!	

5.3.3.4.14. Dem_DcmReadDataOfPID21

Purpose	Service to report the value of PID \$21 computed by the Dem.	
Synopsis	<code>Std_ReturnType Dem_DcmReadDataOfPID21 (uint8 * PID21value);</code>	
Service ID	DEM_SID_DcmReadDataOfPID21	
Sync/Async	Synchronous	



Reentrancy	Reentrant	
Parameters (out)	PID21value	Buffer containing the contents of PID\$21 computed by Dem.
Return Value	Always E_OK is returned, as E_NOT_OK will never appear.	
Description	Note: This API must only be used by the Dcm module!	

5.3.3.4.15. Dem_DcmReadDataOfPID30

Purpose	Service to report the value of PID \$30 computed by the Dem.	
Synopsis	Std_ReturnType Dem_DcmReadDataOfPID30 (uint8 * PID30value);	
Service ID	DEM_SID_DcmReadDataOfPID30	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	PID30value	Buffer containing the contents of PID\$30 computed by Dem.
Return Value	Always E_OK is returned, as E_NOT_OK will never appear.	
Description	Note: This API must only be used by the Dcm module!	

5.3.3.4.16. Dem_DcmReadDataOfPID31

Purpose	Service to report the value of PID \$31 computed by the Dem.	
Synopsis	Std_ReturnType Dem_DcmReadDataOfPID31 (uint8 * PID31value);	
Service ID	DEM_SID_DcmReadDataOfPID31	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	PID31value	Buffer containing the contents of PID\$31 computed by Dem.
Return Value	Always E_OK is returned, as E_NOT_OK will never appear.	
Description	Note: This API must only be used by the Dcm module!	

5.3.3.4.17. Dem_DcmReadDataOfPID41

Purpose	Service to report the value of PID \$41 computed by the Dem.
----------------	--



Synopsis	<code>Std_ReturnType Dem_DcmReadDataOfPID41 (uint8 * PID41value);</code>	
Service ID	DEM_SID_DcmReadDataOfPID41	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	PID41value	Buffer containing the contents of PID\$41 computed by Dem.
Return Value	Always E_OK is returned, as E_NOT_OK will never appear.	
Description	Note: This API must only be used by the Dcm module!	

5.3.3.4.18. Dem_DcmReadDataOfPID4D

Purpose	Service to report the value of PID \$4D computed by the Dem.	
Synopsis	<code>Std_ReturnType Dem_DcmReadDataOfPID4D (uint8 * PID4Dvalue);</code>	
Service ID	DEM_SID_DcmReadDataOfPID4D	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	PID4Dvalue	Buffer containing the contents of PID\$4D computed by Dem.
Return Value	Always E_OK is returned, as E_NOT_OK will never appear.	
Description	Note: Functionality not implemented, only stub available. Note: This API must only be used by the Dcm module!	

5.3.3.4.19. Dem_DcmReadDataOfPID4E

Purpose	Service to report the value of PID \$4E computed by the Dem.	
Synopsis	<code>Std_ReturnType Dem_DcmReadDataOfPID4E (uint8 * PID4Evalue);</code>	
Service ID	DEM_SID_DcmReadDataOfPID4E	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	PID4Evalue	Buffer containing the contents of PID\$4E computed by Dem.
Return Value	Always E_OK is returned, as E_NOT_OK will never appear.	



Description	Note: Functionality not implemented, only stub available. Note: This API must only be used by the Dcm module!
--------------------	--

5.3.3.4.20. Dem_DcmReadDataOfPID91

Purpose	Service to report the value of PID \$91 computed by the Dem. API is needed in OBD-relevant ECUs only.	
Synopsis	<code>Std_ReturnType Dem_DcmReadDataOfPID91 (uint8 * PID91value);</code>	
Service ID	DEM_SID_DcmReadDataOfPID91	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (out)	PID91value	Buffer containing the contents of PID \$91 computed by the Dem. The buffer is provided by the Dcm with the appropriate size, i.e. during configuration, the Dcm identifies the required size from the largest PID in order to configure a PID-Buffer.
Return Value		
Always	E_OK is returned, as E_NOT_OK will never appear.	
Description	API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT) Note: Functionality not implemented, only stub available.	

5.3.3.4.21. Dem_DcmSetStoredDTCFilter

Purpose	Sets a filter for stored DTCs.
Synopsis	<code>Std_ReturnType Dem_DcmSetStoredDTCFilter (uint8 ClientId , Dem_DTCOriginType DTCOrigin , Dem_DataKindType DataKind);</code>
Service ID	DEM_SID_SetStoredDTCFilter
Sync/Async	Synchronous
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.



Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
	DTCOrigin	If the Dem supports more than one event memory this parameter is used to select the source memory the DTCs shall be read from.
	DataKind	DataKind selects only stored DTCs which contain the requested data kind (freeze frame or extended data).
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	Filter was successfully set
	E_NOT_OK	Failed to set stored DTC filter
Description	<p>This API shall be called before Dem_DcmGetNextFilteredStoredDTC(). It sets the new filter criteria and resets the internal counter to the first event that matches the filter settings. The filter mask attributes are used until the next call of Dem_DcmSetStoredDTCFilter or Dem_Init(). This function shall not be called to interrupt a Dem_DcmGetNextFilteredStoredDTC() call sequence and change the filter criteria.</p> <p>Note: This API must only be used by the Dcm module!</p>	

5.3.3.4.22. Dem_DisableDTCRecordUpdate

Purpose	Disables the event memory update of a specific DTC (only one at one time).	
Synopsis	Std_ReturnType Dem_DisableDTCRecordUpdate (uint8 ClientId);	
Service ID	DEM_SID_DisableDTCRecordUpdate	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	Event memory update successfully disabled.
	E_NOT_OK	Failed to disable DTC record update.
	DEM_WRONG_DTC	Selected DTC value in selected format does not exist or a group of DTC was selected.



	DEM_WRONG_DTCORIGIN	Selected DTCOrigin does not exist.
Description	<p>This function shall be used if the freeze frame or extended data record are about to be accessed by subsequent API-calls. It is done to ensure that the data contained in this record is not changed while the freeze frame or extended data record are accessed by the external application, e.g. Dcm.</p> <p>This function shall protect the event related data of the specified DTC within the specified origin from updating or deleting, to allow a consistent read for the following subsequent API-calls: Dem_SelectExtendedDataRecord(), Dem.GetSizeOfExtendedDataRecordSelection(), Dem_GetNextExtendedDataRecord(), Dem_SelectFreezeFrameData(), Dem.GetSizeOfFreezeFrameSelection() and Dem_GetNextFreezeFrameData().</p> <p>New and other events including their associated freeze frames and extended data records can still be added to and changed in the event memory as long as space is available. Event related data might still be updated in background (e.g. Dem-internal data elements).</p> <p>DTC status information update is not affected by this function.</p> <p>Note: This API must only be used by the Dcm module!</p>	

5.3.3.4.23. Dem_DisableDTCSetting

Purpose	Disables the storage and the update of UDS status bytes of all DTCs.	
Synopsis	<code>Std_ReturnType Dem_DisableDTCSetting (uint8 ClientId);</code>	
Service ID	DEM_SID_DisableDTCSetting	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	The operation was successful
	E_NOT_OK	On development error or if DTC setting was already disabled by a different client
Description	<p>This function shall be called to disable DTC setting for all DTCs.</p> <p>This is only for preventing DTCs from being stored in case of an induced failure situations in a system, e.g. during flash-reprogramming of one ECU in a network. In that</p>	



	<p>case all the ECU's are commanded via diagnostic request (linked to the above diagnostic request) to suppress storage of a DTC while maintaining correct fail-safe behavior as the flashed ECU is not participating in the normal communication anymore. If one of the other networked ECUs needs one of the signals which are now missing, this will lead to a failsafe-reaction of the ECU (as per the AUTOSAR concept the failsafe reaction of an ECU is triggered by certain event-status updates or a FiM-command which is itself triggered by an event-status update).</p> <p>Note: This API must only be used by the Dcm module!</p> <p>Note: This API can be used only by one client ID at a time.</p>
--	--

5.3.3.4.24. Dem_EnableDTCRecordUpdate

Purpose	Enables the event memory update of the DTC disabled by Dem_DisableDT-CRecordUpdate() before.	
Synopsis	<code>Std_ReturnType Dem_EnableDTCRecordUpdate (uint8 ClientId);</code>	
Service ID	DEM_SID_EnableDTCRecordUpdate	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Return Value	Status of the operation of type <code>Std_ReturnType</code>	
	<code>E_OK</code>	Event memory update successfully enabled.
	<code>E_NOT_OK</code>	Failed to enable DTC record update.
Description	<p>The function <code>Dem_EnableDTCRecordUpdate</code> is the counterpart to the function Dem_DisableDT-CRecordUpdate(). It enables the event memory update of the DTC disabled by Dem_DisableDT-CRecordUpdate() before. It shall be called after the freeze frame and extended data record were protected by the function Dem_DisableDT-CRecordUpdate(), after the access by subsequent API-calls is finished.</p> <p>It is called to release the currently disabled DTC which has been protected by the function Dem_DisableDT-CRecordUpdate(), so that the data can be updated again.</p> <p>Note: This API must only be used by the Dcm module!</p>	

5.3.3.4.25. Dem_EnableDTCSetting

Purpose	(Re)-Enables the storage and the update of UDS status bytes of all DTCs
----------------	---



Synopsis	<code>Std_ReturnType Dem_EnableDTCSetting (uint8 ClientId);</code>	
Service ID	DEM_SID_EnableDTCSetting	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	The operation was successful
	E_NOT_OK	On development error or if a different client was responsible for disabling the DTC setting
Description	<p>This function shall be called to enable the DTC setting for all DTCs. See also Dem_DisableDTCSetting().</p> <p>Note: This API must only be used by the Dcm module!</p> <p>Note: This API can be used only by one client ID at a time.</p> <p>Note: If configured, the InitMonitorForEvent callback functions are triggered asynchronously from Dem_MainFunction() when the DTC setting is re-enabled.</p>	

5.3.3.4.26. Dem_GetDTCByOccurrenceTime

Purpose	Gets the DTC by occurrence time.	
Synopsis	<code>Std_ReturnType Dem_GetDTCByOccurrenceTime (uint8 ClientId , Dem_DTCRequestType DTCRequest , uint32 * DTC);</code>	
Service ID	DEM_SID_GetDTCByOccurrenceTime	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
		This parameter defines the request type of the DTC.
Parameters (out)	DTC	Receives the DTC value in UDS format returned by the function. If the return value of the function is other than E_OK, this parameter does not contain valid data.



Return Value	Status of the operation of type Std_ReturnType	
	E_OK	The call was successful and the parameter DTC contains the requested element.
	E_NOT_OK	A development error was detected.
	DEM_NO_SUCH_ELEMENT	The requested element is not stored.
Description	The API provides the capability to access specific events stored in the primary memory at certain important points in time. The API call will return one DTC according to the Dem_DTCRequestType .	

5.3.3.4.27. Dem_GetDTCOfEvent

Purpose	Function to get the DTC of an event.	
Synopsis	Std_ReturnType Dem_GetDTCOfEvent (Dem_EventIdType EventId , Dem_DTCFormatType DTCFormat , uint32 * DTCOfEvent);	
Service ID	DEM_SID_GetDTCOfEvent	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
	DTCFormat	This parameter defines the output-format of the requested DTC value, either OBD DTC format or UDS DTC format.
Parameters (out)	DTCOfEvent	Receives the DTC value returned by the function. If the return value of the function is other than OK this parameter does not contain valid data.
Return Value	success of operation	
	E_OK	get of DTC was successful
	E_NOT_OK	get of DTC failed
	DEM_E_NO_DTC_AVAILABLE	there is no DTC configured in the requested format
Description	<p>Gets the DTC which is mapped to EventId by Dem Configuration.</p> <p>Configuration: Mapping of Events to DTCs is configured in Dem. Mapping is "n to 1" or "1 to n".</p>	



5.3.3.4.28. Dem_GetDTCSelectionResult

Purpose	Provides the status of the DTC selection request.	
Synopsis	<code>Std_ReturnType Dem_GetDTCSelectionResult (uint8 ClientId);</code>	
Service ID	DEM_SID_GetDTCSelectionResult	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	The DTC select parameter check is successful and the requested DTC or group of DTC in the selected origin is selected for further operations.
	E_NOT_OK	No DTC selected.
	DEM_WRONG_DTC	Selected DTC value in selected format does not exist.
	DEM_WRONG_DTCORIGIN	Selected DTCOrigin does not exist.
Description	This API provides information if the last call to Dem_SelectDTC() has selected a valid DTC or group of DTCs. The API is optionally called after a Dem_SelectDTC() . APIs which depend on Dem_SelectDTC() have the same behavior with or without the call of Dem_GetDTCSelectionResult.	

5.3.3.4.29. Dem_GetDTCSelectionResultForClearDTC

Purpose	Provides information if the last call to Dem_SelectDTC() has selected a valid DTC or group of DTCs.	
Synopsis	<code>Std_ReturnType Dem_GetDTCSelectionResultForClearDTC (uint8 ClientId);</code>	
Service ID	DEM_SID_GetDTCSelectionResultForClearDTC	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.



Return Value	Status of the operation of type Std_ReturnType	
	E_OK	The DTC select parameter check is successful and the requested DTC or group of DTC in the selected origin is selected for further operations.
	E_NOT_OK	No DTC selected.
	DEM_WRONG_DTC	Selected DTC value in selected format does not exist.
	DEM_WRONG_DTCORIGIN	Selected DTCOrigin does not exist or ClearDTC is prohibited.
Description	The function provides information if the last call to Dem_SelectDTC() has selected a valid DTC or group of DTCs for Dem_ClearDTC() .	

5.3.3.4.30. Dem_GetDTCStatusAvailabilityMask

Purpose	Gets the DTC Status availability mask.	
Synopsis	<code>Std_ReturnType Dem_GetDTCStatusAvailabilityMask (uint8 ClientId , Dem_UdsStatusByteType * DTCStatusMask);</code>	
Service ID	DEM_SID_GetDTCStatusAvailabilityMask	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	ClientId	client ID for which the status availability mask will be retrieved.
Parameters (out)	DTCStatusMask	The value from type Dem_DTCStatusMaskType indicates the supported DTC status bits from the Dem. All supported information is indicated by setting the corresponding status bit to 1.
Return Value	success of operation	
	E_OK	get of DTC status availability mask was successful
	E_NOT_OK	get of DTC status availability mask failed
Description	The API shall be used to get the DTC status availability mask, i.e. the DTC status information (according to ISO14229) supported by the Dem. Only supported bits can be used as filter parameters in the API Dem_SetDTCFilter() .	



Note: This API must only be used by the Dcm module!

5.3.3.4.31. Dem_GetDebouncingOfEvent

Purpose	Gets the debouncing status of an event.	
Synopsis	<code>Std_ReturnType Dem_GetDebouncingOfEvent (Dem_EventIdType EventId , Dem_DebouncingStateType * DebouncingState);</code>	
Service ID	DEM_SID_GetDebouncingOfEvent	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
Parameters (out)	DebouncingState	Bit 0 Temporarily Defective (corresponds to $0 < FDC < 127$) Bit 1 finally Defective (corresponds to $FDC = 127$) Bit 2 temporarily healed (corresponds to $-128 < FDC < 0$) Bit 3 Test complete (corresponds to $FDC = -128$ or $FDC = 127$) Bit 4 DTR Update (= Test complete && Debouncing complete && enable conditions / storage conditions fulfilled)
Return Value	success of operation E_OK E_NOT_OK DEM_E_NODATAAVAILABLE DEM_E_WRONG_RECORDNUMBER DEM_E_WRONG_BUFFERSIZE	
Description	Note: Functionality not implemented, only stub available.	

5.3.3.4.32. Dem_GetEventExtendedDataRecord

Purpose	Gets the data of an extended data record by event.
Synopsis	<code>Std_ReturnType Dem_GetEventExtendedDataRecord (Dem_EventIdType EventId , uint8 RecordNumber , uint8 * DestBuffer);</code>



Service ID	DEM_SID_GetEventExtendedDataRecord	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
	RecordNumber	Identification of requested Extended data record. Valid values are between 0x01 and 0xEF as defined in ISO14229-1. 0xFF means data of all extended data records are returned
Parameters (out)	DestBuffer	This parameter contains a byte pointer that points to the buffer, to which the extended data shall be written to. The format is raw hexadecimal values and contains no header-information.
Return Value	success of operation	
	E_OK	Operation was successful
	E_NOT_OK	Operation failed

5.3.3.4.33. Dem_GetEventExtendedDataRecordEx

Purpose	Gets the data of an extended data record by event.	
Synopsis	<pre>Std_ReturnType Dem_GetEventExtendedDataRecordEx (Dem_EventId- Type EventId , uint8 RecordNumber , uint8 * DestBuffer , uint16 * BufSize);</pre>	
Service ID	DEM_SID_GetEventExtendedDataRecordEx	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
	RecordNumber	Identification of requested Extended data record. Valid values are between 0x01 and 0xEF as defined in ISO14229-1. 0xFF means data of all extended data records are returned



Parameters (out)	DestBuffer	This parameter contains a byte pointer that points to the buffer, to which the extended data shall be written to. The format is raw hexadecimal values and contains no header-information.
	BufSize	When the function is called this parameter contains the maximum number of data bytes that can be written to the buffer. The function returns the actual number of written data bytes in this parameter.
Return Value	success of operation	
	E_OK	Operation was successful
	E_NOT_OK	Operation could not be performed
	DEM_E_NODATAAVAILABLE	The requested event data is not currently stored (but the request was valid)
	DEM_E_WRONG_RECORDNUMBER	The requested record number is not supported by the event
	DEM_E_WRONG_BUFFERSIZE	The provided buffer size is too small.
Description	Note: Functionality not implemented, only stub available.	

5.3.3.4.34. Dem_GetEventFailed

Purpose	Gets the event failed status of an event.	
Synopsis	<pre>Std_ReturnType Dem_GetEventFailed (Dem_EventIdType EventId , boolean * EventFailed);</pre>	
Service ID	DEM_SID_GetEventFailed	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
Parameters (out)	EventFailed	<ul style="list-style-type: none"> ▶ TRUE Last Failed ▶ FALSE not Last Failed
Return Value	success of operation	
	E_OK	get of event failed was successful
	E_NOT_OK	get of event failed failed



Description	This API shall be used to read Bit 0 of Dem_EventStatusExtendedType from the Dem. For the DCM the API Dem_GetStatusOfDTC() is used.
--------------------	--

5.3.3.4.35. Dem_GetEventFreezeFrameData

Purpose	Gets the data of a freeze frame by event.	
Synopsis	<pre>Std_ReturnType Dem_GetEventFreezeFrameData (Dem_EventIdType EventId , uint8 RecordNumber , boolean ReportTotalRecord , uint16 DataId , uint8 * DestBuffer);</pre>	
Service ID	DEM_SID_GetEventFreezeFrameData	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
	RecordNumber	This parameter is a unique identifier for a freeze frame record as defined in ISO15031-5 and ISO14229-1.
	ReportTotalRecord	This parameter specifies whether all PIDs/DIDs are requested (TRUE). A dedicated PID/DID is requested by the parameter DataId (FALSE)
	DataId	This parameter specifies the PID/DID that shall be copied to the destination buffer. If ReportTotalRecord is TRUE, the value of DataId is ignored.
Parameters (out)	DestBuffer	This parameter contains a byte pointer that points to the buffer, to which the freeze frame data record shall be written to. The format is raw hexadecimal values and contains no header-information.
Return Value	success of operation E_OK Operation was successful E_NOT_OK Operation failed	
Description	Note: Optional development freeze frames are not supported by this API.	



5.3.3.4.36. Dem_GetEventFreezeFrameDataEx

Purpose	Gets the data of a freeze frame by event.	
Synopsis	<pre>Std_ReturnType Dem_GetEventFreezeFrameDataEx (Dem_EventIdType EventId , uint8 RecordNumber , uint16 DataId , uint8 * DestBuffer , uint16 * BufSize);</pre>	
Service ID	DEM_SID_GetEventFreezeFrameData	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
	RecordNumber	This parameter is a unique identifier for a freeze frame record as defined in ISO15031-5 and ISO14229-1.
	DataId	This parameter specifies the PID/DID that shall be copied to the destination buffer. If ReportTotalRecord is TRUE, the value of DataId is ignored.
Parameters (out)	DestBuffer	This parameter contains a byte pointer that points to the buffer, to which the freeze frame data record shall be written to. The format is raw hexadecimal values and contains no header-information.
	BufSize	When the function is called this parameter contains the maximum number of data bytes that can be written to the buffer.
Return Value	success of operation	
	E_OK	Operation was successful
	E_NOT_OK	Operation could not be performed
	DEM_E_NODATAAVAILABLE	The requested event data is not currently stored (but the request was valid)
	DEM_E_WRONG_RECORDNUMBER	The requested record number is not supported by the event
	DEM_E_WRONG_DIDNUMBER	The requested DID is not supported by the freeze frame
	DEM_E_WRONG_BUFFERSIZE	The provided buffer size is too small.
	Note: Functionality not implemented, only stub available.	



5.3.3.4.37. Dem_GetEventMemoryOverflow

Purpose	Gets the event memory overflow indication status.	
Synopsis	<pre>Std_ReturnType Dem_GetEventMemoryOverflow (Dem_DTCOriginType DTCOrigin , boolean * OverflowIndication);</pre>	
Service ID	DEM_SID_GetEventMemoryOverflow	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	DTCOrigin	If the Dem supports more than one event memory this parameter is used to select the source memory the overflow indication shall be read from.
Parameters (out)	OverflowIndication	This parameter returns TRUE if the respective event memory was overflowed, otherwise it returns FALSE
Return Value	success of operation E_OK E_NOT_OK	
	E_OK	Operation was successful
	E_NOT_OK	Operation failed or is not supported

5.3.3.4.38. Dem_GetEventStatus

Purpose	Gets the current extended event status of an event.	
Synopsis	<pre>Std_ReturnType Dem_GetEventStatus (Dem_EventIdType EventId , Dem_EventStatusExtendedType * EventStatusExtended);</pre>	
Service ID	DEM_SID_GetEventStatus	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
Parameters (out)	EventStatusExtended	for explanation see Dem_EventStatusExtendedType
Return Value	success of operation E_OK E_NOT_OK	
	E_OK	get of event status was successful
	E_NOT_OK	get of event status failed



Description	This API shall be used to read the event status from the Dem. This API is provided to be used by SW-Components or other basic software modules e.g. FiM. For the DCM the API Dem_GetStatusOfDTC() is used.
--------------------	---

5.3.3.4.39. Dem_GetEventTested

Purpose	Gets the event tested status of an event.					
Synopsis	Std_ReturnType Dem_GetEventTested (Dem_EventIdType EventId , boolean * EventTested);					
Service ID	DEM_SID_GetEventTested					
Sync/Async	Synchronous					
Reentrancy	Reentrant					
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.				
Parameters (out)	EventTested	<ul style="list-style-type: none"> ▶ TRUE event tested this cycle ▶ FALSE event not tested this cycle 				
Return Value	success of operation <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">E_OK</td> <td style="padding: 2px;">get of event state "tested" was successful</td> </tr> <tr> <td style="padding: 2px;">E_NOT_OK</td> <td style="padding: 2px;">get of event state "tested" failed</td> </tr> </table>		E_OK	get of event state "tested" was successful	E_NOT_OK	get of event state "tested" failed
E_OK	get of event state "tested" was successful					
E_NOT_OK	get of event state "tested" failed					
Description	This API shall be used to read negated Bit 6 of Dem_EventStatusExtendedType from the Dem. For the DCM the API Dem_GetStatusOfDTC() is used.					

5.3.3.4.40. Dem_GetFaultDetectionCounter

Purpose	Gets the fault detection counter of an event.
Synopsis	Std_ReturnType Dem_GetFaultDetectionCounter (Dem_EventIdType EventId , sint8 * FaultDetectionCounter);
Service ID	DEM_SID_GetFaultDetectionCounter
Sync/Async	Synchronous
Reentrancy	Non reentrant



Parameters (in)	EventId	Provide the EventId value the fault detection counter is requested for. If the return value of the function is other than OK this parameter does not contain valid data.
Parameters (out)	FaultDetectionCounter	This parameter receives the Fault Detection Counter information of the requested EventId. If the return value of the function call is other than OK this parameter does not contain valid data.
Return Value	success of operation	
	E_OK	request was successful
	E_NOT_OK	request failed
	DEM_E_NO_FDC_AVAILABLE	there is no fault detection counter available for the requested event
Description	This API can only be used through the RTE, and therefore no declaration is exported via Dem.h. The API shall be used by SW-C to request the current Fault Detection Counter for a given EventID.	

5.3.3.4.41. Dem_GetFunctionalUnitOfDTC

Purpose	Gets the functional unit of the selected DTC.	
Synopsis	Std_ReturnType Dem_GetFunctionalUnitOfDTC (uint8 ClientId , uint8 * DTCTFunctionalUnit);	
Service ID	DEM_SID_GetFunctionalUnitOfDTC	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
Parameters (out)	DTCTFunctionalUnit	This parameter contains the Functional unit value of the DTC
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	The DTC functional unit is correctly provided in the DTCTFunctionalUnit parameter
	E_NOT_OK	A development error was detected.



	DEM_WRONG_DTC	Dem_SelectDTC() was called to select a DTC group or a single DTC in OBD format or no DTC matching the DTC selection criteria could be found.
	DEM_WRONG_DTCORIGIN	Selected DTCOrigin does not exist.
Description	<p>This API returns the functional unit of a DTC configured via DemDTCFunctionalUnit. The target DTC whose functional unit is required must be selected first by calling Dem_SelectDTC().</p> <p>Note: This API must only be used by the Dcm module!</p>	

5.3.3.4.42. Dem_GetIndicatorStatus

Purpose	Reads the indicator-status.					
Synopsis	<pre>Std_ReturnType Dem_GetIndicatorStatus (Dem_IndicatorIdType IndicatorId , Dem_IndicatorStatusType * IndicatorStatus);</pre>					
Service ID	DEM_SID_GetIndicatorStatus					
Sync/Async	Synchronous					
Reentrancy	Non reentrant					
Parameters (in)	IndicatorId	Number of indicator				
Parameters (out)	IndicatorStatus	Status of the indicator, like on, off, blinking.				
Return Value	<p>success of operation</p> <table border="1"> <tr> <td>E_OK</td> <td>Operation was successful</td> </tr> <tr> <td>E_NOT_OK</td> <td>Operation failed or is not supported</td> </tr> </table>		E_OK	Operation was successful	E_NOT_OK	Operation failed or is not supported
E_OK	Operation was successful					
E_NOT_OK	Operation failed or is not supported					
Description	<p>This function shall be used to read the indicator-status derived from the event status as a summary of all assigned events.</p> <p>Configuration: The assignment for the Dem_IndicatorId to indicator has to be done.</p> <p>Examples for indicators: lamps, different text messages, icons, ...</p> <p>Note: This API is not available, if no indicators are configured.</p>					

5.3.3.4.43. Dem_GetNextExtendedDataRecord

Purpose	Gets the data of the selected extended data record.
----------------	---



Synopsis	<code>Std_ReturnType Dem_GetNextExtendedDataRecord (uint8 ClientId , uint8 * DestBuffer , uint16 * BufSize);</code>	
Service ID	DEM_SID_GetNextExtendedDataRecord	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
Parameters (in,out)	BufSize	When the function is called, this parameter contains the maximum number of data bytes that can be written to the buffer. The function returns the actual number of written data bytes in this parameter.
Parameters (out)	DestBuffer	This parameter contains a byte pointer that points to the buffer, to which the extended data record shall be written. The format is: {ExtendedDataRecordNumber, data[0], data[1], ..., data[n]}
Return Value	Status of the operation to retrieve extended data by DTC. E_OK E_NOT_OK DEM_BUFFER_TOO_SMALL DEM_NO_SUCH_ELEMENT	
Description	The API copies the data of the (next) extended data record selected by Dem_SelectExtendedDataRecord() to the provided destination buffer. Note: This API must only be used by the Dcm module!	

5.3.3.4.44. Dem_GetNextFilteredDTC

Purpose	Gets the next filtered DTC.
Synopsis	<code>Std_ReturnType Dem_GetNextFilteredDTC (uint8 ClientId , uint32 * DTC , Dem_DTCStatusMaskType * DTCStatus);</code>
Service ID	DEM_SID_GetNextFilteredDTC
Sync/Async	Synchronous
Reentrancy	Non reentrant



Parameters (in)	ClientId	client ID for which the next filtered DTC will be retrieved.
Parameters (out)	DTC	Receives the DTC value in respective format of the filter returned by this function. If the return value of the function is other than E_OK this parameter does not contain valid data
	DTCStatus	This parameter receives the status information of the requested DTC. If the return value of the function call is other than OK this parameter does not contain valid data.
Return Value	Status of the operation to retrieve a DTC from the Dem. The value DEM_PENDING is never returned (because of synchronous behavior).	
	E_OK	Returned next filtered element
	E_NOT_OK	No DTC filter set
	DEM_NO SUCH ELEMENT	No further element matching the filter criteria found.
Description	<p>The API shall be used to return the current DTC and its associated status from the Dem matching the filter criteria defined by the API call Dem_SetDTCFilter(). After having returned the data the function skips to the next DTC matching the filter criteria.</p> <p>To receive all DTCs matching the filter criteria this function shall called continuously until the return value of the function is "NoMatchingDTC".</p> <p>The chronological order shall be reported if the DTC status mask parameter is set to "pending" and/or "confirmed" (no other status bits are allowed to be set). The function shall start with the most recent DTC. The chronological order may vary with the customer specific attributes used by the algorithm for sorting the DTC records (e.g. pre-sorted records or time-stamp attributes of the records).</p> <p>Note: This API must only be used by the Dcm module!</p>	

5.3.3.4.45. Dem_GetNextFilteredDTCAndFDC

Purpose	returns next filtered fault detection counter
Synopsis	Std_ReturnType Dem_GetNextFilteredDTCAndFDC (uint8 ClientId , uint32 * DTC , sint8 * DTCFaultDetectionCounter);
Service ID	DEM_SID_GetNextFilteredDTCAndFDC



Sync/Async	Asynchronous	
Reentrancy	Non reentrant	
Parameters (in)	ClientID	client ID for which the next filtered DTC and FDC will be retrieved.
Parameters (out)	DTC	Receives the DTC value in respective format of the filter returned by this function. If the return value of the function is other than E_OK this parameter does not contain valid data.
	DTCFaultDetectionCounter	This parameter receives the Fault Detection Counter information of the requested DTC. If the return value of the function call is other than OK this parameter does not contain valid data.
Return Value	Status of the operation to retrieve a DTC from the Dem. The value DEM_PENDING is never returned (because of synchronous behavior).	
	E_OK	Returned next filtered element
	E_NOT_OK	No DTC filter set
	DEM_NO SUCH ELEMENT	No further element matching the filter criteria found.
Description	<p>The API shall be used to return the current DTC and its associated Fault Detection Counter (FDC)from the Dem, matching the filter criteria defined by the API call Dem_SetDTCFilter. After having returned the data the function skips to the next DTC matching the filter criteria. To receive all DTCs matching the filter criteria this function shall be called continuously until the return value of the function is "NoMatchingDTC".The interface has an asynchronous behavior, because the FDC might be received asynchronously from a SW-C, too.</p> <p>Note: This API must only be used by the Dcm module!</p>	

5.3.3.4.46. Dem_GetNextFilteredDTCAndSeverity

Purpose	Gets the current DTC and its Severity from the Dem.
Synopsis	Std_ReturnType Dem_GetNextFilteredDTCAndSeverity (uint8 ClientID , uint32 * DTC , uint8 * DTCStatus , Dem_DTCSeverityType * DTCSeverity , uint8 * DTCFunctionalUnit);
Service ID	DEM_SID_GetNextFilteredDTCAndSeverity
Sync/Async	Synchronous



Reentrancy	Non reentrant		
Parameters (in)	ClientId		client ID for which the next filtered DTC and Severity will be retrieved.
Parameters (out)	DTC		Receives the DTC value in respective format of the filter returned by this function. If the return value of the function is other than E_OK this parameter does not contain valid data.
	DTCStatus		Receives the status value returned by the function. If the return value is other than E_OK this parameter does not contain valid data.
	DTCSeverity		Receives the severity value returned by the function. If the return value is other than E_OK this parameter does not contain valid data.
	DTFunctionalUnit		Receives the functional unit value returned by the function. If the return value is other than E_OK this parameter does not contain valid data.
Return Value	Status of the operation to retrieve a DTC from the Dem. The value DEM_PENDING is never returned (because of synchronous behavior).		
	E_OK	Returned next filtered element	
	E_NOT_OK	No DTC filter set	
	DEM_NO SUCH ELEMENT	No further element matching the filter criteria found.	
Description	<p>The API shall return the current DTC and its associated Fault Severity from the Dem, matching the filter criteria defined by the function call Dem_SetDTCFilter().</p> <p>Note: This API must only be used by the Dcm module!</p>		

5.3.3.4.47. Dem_GetNextFilteredRecord

Purpose	Gets the next freeze frame record number and its associated DTC stored in the event memory.
Synopsis	Std_ReturnType Dem_GetNextFilteredRecord (uint8 ClientId , uint32 * DTC , uint8 * RecordNumber);
Service ID	DEM_SID_GetNextFilteredRecord



Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
Parameters (out)	DTC	Receives the DTC value in respective format of the filter returned by this function. If the return value of the function is other than E_OK this parameter does not contain valid data.
	RecordNumber	Freeze frame record number of the reported DTC (relative addressing). If the return value of the function is other than E_OK this parameter does not contain valid data.
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	Returned next filtered element.
	E_NOT_OK	Indicates a development error.
	DEM_NO SUCH ELEMENT	No further element (matching the filter criteria) found
	<p>The API shall be used to get the next freeze frame record number and its associated DTC stored in the event memory. After having returned the data the function skips to the next record matching the filter criteria. To receive all records matching the filter criteria this function shall called continuously until the return value of the function is "DEM_NO SUCH ELEMENT".</p> <p>Note: This API must only be used by the Dcm module!</p>	

5.3.3.4.48. Dem_GetNextFreezeFrameData

Purpose	Gets the data of the selected freeze frame record.	
Synopsis	<pre>Std_ReturnType Dem_GetNextFreezeFrameData (uint8 ClientId , uint8 * DestBuffer , uint16 * BufSize);</pre>	
Service ID	DEM_SID_GetNextFreezeFrameData	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.



Parameters (in,out)	BufSize	When the function is called, this parameter contains the maximum number of data bytes that can be written to the buffer. The function returns the actual number of written data bytes in this parameter.
Parameters (out)	DestBuffer	This parameter contains a byte pointer that points to the buffer to which the freeze frame data shall be written. The format is: {RecordNumber, NumOfDIDs, DID[1], data[1], ..., DID[N], data[N]}.
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	Size and buffer successfully returned.
	DEM_BUFFER_TOO_SMALL	Provided buffer size too small.
	E_NOT_OK	Selection function is not called or a development error is detected.
	DEM_NO_SUCH_ELEMENT	Found no (further) element matching the filter criteria.
Description	This API copies the data of the (next) freeze frame record selected by Dem_SelectFreezeFrameData() to the provided destination buffer. Note: This API must only be used by the Dcm module!	

5.3.3.4.49. Dem_GetNumberOfEventMemoryEntries

Purpose	Gets the number of entries currently stored in the event memory.	
Synopsis	Std_ReturnType Dem_GetNumberOfEventMemoryEntries (Dem_DTCOriginType DTCOrigin , uint8 * NumberOfEventMemoryEntries);	
Service ID	DEM_SID_GetNumberOfEventMemoryEntries	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	DTCOrigin	If the Dem supports more than one event memory this parameter is used to select the source memory the number of entries shall be read from.
Parameters (out)	NumberOfEventMemoryEntries	This parameter returns the number of entries currently stored in the requested event memory.



Return Value	success of operation	
	E_OK	Operation was successful
	E_NOT_OK	Operation failed or is not supported

5.3.3.4.50. Dem_GetNumberOfFilteredDTC

Purpose	Gets the number of a filtered DTC.	
Synopsis	<code>Std_ReturnType Dem_GetNumberOfFilteredDTC (uint8 ClientId , uint16 * NumberOfFilteredDTC);</code>	
Service ID	DEM_SID_GetNumberOfFilteredDTC	
Sync/Async	Asynchronous	
Reentrancy	Non reentrant	
Parameters (in)	ClientId	client ID for which the number of filtered DTCs will be retrieved.
Parameters (out)	NumberOfFilteredDTC	The number of DTCs matching the defined status mask.
Return Value	Status of the operation to retrieve a number of DTC from the Dem. The value DEM_PENDING is never returned (because of synchronous behavior).	
	E_OK	Getting number of filtered DTCs was successful
	E_NOT_OK	No DTC filter set
Description	<p>The API shall be used to get the number of DTCs matching the defined status mask. The DTC Status mask filter is set by the API Dem_SetDTCFilter().</p> <p>Caveats: DTC filter has been set up properly before function call (Dem_SetDTCFilter).</p> <p>Note: This API must only be used by the Dcm module!</p>	

5.3.3.4.51. Dem_GetPfcCycleQualified

Purpose	Returns TRUE if the criteria for the PFC cycle have been met during the current OBD driving cycle. API is needed in OBD-relevant ECUs only.
Synopsis	<code>Std_ReturnType Dem_GetPfcCycleQualified (boolean * isqualified) ;</code>
Service ID	DEM_SID_GetPfcCycleQualified
Sync/Async	Synchronous



Reentrancy	Non Reentrant	
Parameters (out)	isqualified	TRUE: During the current OBD driving cycle the criteria for the PFC cycle have been met. FALSE: During the current OBD driving cycle the criteria for the PFC cycle have not been met or permanent memory is not defined.
Return Value	success of operation	
	E_OK	always E_OK is returned
	E_NOT_OK	E_NOT_OK will never appear

5.3.3.4.52. Dem_GetSI30Status

Purpose	Gets the SI30 Status by event.	
Synopsis	<code>Std_ReturnType Dem_GetSI30Status (Dem_EventIdType EventId , uint8 * Status);</code>	
Service ID	DEM_SID_GetSI30Status	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
Parameters (out)	Status	Pointer to variable which contains the SI30 status of the event
Return Value	success of operation	
	E_OK	Operation was successful
	E_NOT_OK	Operation failed

5.3.3.4.53. Dem_GetSeverityOfDTC

Purpose	Gets the severity of the selected DTC.	
Synopsis	<code>Std_ReturnType Dem_GetSeverityOfDTC (uint8 ClientId , Dem_DTCSeverityType * DTCSeverity);</code>	
Service ID	DEM_SID_GetSeverityOfDTC	
Sync/Async	Synchronous	



Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
Parameters (out)	DTCSeverity	This parameter contains the DTCSeverity-Mask according to ISO14229-1.
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	The DTC severity is correctly provided in the DTCSeverity parameter.
	E_NOT_OK	A development error was detected.
	DEM_WRONG_DTC	Dem_SelectDTC() was called to select a DTC group or a single DTC in OBD format or no DTC matching the DTC selection criteria could be found.
	DEM_WRONG_DTCORIGIN	Selected DTCOrigin does not exist.
Description	This API returns the severity of a DTC configured via DemDTCSeverity. The target DTC whose severity is required must be selected first by calling Dem_SelectDTC() .	

5.3.3.4.54. Dem_GetSizeOfExtendedDataRecordSelection

Purpose	Gets the size of the selected extended data record(s).	
Synopsis	Std_ReturnType DemGetSizeOfExtendedDataRecordSelection (uint8 ClientId , uint16 * SizeOfExtendedDataRecord);	
Service ID	DEM_SID_GetSizeOfExtendedDataRecordSelection	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
Parameters (out)	SizeOfExtendedDataRecord	Size of the requested extended data record(s) including record number. The format for a single ExtendedDataRecord is: {RecordNumber, data[1], ..., data[N]}
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	Size returned successfully.
	E_NOT_OK	Selection function is not called or a development error is detected.



	DEM_NO_SUCH_ELEMENT	Record number is not supported by configuration and therefore invalid.
Description	<p>The API returns the size of the extended data record(s) selected by Dem_SelectExtendedDataRecord(), which represents the number of user data bytes stored in the extended data record, including any extended data record header information i.e. the extended data record number.</p> <p>Note: This API must only be used by the Dcm module!</p>	

5.3.3.4.55. Dem.GetSizeOfFreezeFrameSelection

Purpose	Gets the size of the selected freeze frame record(s).	
Synopsis	<pre>Std_ReturnType Dem.GetSizeOfFreezeFrameSelection (uint8 ClientId , uint16 * SizeOfFreezeFrame);</pre>	
Service ID	DEM_SID_GetSizeOfFreezeFrameSelection	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
Parameters (out)	SizeOfFreezeFrame	Number of bytes in the requested freeze frame.
Return Value	Status of the operation of type Std_ReturnType E_OK E_NOT_OK DEM_NO_SUCH_ELEMENT	
Description	<p>The API returns the size of the freeze frame record(s) selected by Dem_SelectFreezeFrameData(). This size includes the number of user data bytes (pure freeze frame data) as well as the number of bytes in the header (record number, number of DIDs and the individual DIDs).</p> <p>Note: This API must only be used by the Dcm module!</p>	

5.3.3.4.56. Dem_GetStatusOfDTC

Purpose	Gets the status of a DTC.
----------------	---------------------------



Synopsis	<code>Std_ReturnType Dem_GetStatusOfDTC (uint8 ClientId , Dem_DTCStatusMaskType * DTCStatus);</code>	
Service ID	DEM_SID_GetStatusOfDTC	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module
Parameters (out)	DTCStatus	This parameter receives the status information of the requested DTC. It follows the format as defined in ISO14229-1. If the return value of the function call is other than E_OK, this parameter does not contain valid data.
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	Status of the DTC is correctly provided in the DTCStatus parameter
	E_NOT_OK	No DTC selected
	DEM_WRONG_DTC	Selected DTC value in selected format does not exist
	DEM_WRONG_DTCORIGIN	Selected DTCOrigin does not exist
	<p>The DTCs of OBD Events Suppression shall be reported as DEM_WRONG_DTC.</p> <p>This API shall be used to read the status of a DTC to the parameter DTCStatus according to ISO14229.</p> <p>Note: This API must only be used by the Dcm module!</p>	

5.3.3.4.57. Dem_GetTranslationType

Purpose	Gets the supported DTC formats of the ECU.	
Synopsis	<code>Dem_DTCTranslationFormatType Dem_GetTranslationType (uint8 ClientId);</code>	
Service ID	DEM_SID_GetTranslationType	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	ClientId	client ID for which the translation type will be retrieved.



Return Value	The Translation format provides the configured translation formats according to ISO 14229-1 service 0x19
Description	<p>The supported formats are configured via DemTypeOfDTCSupported.</p> <p>Note: This API must only be used by the Dcm module!</p>

5.3.3.4.58. Dem_GetVersionInfo

Purpose	This service returns the version information of this module.	
Synopsis	<code>void Dem_GetVersionInfo (Std_VersionInfoType * versioninfo);</code>	
Service ID	DEM_SID_GetVersionInfo	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	versioninfo	Pointer to where to store the version information of this module.
Description	<p>The version information includes:</p> <ul style="list-style-type: none"> ➤ Module Id ➤ Vendor Id ➤ Vendor specific version numbers (BSW00407). <p>This function shall be pre compile time configurable On/Off by the configuration parameter: DEM_VERSION_INFO_API</p>	

5.3.3.4.59. Dem_Init

Purpose	Initializes or reinitializes the Dem module.
Synopsis	<code>void Dem_Init (void);</code>
Service ID	DEM_SID_Init
Sync/Async	Synchronous
Reentrancy	Non reentrant
Description	<p>This function shall be used during the startup phase of the ECU after the NVRAM Manager has finished the restore of NVRAM data.</p> <p>SW-Components including Monitor Functions are initialized afterwards.</p>



Caveats: The Dem is not functional until this function has been called.

5.3.3.4.60. Dem_J1939DcmClearDTC

Purpose	Clears active DTCs as well as previously active DTCs.	
Synopsis	<code>Dem_ReturnClearDTCType Dem_J1939DcmClearDTC (Dem_-J1939DcmSetClearFilterType DTCTypeFilter , uint8 node);</code>	
Parameters (in)	DTCTypeFilter	The following types are available: DEM_J1939DTC_CLEAR_ALL DEM_J1939DTC_CLEAR_PREVIOUSLY_ACTIVE
	node	Nm node Id of requesting client
Return Value	Status of the operation of type Dem_ReturnClearDTCType.	
DEM_CLEAR_OK		DTC successfully cleared
DEM_ASR42_CLEAR_FAILED		DTC clearing failed
DEM_CLEAR_PENDING		The DTC clearing is performed asynchronously and still pending. The caller can retry later.
DEM_CLEAR_BUSY		DTC not cleared, as another clearing process is in progress. The caller can retry later.
DEM_CLEAR_MEMORY_ERROR		An error occurred during erasing a memory location

5.3.3.4.61. Dem_J1939DcmFirstDTCwithLampStatus

Purpose	Service sets the filter to the first applicable DTC for the DM31 response for a specific node.	
Synopsis	<code>void Dem_J1939DcmFirstDTCwithLampStatus (uint8 node);</code>	
Service ID	DEM_SID_J1939DcmFirstDTCwithLampStatus	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	node	Nm node Id of requesting client.
Description	Note: Functionality not implemented, only stub available.	



5.3.3.4.62. Dem_J1939DcmGetNextDTCwithLampStatus

Purpose	Service to report the next filtered J1939 DTC for DM31 including current LampStatus..		
Synopsis	<pre>Dem_ReturnGetNextFilteredElementType Dem_- J1939DcmGetNextDTCwithLampStatus (Dem_J1939DcmLampStatusType * LampStatus , uint32 * J1939DTC , uint8 * OccurenceCounter);</pre>		
Service ID	DEM_SID_J1939DcmGetNextDTCwithLampStatus		
Sync/Async	Synchronous		
Reentrancy	Non Reentrant		
Parameters (out)	LampStatus	Receives the lamp status returned by this function. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.	
	J1939DTC	Receives the J1939DTC value. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.	
	OccurenceCounter	This parameter receives the corresponding occurrence counter. If the return value of the function call is other than DEM_FILTERED_OK this parameter does not contain valid data.	
Return Value	DTC available or not		
	DEM_FILTERED_OK	DTC is found	
	DEM_FILTERED_NO_MATCHING_ELEMENT	No further element (matching the filter criteria) found	
Description	Note: Functionality not implemented, only stub available.		

5.3.3.4.63. Dem_J1939DcmGetNextFilteredDTC

Purpose	Gets the next filtered J1939 DTC.
Synopsis	<pre>Dem_ReturnGetNextFilteredElementType Dem_- J1939DcmGetNextFilteredDTC (uint32 * J1939DTC , uint8 * Oc- curenceCounter);</pre>
Service ID	DEM_SID_J1939DcmGetNextFilteredDTC
Sync/Async	Synchronous
Reentrancy	Non Reentrant



Parameters (out)	J1939DTC	The next J1939 DTC value that matches the filter. Data is only valid, if return value is DEM_FILTERED_OK.
	OccurrenceCounter	The occurrence counter of the filtered DTC. Data is only valid, if return value is DEM_FILTERED_OK.
Return Value	DTC available or not	
	DEM_FILTERED_OK	DTC is found
	DEM_FILTERED_NO_MATCHING_ELEMENT	No further element (matching the filter criteria) found

5.3.3.4.64. Dem_J1939DcmGetNextFilteredRatio

Purpose	Gets the next filtered J1939 Ratio.	
Synopsis	<pre>Dem_ReturnGetNextFilteredElementType Dem_- J1939DcmGetNextFilteredRatio (uint16 * SPN , uint16 * Numerator , uint16 * Denominator);</pre>	
Service ID	DEM_SID_J1939DcmGetNextFilteredRatio	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (out)	SPN	Receives the SPN of the applicable system monitor. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.
	Numerator	Receives the Numerator of the applicable system monitor. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.
	Denominator	Receives the Denominator of the applicable system monitor. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.
Return Value	Indicates the availability status of the ratio	
	DEM_FILTERED_OK	Ratio available in out parameter



	DEM_FILTERED_NO_MATCHING_ELEMENT	No further element available
Description	Note: Functionality not implemented, only stub available.	

5.3.3.4.65. Dem_J1939DcmGetNextFreezeFrame

Purpose	Gets the next filtered J1939 Freeze Frame data. The function stores the data in the provided DestBuffer.	
Synopsis	<pre>Dem_ReturnGetNextFilteredElementType Dem_- J1939DcmGetNextFreezeFrame (uint32 * J1939DTC , uint8 * Oc- currenceCounter , uint8 * DestBuffer , uint16 * BufSize);</pre>	
Service ID	DEM_SID_J1939DcmGetNextFreezeFrame	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in,out)	BufSize	The buffer size of the available buffer at the caller and will be filled with the size of the freeze frame data in case of success.
	DestBuffer	The buffer that will be filled with the freeze frame data.
Parameters (out)	J1939DTC	The next J1939 DTC value that matches the filter. Data is only valid, if return value is DEM_FILTERED_OK.
	OccurrenceCounter	The occurrence counter of the filtered DTC. Data is only valid, if return value is DEM_FILTERED_OK.
Return Value	Freeze Frame available or not	
	DEM_FILTERED_OK	Freeze Frame is found
	DEM_FILTERED_NO_MATCHING_ELEMENT	Freeze Frame is not available

5.3.3.4.66. Dem_J1939DcmGetNextSPNInFreezeFrame

Purpose	Gets the next SPN.
Synopsis	<pre>Dem_ReturnGetNextFilteredElementType Dem_- J1939DcmGetNextSPNInFreezeFrame (uint32 * SPNSupported , uint8 * SPNDataLength);</pre>



Service ID	DEM_SID_J1939DcmGetNextSPNInFreezeFrame	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (out)	SPNSupported	This parameter contains the next SPN in the ExpandedFreezeFrame
	SPNDataLength	This parameter contains the corresponding dataLength of the SPN.
Return Value	SPN available or not	
	DEM_FILTERED_OK	SPN is found
	DEM_FILTERED_NO_MATCHING_ELEMENT	SPN is not available
Description	Note: Functionality not implemented, only stub available.	

5.3.3.4.67. Dem_J1939DcmGetNumberOfFilteredDTC

Purpose	Gets the count of filtered J1939 DTC.	
Synopsis	<pre>Dem_ReturnGetNumberOfFilteredDTCType Dem_- J1939DcmGetNumberOfFilteredDTC (uint16 * NumberOfFilteredDTC) ;</pre>	
Service ID	DEM_SID_J1939DcmGetNumberOfFilteredDTC	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (out)	NumberOfFilteredDTC	The number of DTCs that have been filtered. The value is limited to 250, which is in accordance with the definition of diagnostic messages.
Return Value	Number of DTCs available or not	
	DEM_NUMBER_OK	Number of Filtered DTCs is available
	DEM_NUMBER_FAILED	Number of Filtered DTCs is not available

5.3.3.4.68. Dem_J1939DcmReadDiagnosticReadiness1

Purpose	Service to report the value of Diagnostic Readiness 1 (DM05) computed by the Dem.
----------------	---



Synopsis	<code>Std_ReturnType Dem_J1939DcmReadDiagnosticReadiness1 (Dem_J1939DcmDiagnosticReadiness1Type * DataValue , uint8 node);</code>	
Service ID	DEM_SID_J1939DcmReadDiagnosticReadiness1	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	node	Nm node Id of requesting client.
Parameters (out)	DataValue	Buffer of 8 bytes containing the contents of Diagnostic Readiness 1 (DM05) computed by the Dem.
Return Value	success of operation.	
	E_OK	Operation was successful
	E_NOT_OK	Operation failed

5.3.3.4.69. Dem_J1939DcmReadDiagnosticReadiness2

Purpose	Service to report the value of Diagnostic Readiness 2 (DM21) computed by the Dem.	
Synopsis	<code>Std_ReturnType Dem_J1939DcmReadDiagnosticReadiness2 (Dem_J1939DcmDiagnosticReadiness2Type * DataValue , uint8 node);</code>	
Service ID	DEM_SID_J1939DcmReadDiagnosticReadiness2	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	node	Nm node Id of requesting client.
Parameters (out)	DataValue	Buffer of 8 bytes containing the contents of Diagnostic Readiness 2 (DM21) computed by the Dem.
Return Value	success of operation.	
	E_OK	Operation was successful
	E_NOT_OK	Operation failed
Description	Note: Functionality not implemented, only stub available.	

5.3.3.4.70. Dem_J1939DcmReadDiagnosticReadiness3

Purpose	Service to report the value of Diagnostic Readiness 3 (DM26) computed by the Dem.
----------------	---



Synopsis	<code>Std_ReturnType Dem_J1939DcmReadDiagnosticReadiness3 (Dem_J1939DcmDiagnosticReadiness3Type * DataValue , uint8 node);</code>	
Service ID	DEM_SID_J1939DcmReadDiagnosticReadiness3	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	node	Nm node Id of requesting client.
Parameters (out)	DataValue	Buffer of 8 bytes containing the contents of Diagnostic Readiness 3 (DM26) computed by the Dem.
Return Value	success of operation.	
	E_OK	Operation was successful
	E_NOT_OK	Operation failed
Description	Note: Functionality not implemented, only stub available.	

5.3.3.4.71. Dem_J1939DcmSetDTCFilter

Purpose	The function sets the DTC filter for a specific node and returns the composite lamp status of the filtered DTCs.	
Synopsis	<code>Dem_ReturnSetFilterType Dem_J1939DcmSetDTCFilter (Dem_J1939DcmDTCStatusFilterType DTCStatusFilter , Dem_DTCKindType DTCKind , uint8 node , Dem_J1939DcmLampStatusType * LampStatus);</code>	
Service ID	DEM_SID_J1939DcmSetDTCFilter	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	DTCStatusFilter	The filter type to filter the DTCs.
	DTCKind	The kind of DTCs to filter.
	node	The requesting NM node.
Parameters (out)	LampStatus	The status of all the 4 lamps.
Return Value	Filter status available or not	
	DEM_FILTER_ACCEPTED	filter is accepted
	DEM_WRONG_FILTER	filter is not accepted (Filter settings that are not representing a diagnostic message, i.e. "DEM_J1939DTC_PERMA-



	NENT + DEM_DTC_KIND_ALL_DTCS" and "DEM_J1939DTC_CURRENTLY_ACTIVE + DEM_DTC_KIND_EMISSION_REL_DTCS" are not accepted and return DEM_WRONG_FILTER.)
--	---

5.3.3.4.72. Dem_J1939DcmSetFreezeFrameFilter

Purpose	Sets filter for the J1939 Freeze Frame.	
Synopsis	<pre>Dem_ReturnSetFilterType Dem_J1939DcmSetFreezeFrameFilter (Dem_J1939DcmSetFreezeFrameFilterType FreezeFrameKind , uint8 node);</pre>	
Service ID	DEM_SID_J1939DcmSetFreezeFrameFilter	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	FreezeFrameKind	The type of Freeze Frame to be filtered.
	node	The Nm node for which the Freeze Frames should be filtered.
Return Value	Filter status available or not	
	DEM_FILTER_ACCEPTED	filter is accepted
	DEM_WRONG_FILTER	filter is not accepted

5.3.3.4.73. Dem_J1939DcmSetRatioFilter

Purpose	The function sets the Ratio filter for a specific node and returns the corresponding Ignition Cycle Counter and General Denominator.	
Synopsis	<pre>Dem_ReturnSetFilterType Dem_J1939DcmSetRatioFilter (uint16 * IgnitionCycleCounter , uint16 * OBDMonitoringConditionsEncoun- tered , uint8 node);</pre>	
Service ID	DEM_SID_J1939DcmSetRatioFilter	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OBDMonitoringConditionsEncoun- tered	OBD Monitoring Conditions Encountered



	node	Nm node Id of requesting client.
Parameters (out)	IgnitionCycleCounter	Ignition Cycle Counter.
Return Value	Filter status available or not	
	DEM_FILTER_ACCEPTED	filter is accepted
	DEM_WRONG_FILTER	filter is not accepted
Description	Note: Functionality not implemented, only stub available.	

5.3.3.4.74. Dem_MainFunction

Purpose	Processes event memory entries from error-queue.
Synopsis	<code>void Dem_MainFunction (void);</code>
Service ID	DEM_SID_MainFunction
Sync/Async	Synchronous
Reentrancy	Non reentrant
Description	<p>This function is used to process all not event based Dem internal functions. It shall be called periodically as cyclic task by the software system (e.g. by operating system).</p> <p>Timing: fixed cyclic</p> <p>Configuration: The cyclic time for the main function has to be defined as an operating system task or runnable entity.</p>

5.3.3.4.75. Dem_NvMEventStatusBlockCallback

Purpose	Callback-Function from NvM for default data block notifying successful completion of single block request.	
Synopsis	<code>Std_ReturnType Dem_NvMEventStatusBlockCallback (uint8 ServiceId , NvM_RequestResultType JobResult);</code>	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	ServiceId	
	JobResult	
Return Value	returns always E_OK	



5.3.3.4.76. Dem_NvMGateEntryMirrorBlockCallback

Purpose	Callback-Function from NvM for mirror NV gate-entry data block notifying successful completion of single block request.	
Synopsis	<code>Std_ReturnType Dem_NvMGateEntryMirrorBlockCallback (uint8 ServiceId , NvM_RequestResultType JobResult);</code>	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	ServiceId	
	JobResult	
Return Value	returns always E_OK	

5.3.3.4.77. Dem_NvMGateEntryPrimaryBlockCallback

Purpose	Callback-Function from NvM for primary NV gate-entry data block notifying successful completion of single block request.	
Synopsis	<code>Std_ReturnType Dem_NvMGateEntryPrimaryBlockCallback (uint8 ServiceId , NvM_RequestResultType JobResult);</code>	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	ServiceId	
	JobResult	
Return Value	returns always E_OK	

5.3.3.4.78. Dem_NvMGateEntrySecondaryBlockCallback

Purpose	Callback-Function from NvM for secondary NV gate-entry data block notifying successful completion of single block request.	
Synopsis	<code>Std_ReturnType Dem_NvMGateEntrySecondaryBlockCallback (uint8 ServiceId , NvM_RequestResultType JobResult);</code>	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	ServiceId	
	JobResult	



Return Value	returns always E_OK
---------------------	---------------------

5.3.3.4.79. Dem_NvMReadCopyEventStatusBlock

Purpose	Callback-Function from NvM for default data block to request the data to be copied from the NvM RAM mirror during NvM_ReadBlock() or NvM_ReadAll().	
Synopsis	<code>Std_ReturnType Dem_NvMReadCopyEventStatusBlock (const void * NvMBuffer);</code>	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	NvMBuffer	Pointer to NvM RAM mirror
Return Value	returns always E_OK	

5.3.3.4.80. Dem_NvMReadCopyPermanentMemory

Purpose	Block specific callback routine which is called in order to let the Dem copy Dem_EventMemoryPermanent[] array data from NvM block to RAM array.	
Synopsis	<code>Std_ReturnType Dem_NvMReadCopyPermanentMemory (const void * NvMBuffer);</code>	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	NvMBuffer	
Return Value	returns E_OK on success otherwise E_NOT_OK	

5.3.3.4.81. Dem_NvMWriteCopyEventStatusBlock

Purpose	Callback-Function from NvM for default data block to request the data to be copied into the NvM RAM mirror during NvM_WriteBlock() or NvM_WriteAll().	
Synopsis	<code>Std_ReturnType Dem_NvMWriteCopyEventStatusBlock (void * NvMBuffer);</code>	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	NvMBuffer	Pointer to NvM RAM mirror



Return Value	returns always E_OK
---------------------	---------------------

5.3.3.4.82. Dem_NvMWriteCopyPermanentMemory

Purpose	Block specific callback routine which is called in order to let the Dem copy Dem_EventMemoryPermanent[] array data from RAM block to NvM permanent memory block.	
Synopsis	<code>Std_ReturnType Dem_NvMWriteCopyPermanentMemory (void * NvM-Buffer);</code>	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (out)	NvMBuffer	
Return Value	returns E_OK on success otherwise E_NOT_OK	

5.3.3.4.83. Dem_NvMWriteFinishedPermanentMemory

Purpose	Callback-Function from NvM for permanent NV data block notifying successful completion of single block request. Function is provided only if permanent event memory is enabled. If JobResult is NVM_REQ_OK, the Dem_EventMemoryPermanentNonVolatileStored[] array is updated considering the content of Dem_EventMemoryPermanent[] array.	
Synopsis	<code>Std_ReturnType Dem_NvMWriteFinishedPermanentMemory (uint8 ServiceId , NvM_RequestResultType JobResult);</code>	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	ServiceId	
	JobResult	
Return Value	returns E_OK according to NvM requirement NVM368 return E_NOT_OK, in case DET reporting and BSW Distribution are enabled and the callback function is called from another context apart from the master core	

5.3.3.4.84. Dem_Prelinit

Purpose	PreInit function to make the Dem queue operational.
----------------	---



Synopsis	<code>void Dem_PreInit (const Dem_ConfigType * ConfigPtr);</code>	
Service ID	DEM_SID_PreInit	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	ConfigPtr	refer to Dem_ConfigType
Description	<p>This function shall be used to initialize the internal states necessary to process events reported by BSWs</p> <p>Dem_PreInit() shall be called by the ECU State Manager during the startup phase of the ECU before the NVRAM Manager has finished the restore of NVRAM data.</p>	

5.3.3.4.85. Dem_PrestoreFreezeFrame

Purpose	Captures the freeze frame data for a specific event.					
Synopsis	<code>Std_ReturnType Dem_PrestoreFreezeFrame (Dem_EventIdType EventId);</code>					
Service ID	DEM_SID_PrestoreFreezeFrame					
Sync/Async	Synchronous					
Reentrancy	Reentrant					
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.				
Return Value	<p>success of operation</p> <table> <tr> <td>E_OK</td> <td>PreStoreFreezeFrame was successful</td> </tr> <tr> <td>E_NOT_OK</td> <td>PreStoreFreezeFrame failed</td> </tr> </table>		E_OK	PreStoreFreezeFrame was successful	E_NOT_OK	PreStoreFreezeFrame failed
E_OK	PreStoreFreezeFrame was successful					
E_NOT_OK	PreStoreFreezeFrame failed					
Description	<p>This API can only be used through the RTE and therefore no declaration is exported via Dem.h.</p> <p>If the Dem does not receive any request to pre-store a freeze frame, freeze frame capture is linked to the API call Dem_SetEventStatus(). This API call triggers the freeze frame storage.</p> <p>If Dem_SetEventStatus(EventId, Passed) is called the corresponding pre-stored freeze frame is discarded (same behavior like Dem_ClearPrestoredFreezeFrame()). The API call Dem_ResetEventStatus() does not influence the pre-stored freeze frame.</p> <p>API is called from Monitor Function.</p>					



	<p>Caveats: Dem configuration during integration of Monitor Functions is system specific.</p> <p>Configuration: While configuring the Dem the capability of pre-store functionality for the required event has to be defined.</p> <p>Note: Optional development freeze frames are not supported by this API.</p>
--	--

5.3.3.4.86. Dem_ReplUMPRDenLock

Purpose	Service to lock the denominator of a specific monitor.	
Synopsis	Std_ReturnType Dem_ReplUMPRDenLock (Dem_RatioIDType RatioID);	
Service ID	DEM_SID_ReplUMPRDenLock	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	RatioID	Ratio Identifier reporting that specific denominator is locked (for physical reasons e.g. temperature conditions or minimum activity)
Return Value	success of operation	
	E_OK	IUMPR denominator status was successfully reported
	E_NOT_OK	IUMPR denominator status was not successfully reported
Description	Note: Functionality not implemented, only stub available.	

5.3.3.4.87. Dem_ReplUMPRDenRelease

Purpose	Service to release the denominator of a specific monitor.	
Synopsis	Std_ReturnType Dem_ReplUMPRDenRelease (Dem_RatioIDType RatioID);	
Service ID	DEM_SID_ReplUMPRDenRelease	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	RatioID	Ratio Identifier reporting that specific denominator is released (for physical rea-



		sons e.g. temperature conditions or minimum activity)
Return Value	success of operation	
	E_OK	IUMPR denominator status was successfully reported
	E_NOT_OK	IUMPR denominator status was not successfully reported
Description	Note: Functionality not implemented, only stub available.	

5.3.3.4.88. Dem_RepIUMPRFaultDetect

Purpose	Service for reporting that faults are possibly found because all conditions are fulfilled.	
Synopsis	Std_ReturnType Dem_RepIUMPRFaultDetect (Dem_RatioIdType RatioID);	
Service ID	DEM_SID_ReportIUMPRFaultDetect	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	RatioID	Ratio Identifier reporting that a respective diagnostic function could have found a fault -only used when interface option "API" is selected.
Return Value	success of operation	
	E_OK	report of IUMPR was successfully reported
	E_NOT_OK	report of IUMPR was not reported
Description	Note: Functionality not implemented, only stub available.	

5.3.3.4.89. Dem_ReportErrorStatus

Purpose	Function to report the status of diagnostic events of BSW.
Synopsis	void Dem_ReportErrorStatus (Dem_EventIdType EventId , Dem_EventStatusType EventStatus);
Service ID	DEM_SID_ReportErrorStatus



Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
	EventStatus	Monitor test result
Description	<p>Interface for BSW Components to report Errors during start up (even before Dem initialization) and normal operation. At a first step, it is assumed, that all incoming results are considered as debounced. If a central pre-debouncing is provided, this API shall be used to support them for the BSW.</p> <p>In case DEM_BSW_DISTRIBUTION_FOR_BSW_EVENT_REPORTING is enabled, this API supports calls from the configured Dem satellite instances.</p> <p>Configurations: The size of the buffer queue needs to be configured (ref. to DEM_BSW_ERROR_BUFFER_SIZE)</p>	

5.3.3.4.90. Dem_ResetEventDebounceStatus

Purpose	Reset/freeze counter, time, frequency-based event debouncing.	
Synopsis	<pre>Std_ReturnType Dem_ResetEventDebounceStatus (Dem_EventIdType EventId , Dem_DebounceResetStatusType DebounceResetStatus);</pre>	
Service ID	DEM_SID_ResetEventDebounceStatus	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId. The Event Number is configured in the DEM. Min.: 1 (0: Indication of no Event or Failure) Max.:Result of configuration of Event Numbers in DEM (Max is either 255 or 65535). **
	DebounceResetStatus	Identification of freeze or reset behavior of the internal debounce counter/timer/frequency.
Return Value	<p>success of operation</p> <p>E_OK</p>	
	Freeze/reset of event debounce counter/timer/frequency was successful.	



	E_NOT_OK	Only on development error, Freeze/reset was not successful.
--	----------	---

5.3.3.4.91. Dem_ResetEventStatus

Purpose	Resets the event failed status.	
Synopsis	<code>Std_ReturnType Dem_ResetEventStatus (Dem_EventIdType EventId) ;</code>	
Service ID	DEM_SID_ResetEventStatus	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
Return Value	success of operation	
	E_OK	reset of event status was successful
	E_NOT_OK	reset of event status failed
Description	<p>This API can only be used through the RTE and therefore no declaration is exported via Dem.h. This service shall be used to reset the Event Status stored in the Event Memory in the Dem, without the usage of API Dem_SetEventStatus(EventId, Passed), because no new test result is available at this time.</p> <p>With this API the status bit "Failed" defined by Dem_EventStatusExtendedType is set to 0. API is called by Monitor Function.</p> <p>Refer to ISO14229: DTC Status Bit Definition, Table D.14, Bit0 Test failed.</p> <p>Dem_ResetEventStatus does not influence the status bit 6 ("testNotCompletedThisMonitoringCycle")</p> <p>Caveats: Dem configuration during integration of Monitor Functions is system specific</p>	

5.3.3.4.92. Dem_ResetReadiness

Purpose	Sets the TNCSLC bit for all not confirmed (CDTC==0) and not pending (PDTC==0) events assigned to any ReadinessGroup reported by PID\$01.
Synopsis	<code>void Dem_ResetReadiness (void) ;</code>



Service ID	DEM_SID_ResetReadiness
Sync/Async	Synchronous
Reentrancy	Non Reentrant

5.3.3.4.93. Dem_SelectDTC

Purpose	Selects a DTC or DTC group as target for further operations.	
Synopsis	<code>Std_ReturnType Dem_SelectDTC (uint8 ClientId , uint32 DTC , Dem_DTCFormatType DTCFormat , Dem_DTCOriginType DTCOrigin);</code>	
Service ID	DEM_SID_SelectDTC	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
	DTC	Defines the DTC in respective format that is selected. If the DTC fits to a DTC group number, the DTC group is selected.
	DTCFormat	Defines the input-format of the provided DTC value.
	DTCOrigin	The event memory of the requested DTC or group of DTC.
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	DTC selection successfully requested.
	E_NOT_OK	DTC selection request failed.
Description	This API shall be called before the following operations: Dem_ClearDTC() , Dem_DisableDTCRecordUpdate() , Dem_EnableDTCRecordUpdate() , Dem_GetDTCSelectionResult() , Dem_GetFunctionalUnitOfDTC() , Dem_GetSeverityOfDTC() , Dem_GetStatusOfDTC() , Dem_SelectFreezeFrameData() , Dem_SelectExtendedDataRecord() and Dem_GetDTCSelectionResultForClearDTC() .	

5.3.3.4.94. Dem_SelectExtendedDataRecord

Purpose	Triggers the Dem internal Extended Data Record selection process on the event memory assigned to the ClientId.
----------------	--



Synopsis	<code>Std_ReturnType Dem_SelectExtendedDataRecord (uint8 ClientId , uint8 ExtendedDataNumber);</code>	
Service ID	DEM_SID_SelectExtendedDataRecord	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
	ExtendedDataNumber	Identification/Number of requested extended data record. Additionally the values 0xFE and 0xFF are explicitly allowed to request the overall size of all OBD records / all records.
Return Value	Status of the operation to select the extended data record for the given client.	
	E_OK	Extended data record successfully selected.
	E_NOT_OK	A development error was detected.
	DEM_WRONG_DTC	<code>Dem_SelectDTC()</code> was called to select a DTC group or a single DTC in OBD format or no DTC matching the DTC selection criteria was found.
	DEM_WRONG_DTCORIGIN	Selected DTCOrigin does not exist.
Description	<p>This API sets the filter to be used by Dem_GetNextExtendedDataRecord() and Dem_GetSizeOfExtendedDataRecordSelection(). The target DTC whose extended data record is to be read must be selected first by calling Dem_SelectDTC().</p> <p>Note: This API must only be used by the Dcm module!</p>	

5.3.3.4.95. Dem_SelectFreezeFrameData

Purpose	Triggers the Dem internal Freeze Frame selection process on the event memory assigned to the ClientId.
Synopsis	<code>Std_ReturnType Dem_SelectFreezeFrameData (uint8 ClientId , uint8 RecordNumber);</code>
Service ID	DEM_SID_SelectFreezeFrameData
Sync/Async	Synchronous
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.



Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
	RecordNumber	Unique identifier for a snapshot record as defined in ISO 14229-1. The value 0xFF is a placeholder referencing all snapshot records of the addressed DTC. The value 0x00 indicates the DTC-specific WWHOBD snapshot record.
Return Value	Status of the operation to select the freeze frame record for the given client.	
	E_OK	Freeze frame data successfully selected.
	DEM_WRONG_DTC	Dem_SelectDTC() was called to select a DTC group or a single DTC in OBD format or no DTC matching the DTC selection criteria was found.
	DEM_WRONG_DTCORIGIN	Selected DTCOrigin does not exist.
	E_NOT_OK	A development error was detected.
Description	<p>This API sets the filter to be used by Dem_GetNextFreezeFrameData() and Dem_GetSizeOfFreezeFrameSelection(). The target DTC whose freeze frame is to be read must be selected first by calling Dem_SelectDTC().</p> <p>Note: This API must only be used by the Dcm module!</p>	

5.3.3.4.96. Dem_SetAgingCycleCounterValue

Purpose	Provides the value of the external aging cycle counter.	
Synopsis	Std_ReturnType Dem_SetAgingCycleCounterValue (uint8 CounterValue);	
Service ID	DEM_SID_SetAgingCycleCounterValue	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CounterValue	Current external aging cycle counter value.
Return Value	success of operation	
	E_OK	set of aging cycle counter was successful
	E_NOT_OK	set of aging cycle counter failed
Description	This API can only be used through the RTE, and therefore no declaration is exported via Dem.h.	



	Note: Functionality not implemented, only stub available.
--	---

5.3.3.4.97. Dem_SetAgingCycleState

Purpose	Triggers the next aging cycle state.	
Synopsis	<code>Std_ReturnType Dem_SetAgingCycleState (uint8 AgingCycleId);</code>	
Service ID	DEM_SID_SetAgingCycleState	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	AgingCycleId	Identification of aging cycle.
Return Value	success of operation E_OK set of aging cycle was successful E_NOT_OK set of aging cycle failed	
Description	This API can only be used through the RTE, and therefore no declaration is exported via Dem.h.	

5.3.3.4.98. Dem_SetCycleQualified

Purpose	Sets an operation cycle as qualified.	
Synopsis	<code>Std_ReturnType Dem_SetCycleQualified (Dem_OperationCycleIdType OperationCycleId);</code>	
Service ID	DEM_SID_SetCycleQualified	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	OperationCycleId	Identification of operation cycle Note: Only driving cycle is allowed!
Return Value	success of operation E_OK set operation cycle as qualified has succeeded E_NOT_OK operation cycle could not be set as qualified	
Description	Dem shall be called by the SW-Component as soon as it detects that the operation cycle is qualified.	



5.3.3.4.99. Dem_SetDTCFilter

Purpose	Allows filtering for DTCs with severity information.	
Synopsis	<pre>Std_ReturnType Dem_SetDTCFilter (uint8 ClientId , uint8 DTCS- statusMask , Dem_DTCFormatType DTCFormat , Dem_DTCOriginType DTCOrigin , boolean FilterWithSeverity , Dem_DTCSeverityType DTCSeverityMask , boolean FilterForFaultDetectionCounter);</pre>	
Service ID	DEM_SID_SetDTCFilter	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module
	DTCStatusMask	Status-byte mask for DTC status-byte filtering Values: 0x00: Autosar-specific value to deactivate the status-byte filtering (different meaning than in ISO 14229-1) to report all supported DTCs (used for service 0x19 subfunctions 0x0A/0x15) 0x01...0xFF: Status-byte mask according to ISO 14229-1 DTCStatusMask (handed over by Dcm from service request directly) to filter for DTCs with at least one status bit set matching this status-byte mask
	DTCFormat	Defines the output-format of the requested DTC values for the sub-sequent API calls
	DTCOrigin	If the Dem supports more than one event memory, this parameter is used to select the source memory the DTCs shall be read from.
	FilterWithSeverity	This flag defines whether severity information (ref. to parameter below) shall be used for filtering. This is to allow for coexistence of DTCs with and without severity information. ▶ TRUE Severity information is a filter criteria



		<ul style="list-style-type: none"> ▶ FALSE Severity information is not a filter criteria
	DTCSeverityMask	This parameter contains the DTCSeverity-Mask according to ISO14229-1.
	FilterForFaultDetectionCounter	<p>This flag defines whether Fault Detection Counter(FDC) information shall be used for filtering. This is to allow for coexistence of DTCs with and without FDC information. If FDC information is a filter criteria, only those DTCs with a FDC value between 1 and 0x7E shall be reported.</p> <ul style="list-style-type: none"> ▶ TRUE FDC information is a filter criteria ▶ FALSE FDC information is not a filter criteria
Return Value		Status of the operation to (re-)set a DTC filter
	E_OK	Indicates that the filter was successfully set
	E_NOT_OK	Indicates a wrong DTCOrigin or DTCFormat
Description		<p>This API shall be used for the API's Dem_GetNextFilteredDTC(), Dem_GetNextFilteredDTCAndFDC() and Dem_GetNextFilteredDTCAndSeverity(). The function resets the internal counter to the first event that matches the filter settings. The filter mask attributes are used until the next call of Dem_SetDTCFilter or Dem_Init().</p> <p>Note: This API must only be used by the Dcm module! Remark: If the event does not uses the debouncing inside Dem, then the Dem must request this information via Xxx_DemGetFaultDetectionCounter.</p>

5.3.3.4.100. Dem_SetDTCSuppression

Purpose	Set the suppression status of a specific DTC.
Synopsis	Std_ReturnType Dem_SetDTCSuppression (uint32 DTC , Dem_DTCFormatType DTCFormat , boolean SuppressionStatus);
Service ID	DEM_SID_SetDTCSuppression
Sync/Async	Synchronous
Reentrancy	Non Reentrant



Parameters (in)	DTC	Diagnostic Trouble Code
	DTCFormat	Defines the input-format of the provided DTC value.
	SuppressionStatus	This parameter specifies whether the respective DTC shall be disabled (TRUE) or enabled (FALSE).
Return Value	success of operation	
	E_OK	Operation was successful
	E_NOT_OK	Operation failed or event entry for this DTC still exists
Description	This API shall be used to suppress the external reporting (e.g. DTC number, UDS status, DTC extended data records, DTC statistical data,...) of a DTC.	

5.3.3.4.101. Dem_SetEnableCondition

Purpose	Function to set enable condition.	
Synopsis	Std_ReturnType Dem_SetEnableCondition (uint8 EnableConditionID , boolean ConditionFulfilled);	
Service ID	DEM_SID_SetEnableCondition	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EnableConditionID	This parameter identifies the enable condition.
	ConditionFulfilled	This parameter specifies whether the enable condition assigned to the EnableConditionID is fulfilled (TRUE) or not fulfilled (FALSE).
Return Value	success of operation	
	E_OK	enable condition could be set successfully
	E_NOT_OK	enable condition could not be set
Description	<p>This API shall be used to set the enable condition. For each event an enable condition value is assigned to. An enable condition specifies a certain number of checks (e.g. correct voltage range) for an event before the event can be qualified as confirmed.</p> <p>Configuration: Required configuration parameters per event:</p> <ul style="list-style-type: none"> ▶ EnableConditionID 	



▶ EnableConditionStatus

This API is optional and depends on the automotive manufacturer.

Note: This API is not available, if no enable condition are configured.

5.3.3.4.102. Dem_SetEventAvailable

Purpose	Set the available status of a specific event.	
Synopsis	<code>Std_ReturnType Dem_SetEventAvailable (Dem_EventIdType EventId, , boolean AvailableStatus);</code>	
Service ID	DEM_SID_SetEventAvailable	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
	AvailableStatus	Specifies whether the respective event shall be disabled (FALSE) or enabled (TRUE).
Return Value	success of operation	
	E_OK	Operation was successful
	E_NOT_OK	Operation failed or change of available status not accepted

5.3.3.4.103. Dem_SetEventDisabled

Purpose	Service for reporting the Event as disabled to the Dem for the PID \$41 computation.	
Synopsis	<code>Std_ReturnType Dem_SetEventDisabled (Dem_EventIdType EventId);</code>	
Service ID	DEM_SID_SetEventDisabled	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	EventId	Identification of an Event by assigned EventId.
Return Value	success of operation	



	E_OK	set of event to disabled was successful
	E_NOT_OK	set of event disabled failed

5.3.3.4.104. Dem_SetEventStatus

Purpose	Processes the events reported by SW-Cs via RTE.	
Synopsis	Std_ReturnType Dem_SetEventStatus (Dem_EventIdType EventId , Dem_EventStatusType EventStatus);	
Service ID	DEM_SID_SetEventStatus	
Sync/Async	Synchronous/Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned Event ID. The Event ID is configured in the Dem.
	EventStatus	Monitor test result
Return Value	success of operation	
	E_OK	set of event status was successful
	E_NOT_OK	set of event status failed or could not be accepted
Description	<p>This API can only be used through the RTE, and therefore no declaration is exported via Dem.h. Some bits of the UDS DTC status byte changes synchronously or asynchronously (refer to Dem036 and Dem379).</p> <p>Service for reporting the Event Status to the Dem. This service shall be used to report an Event Status as soon as a new test result is available.</p> <p>This Service stores the event in the Event Memory.</p> <p>API is called from the Monitor Function.</p> <p>Caveats: Dem configuration during integration of Monitor Functions is system specific.</p>	

5.3.3.4.105. Dem_SetFreezeFrameRecordFilter

Purpose	Sets a freeze frame record filter.
Synopsis	Std_ReturnType Dem_SetFreezeFrameRecordFilter (uint8 ClientId , Dem_DTCFormatType DTCFormat , uint16 * NumberOfFilteredRecords);



Service ID	DEM_SID_SetFreezeFrameRecordFilter	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different client IDs. Non reentrant for the same client ID.	
Parameters (in)	ClientId	Unique client ID, assigned to the instance of the calling module.
	DTCFormat	Defines the output-format of the requested DTC values.
Parameters (out)	NumberOfFilteredRecords	Number of freeze frame records currently stored in the event memory.
Return Value	Status of the operation of type Std_ReturnType	
	E_OK	Filter is accepted.
	E_NOT_OK	Wrong filter selected.
Description	<p>The filtered freeze frame records can be retrieved by the function Dem_GetNextFilteredRecord(). This filter always belongs to primary memory.</p> <p>Note: This API must only be used by the Dcm module!</p>	

5.3.3.4.106. Dem_SetOperationCycleCntValue

Purpose	Provides the value of the external operation cycle counter.	
Synopsis	Std_ReturnType Dem_SetOperationCycleCntValue (uint8 OperationCycleId , uint8 CounterValue);	
Service ID	DEM_SID_SetOperationCycleCntValue	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	OperationCycleId	Identification of operation cycle, like power cycle, driving cycle, ... (0 <= OpCycleId < DEM_NUM_OPCYCLES)
	CounterValue	Current external counter value of the respective operation cycle.
Return Value	success of operation	
	E_OK	set of operation cycle counter was successful
	E_NOT_OK	set of operation cycle counter failed
Description	This API can only be used through the RTE, and therefore no declaration is exported via Dem.h.	



Note: Functionality not implemented, only stub available.

5.3.3.4.107. Dem_SetOperationCycleState

Purpose	Sets an operation cycle state.	
Synopsis	<code>Std_ReturnType Dem_SetOperationCycleState (Dem_OperationCycleIdType OperationCycleId , Dem_OperationCycleStateType CycleState);</code>	
Service ID	DEM_SID_SetOperationCycleState	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	OperationCycleId	Identification of operation cycle, like power cycle, driving cycle, ... (0 <= OpCycleId < DEM_NUM_OPCYCLES)
	CycleState	New operation cycle state: (re-)start or end
Return Value	success of operation	
	E_OK	set of operation cycle was successful
	E_NOT_OK	set of operation cycle failed
Description	<p>Dem shall be called by the SW-Component as soon as it detects the status change of the CycleState for the Operation Cycle.</p> <p>Configuration: The OperationCycleId shall be configured in view of sender receiver communication.</p>	

5.3.3.4.108. Dem_SetPfcCycleQualified

Purpose	Marks the current OBD driving cycle as having met the criteria for the PFC cycle. API is needed in OBD-relevant ECUs only.	
Synopsis	<code>Std_ReturnType Dem_SetPfcCycleQualified (void);</code>	
Service ID	DEM_SID_SetPfcCycleQualified	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Return Value	success of operation	
	E_OK	always E_OK is returned
	E_NOT_OK	E_NOT_OK will never appear



5.3.3.4.109. Dem_SetPtoStatus

Purpose	Service to set the status of the PTO.	
Synopsis	Std_ReturnType Dem_SetPtoStatus (boolean PtoStatus);	
Service ID	DEM_SID_SetPtoStatus	
Sync/Async	Synchronous	
Reentrancy	Non reentrant	
Parameters (in)	PtoStatus	sets the status of the PTO (TRUE == active, FALSE == inactive)
Return Value	success of operation	
	E_OK	new PTO status has been adopted by Dem
	E_NOT_OK	new PTO status has not been adopted by Dem
Description	Note: Functionality not implemented, only stub available.	

5.3.3.4.110. Dem_SetSI30Notification

Purpose	Sets the SI30 Status by event.Set WarningIndicatorRequestedSinceLastClear (SI30.-bit5).	
Synopsis	Std_ReturnType Dem_SetSI30Notification (Dem_EventIdType EventId);	
Service ID	DEM_SID_SetSI30Notification	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of the Event for which the status shall be set.
Return Value	success of operation	
	E_OK	Operation was successful
	E_NOT_OK	Operation failed

5.3.3.4.111. Dem_SetSI30Symptom

Purpose	Set SymptomSinceLastClear (SI30.bit4).
Synopsis	Std_ReturnType Dem_SetSI30Symptom (Dem_EventIdType EventId);
Service ID	DEM_SID_SetSI30Symptom



Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of the Event for which the status shall be set.
Return Value	success of operation	
	E_OK	Operation was successful
	E_NOT_OK	Operation failed

5.3.3.4.112. Dem_SetStorageCondition

Purpose	Sets a storage condition.	
Synopsis	Std_ReturnType Dem_SetStorageCondition (uint8 StorageConditionID , boolean ConditionFulfilled);	
Service ID	DEM_SID_SetStorageCondition	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	StorageConditionID	This parameter identifies the storage condition.
	ConditionFulfilled	This parameter specifies whether the enable condition assigned to the EnableConditionID is fulfilled (TRUE) or not fulfilled (FALSE).
Return Value	success of operation	
	E_OK	storage condition could be set successfully
	E_NOT_OK	storage condition could not be set
Description	Note: Functionality not implemented, only stub available.	

5.3.3.4.113. Dem_SetWIRStatus

Purpose	Sets the WIR status bit via failsafe SW-Cs.	
Synopsis	Std_ReturnType Dem_SetWIRStatus (Dem_EventIdType EventId , boolean WIRStatus);	
Service ID	DEM_SID_SetWIRStatus	
Sync/Async	Synchronous	



Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId. The Event Number is configured in the DEM. Min.: 1 (0: Indication of no Event or Failure). Max.:Result of configuration of Event Numbers in DEM (Max is either 255 or 65535).
	WIRStatus	Requested status of event related WIR-bit (regarding to the current status of function inhibition). WIRStatus = TRUE -> WIR-bit shall be set to "1". WIRStatus = FALSE -> WIR-bit shall be set to "0", if no internal indicator requested it on.
Return Value	success of operation	
	E_OK	The request is accepted.
	E_NOT_OK	The request is not accepted (e.g. disabled controidTCSetting) and should be repeated.
Description	<p>Note: Because the user controlled warning indicator requested bits are only stored volatile, after a module restart this information is lost for Dem. The failsafe SW-C application must know the current UserCtrlWIR and request it again, if necessary.</p> <p>This API can only be used through the RTE, and therefore no declaration is exported via Dem.h.</p>	

5.3.3.4.114. Dem_Shutdown

Purpose	Shutdown function.
Synopsis	<code>void Dem_Shutdown (void);</code>
Service ID	DEM_SID_Shutdown
Sync/Async	Synchronous
Reentrancy	Non reentrant
Description	<p>This function is used to complete all pending operations in the Dem by preparing the internal states and event data for transfer to the NVRAM</p> <p>Caveats: Once this function has been executed no further updates are applied to the Dem internal event data.</p>



5.3.4. Integration notes

5.3.4.1. Exclusive areas

This section describes the exclusive areas used by the `Dem` module.

5.3.4.1.1. SCHM_DEM_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.

5.3.4.1.2. Exclusive areas for BSW distribution functionality

In case the feature `DemDistributedBSWEventReporting` is enabled, each `Dem` satellite instance will have an exclusive area of its own to protect their local resources necessary for buffer handling in the pre-initialized phase.

The exclusive area of the satellite instances must be of type *Interrupt Locking*.

The `Dem` master instance exclusive area will be named `SCHM_DEM_EXCLUSIVE_AREA_MASTER` and the `Dem` satellite's exclusive area will be named like `SCHM_EXCLUSIVE_AREA_SatelliteCore_<SatelliteCoreId>`

5.3.4.2. Production errors

Production errors are not reported by the `Dem` module.

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
VAR_CLEARED_8
VAR_CLEARED_16
VAR_CLEARED_UNSPECIFIED

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 guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

Integration requirements are not listed for the Dem module.

5.4. FiM

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.
FiMConfigSet	1..1	This container contains the configuration parameters and subcontainers of the FiM module supporting multiple configuration sets. This container is a Multiple Configuration Container, i.e. this container and its subcontainers exist once per configuration set.
FiMGeneral	1..1	Label: General Configuration This container contains the configuration parameters of the FiM.

**Containers included**

PublishedInformation	1..1	Label: EB Published Information Additional published parameters not covered by Common-PublishedInformation container.
--------------------------------------	------	---

Parameters included

Parameter name	Multiplicity
IMPLEMENTATION_CONFIG_VARIANT	1..1

Parameter Name	IMPLEMENTATION_CONFIG_VARIANT	
Label	Configuration Variant	
Multiplicity	1..1	
Type	ENUMERATION	
Default value	VariantPreCompile	
Range	VariantPreCompile	
Configuration class	VariantPreCompile:	VariantPreCompile

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	2
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	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	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	4
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	11
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

5.4.1.2. FiMConfigSet

Containers included		
Container name	Multiplicity	Description
FiMEventSummary	0..n	The event summary link associates an event summary with a specific Dem event ID. This link means that a particular FID, which has to be disabled in case of a summarized event, is to be disabled in any of the specific events.
FiMFID	0..n	This container corresponds to an ECU function that is controlled by the FIM module.
FiMMessage	0..254	This container defines a FIM message.
FiMInhibitionConfiguration	1..n	This container includes all configuration parameters that specify the relationship between event and FID.
FiMSummaryEventId	0..n	This container defines the name of a summarized event.

5.4.1.3. FiMEventSummary

Parameters included	
Parameter name	Multiplicity
FiMInputSumEventRef	1..1



Parameters included

FiMOutputSumEventRef	1..1
--------------------------------------	------

Parameter Name	FiMInputSumEventRef
Label	Dem Event
Description	Defines a Dem event ID that is assigned to the event summary.
Multiplicity	1..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	FiMOutputSumEventRef
Label	Event Summary
Description	Defines the event summary that groups the Dem events.
Multiplicity	1..1
Type	SYMBOLIC-NAME-REFERENCE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.4.1.4. FiMFID

Parameters included

Parameter name	Multiplicity
FiMFunctionId	1..1
FiMSetSymptom	1..1
FiMMessageRef	1..1
FiMConditionEventRef	1..1
FiMForcedReleaseEventRef	1..1
FiMRecoveryEventRef	1..1

Parameter Name	FiMFunctionId
Label	Function Id
Description	Defines a one-based and unique numeric identifier that represents a functionality.



	Note: The FiM generates preprocessor macros with names that contain the symbolic names of configured Function IDs, i.e., <code>FimConf_FiMFID_<configured FiMFID name></code> . These macros are defined with the corresponding configured values of the <code>FimFunctionId</code> .
Multiplicity	1..1
Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	FiMSetSymptom	
Label	Set Symptom	
Description	<p>Enables the reporting of a symptom to the Dem for this FID.</p> <p>Note: If the parameter is not available, it is interpreted as 'false'.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Extended Support (<code>FiMExtendedSupport</code>): must be enabled. ▶ Extended Level (<code>FiMExtendedLevel</code>): must be set to <code>FIM_EXTENSION_LVL1</code> or <code>FIM_EXTENSION_LVL2</code>. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	VCC	

Parameter Name	FiMMessagRef	
Label	Message Reference	
Description	<p>Reference to the message that is connected to this FID.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Extended Support (<code>FiMExtendedSupport</code>): must be enabled. ▶ Extended Level (<code>FiMExtendedLevel</code>): must be set to <code>FIM_EXTENSION_LVL1</code> or <code>FIM_EXTENSION_LVL2</code>. 	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile



Origin	VCC
---------------	-----

Parameter Name	FiMConditionEventRef	
Label	Condition Event Reference	
Description	<p>Reference to the Dem condition event. If this event occurs, the corresponding message is sent.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Extended Support (<code>FiMExtendedSupport</code>): must be enabled. ▶ Extended Level (<code>FiMExtendedLevel</code>): must be set to <code>FIM_EXTENSION_LVL1</code> or <code>FIM_EXTENSION_LVL2</code>. 	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	VCC	

Parameter Name	FiMForcedReleaseEventRef	
Label	Forced Release Event Reference	
Description	<p>Reference to the event that forces the release of the FID.</p> <p>Note: For some FIDs, a so-called forced release is required to avoid deadlock situations. If the FID evaluation results in inhibition, the status of the force-release event is checked with <code>Dem_GetEventStatus()</code>. As long as the <code>testFailed</code> bit is set for the force-release event, the FID is not inhibited.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Extended Support (<code>FiMExtendedSupport</code>): must be enabled. ▶ Extended Level (<code>FiMExtendedLevel</code>): must be set to <code>FIM_EXTENSION_LVL1</code> or <code>FIM_EXTENSION_LVL2</code>. 	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	VCC	

Parameter Name	FiMRecoveryEventRef	
Label	Recovery Event Reference	



Description	<p>Reference to the Dem recovery event for the FID.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Extended Support (<code>FiMExtendedSupport</code>): must be enabled. ▶ Extended Level (<code>FiMExtendedLevel</code>): must be set to <code>FIM_EXTENSION_LVL2</code>. <p>Note: As long as the recovery event is active (calibrated and with status Failed), the error message status of the FID is kept enabled. So, regardless of whether the FID is reset to permit, the FID is considered as active until the recovery event is reset (status changed to Passed).</p>
Multiplicity	1..1
Type	REFERENCE
Configuration class	PreCompile: VariantPreCompile
Origin	VCC

5.4.1.5. FiMMessage

Parameters included	
Parameter name	Multiplicity
FiMMessageld	1..1

Parameter Name	FiMMessageld
Label	Message Id
Description	<p>Defines a unique identifier of a message. Click on <i>Calculate value</i> to set the ID automatically. You can also use the Handle ID wizard to set the ID automatically.</p> <p>Note: Do not set this value manually. The ID should be generated to prevent gaps and multiple use of an ID. The messages must be sequentially numbered beginning with 1 and no gaps in between. 0 is an invalid value to indicate <i>no message</i>.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Extended Support (<code>FiMExtendedSupport</code>): must be enabled. ▶ Extended Level (<code>FiMExtendedLevel</code>): must be set to <code>FIM_EXTENSION_LVL1</code> or <code>FIM_EXTENSION_LVL2</code>.
Multiplicity	1..1



Type	INTEGER	
Configuration class	PreCompile:	VariantPreCompile
Origin	VCC	

5.4.1.6. FiMInhibitionConfiguration

Containers included		
Container name	Multiplicity	Description
FiMInhEventId	1..1	Label: Inhibition Event Id The configuration parameter is used for both an existing Dem event and summarized events.

Parameters included	
Parameter name	Multiplicity
FiMInhInhibitionMask	1..1
FiMHighPrioInhibition	1..1
FiMInhFunctionIdRef	1..1

Parameter Name	FiMInhInhibitionMask
Label	Inhibition Mask
Description	<p>Defines the inhibition mask for an event to FID relation.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ FIM_LAST_FAILED: Last failed - DEM_UDS_STATUS_TF flag of Dem event status-byte is set (Use case: Re-configuration, avoiding follow-up errors). ▶ FIM_NOT_TESTED: Not tested this cycle - DEM_UDS_STATUS_TNCTOC flag of Dem event status-byte is set (Use case: Scheduling of monitors). ▶ FIM_TESTED: Tested - DEM_UDS_STATUS_TNCTOC flag of Dem event status-byte is not set (Use case: Self deactivation, check during driving cycle). ▶ FIM_TESTED_AND_FAILED: Tested and failed - DEM_UDS_STATUS_TF flag of Dem event status-byte is set and DEM_UDS_STATUS_TNCTOC flag is not set (Use case: Avoiding deadlocks, repeated monitoring). ▶ FIM_LAST_UNCONFIRMED: SI30 status byte bit 0 (UnconfirmedDTC) is set. Only available if extended support is enabled.



	<ul style="list-style-type: none"> ▶ FIM_FAILED_THIS_OP_CYCLE: UDS status byte bit 1 (TestFailedThisOperationCycle) is set. Only available if extended support is enabled. ▶ FIM_UNCONFIRMED_THIS_OP_CYCLE: SI30 status byte bit 1 (UnconfirmedDTCThisOperationCycle) is set. Only available if extended support is enabled.
Multiplicity	1..1
Type	ENUMERATION
Default value	FIM_LAST_FAILED
Range	FIM_LAST_FAILED FIM_NOT_TESTED FIM_TESTED FIM_TESTED_AND_FAILED FIM_LAST_UNCONFIRMED FIM_FAILED_THIS_OP_CYCLE FIM_UNCONFIRMED_THIS_OP_CYCLE
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	FiMHighPriInhibition	
Label	High Priority Inhibition	
Description	<p>Defines if an inhibition has high priority. If not present, the parameter is interpreted as false.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Extended Support (FiMExtendedSupport): must be enabled. ▶ Extended Level (FiMExtendedLevel): must be set to FIM_EXTENSION_LVL1 or FIM_EXTENSION_LVL2. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	PreCompile:	VariantPreCompile
Origin	VCC	

Parameter Name	FiMInhFunctionIdRef	
Label	Inhibition Function Id	



Description	The configuration parameter is used to specify the FID for an event - FID relation.	
Multiplicity	1..1	
Type	REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

5.4.1.7. FiMInhEventId

Containers included		
Container name	Multiplicity	Description
FiMInhRefChoice	1..1	<p>Label: Inhibition Choice</p> <p>Defines whether the inhibition relates to an existing Dem event or an event summary.</p>

5.4.1.8. FiMInhRefChoice

Containers included		
Container name	Multiplicity	Description
FiMInhChoiceDemRef	1..1	<p>Label: Inhibition Choice Dem</p> <p>Defines that the inhibition relates to an existing Dem event.</p>
FiMInhChoiceSumRef	1..1	<p>Label: Inhibition Choice Summary</p> <p>Defines that the inhibition relates an event summary.</p>

5.4.1.9. FiMInhChoiceDemRef

Parameters included	
Parameter name	Multiplicity
FiMInhEventRef	1..1
Parameter Name	FiMInhEventRef
Label	Inhibition Dem Event
Description	Reference to an existing Dem event.
Multiplicity	1..1



Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.4.1.10. FiMInhChoiceSumRef

Parameters included	
Parameter name	Multiplicity
FiMInhSumRef	1..1

Parameter Name	FiMInhSumRef	
Label	Inhibition Event Summary	
Description	Reference to an event summary.	
Multiplicity	1..1	
Type	SYMBOLIC-NAME-REFERENCE	
Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

5.4.1.11. FiMSummaryEventId

Parameters included	
Parameter name	Multiplicity
FiMEventSumId	1..1

Parameter Name	FiMEventSumId
Label	Event Summary Id
Description	<p><i>This parameter is not required for FiM usage. However, symbolic names are generated.</i></p> <p>The summarized event ID definition record defines the existence of a summarized event with a specific name. This summarized event can be referenced in the event summary (as FiMSummaryEventId) and inhibition configuration (as FiMInhEventId).</p>
Multiplicity	1..1



Type	INTEGER
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

5.4.1.12. FiMGeneral

Parameters included	
Parameter name	Multiplicity
FiMExtendedSupport	1..1
FiMExtendedLevel	1..1
FiMMaxHighPrioEventToFidLinks	1..1
FiMMaxHandledFidEventLinksPerScheduling	1..1
FiMDataFixed	1..1
FiMDevErrorDetect	1..1
FiMEventUpdateTriggeredByDem	1..1
FiMRteUsage	1..1
FiMMaxEventFidLinks	1..1
FiMMaxEventsPerFid	1..1
FiMMaxFidsPerEvent	1..1
FiMMaxSummaryEvents	1..1
FiMMaxSummaryLinks	1..1
FiMMaxTotalLinks	1..1
FiMTaskTime	1..1
FiMVersionInfoApi	1..1
FiMMeasurementSupport	1..1

Parameter Name	FiMExtendedSupport	
Label	Extended Support	
Description	Enables VCC-specific extensions.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile: VariantPreCompile	



Origin	VCC
---------------	-----

Parameter Name	FiMExtendedLevel							
Label	Extended Level							
Description	<p>Serves to select the VCC-specific extension set.</p> <p>Range:</p> <ul style="list-style-type: none"> ▶ FIM_NO_EXTENSION: No extensions. ▶ FIM_EXTENSION_LVL1: <ul style="list-style-type: none"> ▶ Extended FiMInhlInhibitionMask ▶ Symptom ▶ Notification with Condition Event including callout to Dem ▶ FIM_EXTENSION_LVL2: <ul style="list-style-type: none"> ▶ Extended FiMInhlInhibitionMask ▶ Symptom ▶ Notification with Condition Event including callout to Dem ▶ Force-release event ▶ Recovery event <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Extended Support (<code>FiMExtendedSupport</code>): must be enabled. 							
Multiplicity	1..1							
Type	ENUMERATION							
Default value	FIM_NO_EXTENSION							
Range	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: left; padding-right: 10px;">FIM_NO_EXTENSION</td> <td style="border-top: 1px solid black; border-bottom: 1px solid black;"></td> </tr> <tr> <td style="text-align: left; padding-right: 10px;">FIM_EXTENSION_LVL1</td> <td style="border-bottom: 1px solid black;"></td> </tr> <tr> <td style="text-align: left; padding-right: 10px;">FIM_EXTENSION_LVL2</td> <td></td> </tr> </table>		FIM_NO_EXTENSION		FIM_EXTENSION_LVL1		FIM_EXTENSION_LVL2	
FIM_NO_EXTENSION								
FIM_EXTENSION_LVL1								
FIM_EXTENSION_LVL2								
Configuration class	PreCompile:	VariantPreCompile						
Origin	VCC							

Parameter Name	FiMMaxHighPrioEventToFidLinks	
Label	Max High Prio Event To Fid Links	
Description	Vendor-specific configuration parameter that defines the maximum number of inhibitions that can be configured or calibrated as high priority inhibitions.	



	<p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Extended Support (<code>FiMExtendedSupport</code>): must be enabled. ▶ Extended Level (<code>FiMExtendedLevel</code>): must be set to <code>FIM_EXTENSION_LVL1</code> or <code>FIM_EXTENSION_LVL2</code>.
Multiplicity	1..1
Type	INTEGER
Default value	1
Configuration class	PreCompile: VariantPreCompile
Origin	VCC

Parameter Name	FiMMaxHandledFidEventLinksPerScheduling
Label	Max Handled Fid Event Links Per Scheduling
Description	<p>Defines the number of FID - event links that are handled per MainFunction call.</p> <p>Dependency on parameter(s):</p> <ul style="list-style-type: none"> ▶ Extended Support (<code>FiMExtendedSupport</code>): must be enabled. ▶ Extended Level (<code>FiMExtendedLevel</code>): must be set to <code>FIM_EXTENSION_LVL1</code> or <code>FIM_EXTENSION_LVL2</code>.
Multiplicity	1..1
Type	INTEGER
Default value	65535
Configuration class	PreCompile: VariantPreCompile
Origin	VCC

Parameter Name	FiMDataFixed
Label	Data Fixed
Description	<p>Enables the calibration of inhibit relations. The scope of the parameter is not only to calibrate inhibit data but also to provide the option to protect inhibit data for consistency reasons.</p> <p>This parameter is only available if extended FiM support is disabled. Extended FiM support has calibration always enabled.</p>
Multiplicity	1..1
Type	BOOLEAN
Default value	true



Configuration class	PreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	FIMDevErrorDetect	
Label	Enable Development Error Detection	
Description	Enables the detection of development errors during development.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	true	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	FIMEEventUpdateTriggeredByDem	
Label	Event Update Triggered By Dem	
Description	Specifies how FiM obtains the status of event IDs. If enabled, DemTriggerFiM-Reports in Dem must also be enabled, so that Dem notifies FiM about its event status changes through the API <code>Fim_DemTriggerOnEventStatus()</code> . <ul style="list-style-type: none"> ▶ true: Dem informs FiM about changes of the event status. ▶ false: FiM polls the event status from the Dem module either cyclically or on demand. 	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECU	

Parameter Name	FIMRteUsage	
Label	Enable Rte Usage	
Description	Enables the use of the RTE for this module. For an easy integration, it is recommended to disable the use of the RTE at the beginning of the integration work.	
Multiplicity	1..1	
Type	BOOLEAN	



Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

Parameter Name	FiMMaxEventFidLinks	
Label	Maximum Event-Fid Links	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This configuration parameter specifies the total maximum number of links between event IDs and FIDs.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	65535	
Range	<p><=65535</p> <p>>=1</p>	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	FiMMaxEventsPerFid	
Label	Maximum Events Per Fid	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This configuration parameter specifies the maximum number of event IDs that can be linked to a single FID.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	255	
Range	<p><=65535</p> <p>>=1</p>	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	FiMMaxFidsPerEvent	
-----------------------	---------------------------	--



Label	Maximum Fids Per Event	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This configuration parameter specifies the maximum number of FIDs that can be linked to a single event.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	65535	
Range	<p><=65535</p> <p>>=1</p>	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	FIMMaxSummaryEvents	
Label	Maximum Summary Events	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This configuration parameter specifies the maximum number of summarized events that can be configured.</p>	
Multiplicity	1..1	
Type	INTEGER	
Default value	65535	
Range	<p><=65535</p> <p>>=0</p>	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	FIMMaxSummaryLinks	
Label	Maximum Summary Links	
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This configuration parameter specifies the total maximum number of links between event IDs and summarized events.</p>	



Multiplicity	1..1
Type	INTEGER
Default value	65535
Range	<=65535 >=0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	FiMMaxTotalLinks
Label	Maximum Total Links
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>This configuration parameter specifies the total maximum number of links between event IDs and FIDs plus the number of links between event IDs and summarized events.</p>
Multiplicity	1..1
Type	INTEGER
Default value	65535
Range	<=65535 >=1
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC

Parameter Name	FiMTaskTime
Description	<p><i>The functionality related to this parameter is not supported by the current implementation.</i></p> <p>Defines the time for the periodic cyclic task.</p> <p>Note: This configuration value must be equal to the value in the Basic Software Scheduler configuration of the RTE module.</p> <p>The AUTOSAR configuration standard is to use SI units. So this parameter is defined as float value in seconds.</p> <p><i>min:</i> A negative value is not allowed.</p> <p><i>max:</i> FID must be set after a maximal time of 100ms after the Dem status is set.</p>



	<i>upperMultiplicity</i> : Exactly one TaskTime must be specified per configuration. <i>lowerMultiplicity</i> : Exactly one TaskTime must be specified per configuration.
Multiplicity	1..1
Type	FLOAT
Default value	0.02
Range	<=0.1 >=0.0
Configuration class	VariantPreCompile: VariantPreCompile
Origin	AUTOSAR_ECUC V1.0.0

Parameter Name	FiMVersionInfoApi	
Label	Enable Version Info API	
Description	Enables the API to get the version information.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	AUTOSAR_ECUC	

Parameter Name	FiMMeasurementSupport	
Label	Enable Measurement Support	
Description	Enables the measurement support information.	
Multiplicity	1..1	
Type	BOOLEAN	
Default value	false	
Configuration class	VariantPreCompile:	VariantPreCompile
Origin	Elektrobit Automotive GmbH	

5.4.1.13. PublishedInformation

Parameters included	
Parameter name	Multiplicity



Parameters included

PbcfgMSupport	1..1
-------------------------------	------

Parameter Name	PbcfgMSupport
Label	PbcfgM support
Description	Specifies whether or not the FiM can use the PbcfgM module for post-build support.
Multiplicity	1..1
Type	BOOLEAN
Default value	false
Configuration class	PublishedInformation:
Origin	Elektrobit Automotive GmbH

5.4.2. Application programming interface (API)

5.4.2.1. Macro constants

5.4.2.1.1. FIM_AR_RELEASE_MAJOR_VERSION

Purpose	AUTOSAR release major version.
Value	4U

5.4.2.1.2. FIM_AR_RELEASE_MINOR_VERSION

Purpose	AUTOSAR release minor version.
Value	0U

5.4.2.1.3. FIM_AR_RELEASE_REVISION_VERSION

Purpose	AUTOSAR release revision version.
----------------	-----------------------------------



Value	3U
--------------	----

5.4.2.1.4. FIM_EXTENSION_LVL1

Purpose	Level of extended support.
Value	1U
Description	Extended support level 1 enables message id, symptom bit, high priority inhibitions and condition event.

5.4.2.1.5. FIM_EXTENSION_LVL2

Purpose	Level of extended support.
Value	2U
Description	In addition to extended support level 1 this option enables support for recovery and force release event.

5.4.2.1.6. FIM_E_EVENTID_OUT_OF_RANGE

Purpose	Error Code.
Value	0x04U
Description	Dem calls FiM with wrong EventId.

5.4.2.1.7. FIM_E_FID_OUT_OF_RANGE

Purpose	Error Code.
Value	0x03U
Description	FiM_GetFunctionPermission() called with wrong FID.

5.4.2.1.8. FIM_E_INVALID_POINTER

Purpose	Error Code.
Value	0x05U



Description	API is invoked with NULL Pointer.
-------------	-----------------------------------

5.4.2.1.9. FIM_E_MESSAGEID_OUT_OF_RANGE

Purpose	Error Code.
Value	0x0AU
Description	FiM is not initialized.

5.4.2.1.10. FIM_E_UNINIT

Purpose	Error Code.
Value	0x06U
Description	FiM is not initialized.

5.4.2.1.11. FIM_E_WRONG_PERMISSION_REQ

Purpose	Error Code.
Value	0x01U
Description	FIM_GetFunctionPermission() is called by the application (SW-C) before complete initialization.

5.4.2.1.12. FIM_E_WRONG_TRIGGER_ON_EVENT

Purpose	Error Code.
Value	0x02U
Description	Dem calls FiM before the FiM is initialized.

5.4.2.1.13. FIM_INSTANCE_ID

Purpose	Id of instance of FiM.
Value	0U



5.4.2.1.14. FIM_MODULE_ID

Purpose	AUTOSAR module identification.
Value	11U

5.4.2.1.15. FIM_NO_EXTENSION

Purpose	Level of extended support.
Value	0U
Description	Extended support is disabled.

5.4.2.1.16. FIM_SID_DEM_INIT

Purpose	AUTOSAR API service ID.
Value	0x03U
Description	Definition of service ID for FiM_DemInit() .

5.4.2.1.17. FIM_SID_DEM_TRIGGER_ON_EVENT

Purpose	AUTOSAR API service ID.
Value	0x02U
Description	Definition of service ID for FiM_DemTriggerOnEventStatus() .

5.4.2.1.18. FIM_SID_DEM_TRIGGER_ON_UNCONFIRMED

Purpose	Extended FIM API service ID.
Value	0xFFU
Description	Definition of service ID for FiM_DemTriggerOnUnconfirmed() .

5.4.2.1.19. FIM_SID_GET_FUNCTION_PERMISSION

Purpose	AUTOSAR API service ID.
----------------	-------------------------



Value	0x01U
Description	Definition of service ID for FiM_GetFunctionPermission() .

5.4.2.1.20. FIM_SID_GET_MESSAGE_STATUS

Purpose	Extended FIM API service ID.
Value	0xFEU
Description	Definition of service ID for FiM_GetMessageStatus() .

5.4.2.1.21. FIM_SID_GET_VERSION_INFO

Purpose	AUTOSAR API service ID.
Value	0x04U
Description	Definition of service ID for FiM_GetVersionInfo() .

5.4.2.1.22. FIM_SID_INIT

Purpose	AUTOSAR API service ID.
Value	0x00U
Description	Definition of service ID for FiM_Init() .

5.4.2.1.23. FIM_SID_MAIN_FUNCTION

Purpose	AUTOSAR API service ID.
Value	0x05U
Description	Definition of service ID for FiM_MainFunction() .

5.4.2.1.24. FIM_SW_MAJOR_VERSION

Purpose	AUTOSAR module major version.
Value	2U



5.4.2.1.25. FIM_SW_MINOR_VERSION

Purpose	AUTOSAR module minor version.
Value	5U

5.4.2.1.26. FIM_SW_PATCH_VERSION

Purpose	AUTOSAR module patch version.
Value	4U

5.4.2.1.27. FIM_VENDOR_ID

Purpose	AUTOSAR vendor identification: Elektrobit Automotive GmbH.
Value	1U

5.4.2.2. Objects

5.4.2.2.1. FiM_FidInhibitionFlag

Purpose	SI30 Inhibition flag of FiM FIDs.
Type	uint8
Description	This variable holds the FID status of the SI30 events, one bit per FID.

5.4.2.3. Functions

5.4.2.3.1. FiM_DemInit

Purpose	This service re-initializes the FiM.
Synopsis	<code>void FiM_DemInit (void);</code>
Service ID	FIM_SIDDEMINIT
Sync/Async	Synchronous
Reentrancy	Non-Reentrant



Description	The Dem shall call this function to re-initialize the FiM module in case the Dem detects a status change of a certain number of events.
	Precondition: none

5.4.2.3.2. **FIM_DemTriggerOnEventStatus**

Purpose	This service to be provided to the Dem in order to call FiM upon status changes.	
Synopsis	<pre>void FIM_DemTriggerOnEventStatus (Dem_EventIdType EventId , uint8 EventStatusOld , uint8 EventStatusNew);</pre>	
Service ID	FIM_SID DEM_TRIGGER_ON_EVENT	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an Event by assigned event number. The Event Number is configured in the Dem.
	EventStatusOld	Extended event status before change
	EventStatusNew	Detected / reported of event status
Description	<p>The module Dem shall call this function whenever the status of an events changes.</p> <p>Precondition: FIM_EVENT_UPDATE_TRIGGERED_BY_DEM = STD_ON</p>	

5.4.2.3.3. **FIM_GetFunctionPermission**

Purpose	Reports the permission state to the functionality.	
Synopsis	<pre>Std_ReturnType FIM_GetFunctionPermission (Fim_FunctionIdType FID , boolean * Permission);</pre>	
Service ID	FIM_SID_GET_FUNCTION_PERMISSION	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	FID	Identification of a functionality by assigned FID. The FunctionId is configured in the FiM.
Parameters (out)	Permission	Pointer to permission to be filled with the permission information



Return Value	success of operation	
	E_OK	request is accepted
	E_NOT_OK	request is not accepted
Description	<p>The SW Components and the BSW shall use this function to query for the permission to execute a certain functionality represented by the respective FID.</p> <p>Precondition: none</p>	

5.4.2.3.4. **FiM_GetVersionInfo**

Purpose	Return the modules version information.	
Synopsis	<pre>void FiM_GetVersionInfo (Std_VersionInfoType * VersionInfoPtr) ;</pre>	
Service ID	<u>FIM_SID_GET_VERSION_INFO</u>	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (out)	VersionInfoPtr	Pointer to struct to be filled with the version information
Description	<p>This function provides the information to module vendor ID, module ID and software version major.minor.patch</p> <p>Precondition: FIM_VERSION_INFO_API = STD_ON</p>	

5.4.2.3.5. **FiM_Init**

Purpose	Initializes the FiM.	
Synopsis	<pre>void FiM_Init (const FiM_ConfigType * FiMConfigPtr) ;</pre>	
Service ID	<u>FIM_SID_INIT</u>	
Sync/Async	Synchronous	
Reentrancy	Non-Reentrant	
Parameters (in)	FiMConfigPtr	refer to FiM_ConfigType
Description	<p>This function initializes the FiM module.</p> <p>Precondition: none</p>	



5.4.2.3.6. **FIM_MainFunction**

Purpose	This function is used to evaluate permission states cyclically.
Synopsis	<code>void FIM_MainFunction (void);</code>
Service ID	<u>FIM_SID_MAIN_FUNCTION</u>
Description	Cyclic evaluation of the permission state of the fid is done using this function. Precondition: FIM_EVENT_UPDATE_TRIGGERED_BY_DEM = STD_OFF

5.4.3. Integration notes

5.4.3.1. Exclusive areas

This section describes the exclusive areas used by the **FIM** module.

5.4.3.1.1. **SCHM_FIM_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.

5.4.3.2. Production errors

Production errors are not reported by the **FIM** 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
CONST_UNSPECIFIED
CALIB_16
CALIB_8
CODE
VAR_INIT_8
VAR_CLEARED_UNSPECIFIED
VAR_CLEARED_8

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 guide, release notes, and EB tresos AutoCore known issues to successfully integrate your product.

Integration requirements are not listed for the FiM module.



6. ACG8 Diagnostic Stack tutorial

6.1. Overview

6.1.1. Goal of this tutorial

This chapter is helpful to make yourself familiar with some basic functions of the EB tresos AutoCore Generic 8 Diagnostic Stack and its modules. Furthermore, it helps you to make yourself familiar with how to configure basic software modules with EB tresos Studio.

6.1.2. Scope of this tutorial

This tutorial describes how to configure some of the basic functions of the Diagnostic Stack. This basic configuration includes the `Dem`, `Dcm`, and `Fim` modules.

To complete this tutorial, the following modules need to be configured in addition to the Diagnostic Stack modules:

- ▶ `NvM` (ACG8 Memory Stack)
- ▶ `PduR` (ACG8 COM Services)
- ▶ `EcuC` (ACG8 Base)
- ▶ `ComM` (ACG8 Mode Management)

6.1.3. Prerequisites

Required background knowledge

The following knowledge is necessary to successfully complete this tutorial:

- ▶ Basic knowledge of the AUTOSAR standard
- ▶ Knowledge of diagnostic concepts for cars, especially the UDS protocol

Required products

To complete this tutorial, you need an EB tresos Studio installation with at least the following modules:

- ▶ `EcuC`, contained in the EB tresos AutoCore Generic 8 Base product from Elektrobit (EB)



- ▶ Diagnostic Stack
- ▶ PduR, contained in the EB tresos AutoCore Generic 8 COM Services product from Elektrobit (EB)
- ▶ ComM, contained in the EB tresos AutoCore Generic 8 Mode Management product from Elektrobit (EB)
- ▶ NvM, contained in the EB tresos AutoCore Generic 8 Memory Stack product from Elektrobit (EB)

Required time

This tutorial takes approximately *1 hour*.

6.2. Background information

6.2.1. Purpose of the Diagnostic Stack

The basic software modules and the software components monitor their proper functionality. If an error occurs, the basic software modules or software components report the detected errors to the Diagnostic Stack. The purpose of the Diagnostic Stack is to manage and store the error information. This information allows a car mechanic to acquire the error information by using a diagnostic tester tool, also called tester. The error information helps the car mechanic to identify the root cause of the error and to fix it. The Diagnostic Stack consists of the following modules:

- ▶ Dem
The Diagnostic Event Manager (`Dem`) processes and stores diagnostic events and associated data.
- ▶ Dcm
The Diagnostic Communication Manager (`Dcm`) ensures diagnostic data flow and manages the diagnostic states, especially diagnostic sessions. Furthermore, the `Dcm` module checks if a diagnostic service is supported and can be executed in the current diagnostic session.
- ▶ Fim
The Function Inhibition Manager (`Fim`) provides a control mechanism for software components and the functionality therein. If necessary, the `Fim` can deactivate a defective software function and functions that depend on this defective function.

For more information on the Diagnostic Stack, see [Section 4.2, “Background information”](#).

Interaction with modules of other stacks

The Diagnostic Stack modules, among others, interact with the following modules that are not part of the Diagnostic Stack:



▶ NvM

The `Dem` module uses the non-volatile RAM manager to persistently store the diagnostic events.

▶ PduR

The PDU Router defines the paths of the diagnostic data flow.

▶ EcuC

The ECU Configuration module contains the PDUs that are required for the diagnostic data flow.

▶ ComM

The Communication Manager specifies if and how the `Dcm` can receive requests and transmit responses.

This list is not exhaustive. For a complete list of all non-diagnostic modules that interact with the Diagnostic Stack, see the AUTOSAR specifications of the `Dem` or `Dcm` modules.

6.2.2. Concept of this configuration

In this tutorial you perform a basic configuration of the Diagnostic Stack. The basic configuration that is described in this tutorial allows the basic software to do the following:

1. The `Dem` module receives monitor reports from other basic software modules and processes them. The monitor reports contain the diagnostic events.
2. If a diagnostic event occurs, the `Fim` checks if other software functions depend on the defective function. If necessary, the dependent software functions are disabled.
3. The `Dem` module stores the detected diagnostic events on a non-volatile memory via the `NvM` module. Therefore, the diagnostic event is not deleted when the engine is turned off.
4. The `Dcm` allows an off-board tester to gather diagnostic information about the status of the ECU. To do so, the car mechanic can use UDS services, e.g. to gather diagnostic trouble codes (DTCs). The `Dcm` processes the service request from the tester and gathers the requested information, i.e. the DTCs.

6.3. Creating a new project

This chapter describes how to create a new project in EB tresos Studio.



Creating a new project

Step 1

In the menu bar, select **File → New → Configuration Project**.



The **New Project Wizard** dialog opens up.

Step 2

In the **Project name** text box, enter `Test_DiagStack`.

Step 3

Click **Next**.

Step 4

In the **ECU ID** text box, enter `DiagStack_ECU`.

Step 5

From the **Target** drop-down list box, select your platform.

Step 6

Click **Next**.

Step 7

In the **Available Modules** tree area, double-click the following modules:

- ▶ `Dem`
- ▶ `Dcm`
- ▶ `Fim`
- ▶ `NvM`

Step 8

In the **Module Configurations** table area, click the `Dcm` entry.

Step 9

In the **Recommended Configuration** drop-down list box in the **Details** view, select `DcmRecConfigurationBasic`.

This recommended configuration contains a basic configuration of the `Dcm`.

Step 10

Click **Finish**.

The project is added to the **Project Explorer**.

6.4. Configuring the Diagnostic Event Manager (Dem)

The `Dem` module handles the event management. In this chapter, you configure the following functions of the `Dem`:

- ▶ 1 data element class
- ▶ 1 extended data record
- ▶ 1 freeze frame



- ▶ 1 event list
- ▶ DTCs
- ▶ Non-volatile memory blocks

For more information on the `Dem` module, see [Section 4.5.2, “Background information”](#).

For more information on the `NvM` module, see the EB tresos AutoCore Generic 8 Memory Stack documentation.



Configuring a data element class

First, you create a data element class. This data element class is referenced in both the extended data record and the freeze frame and defines how the `Dem` obtains the value of the data element.

Step 1

In the **Project Explorer**, double-click the **Dem** configuration node.

In the editor view, the **Dem** editor opens up.

Step 2

Click the **Data Elements** tab.

Step 3

In the **Data Element Class List** table, add an entry.

A **DemDataElementClass_0** entry is added to the table.

NOTE

Entry names in this tutorial are examples



When you add an entry to a table, the entry gets a default name. In this tutorial, these default table entry names are used. In a real project, you can rename the table entries.



Configuring an extended data record

In this step, you create an extended data record. The extended data record contains error information.

Step 1

Click the **Extended Data** tab.

Step 2

In the **Extended Data Class List** table, add an entry.

A **DemExtendedDataClass_0** entry is added to the table.

Step 3

In the **Extended Data Record List** table, add an entry.



An **EXTDATAREC_0** entry is added to the table. This entry is an extended data record.

Step 4

Open the **EXTDATAREC_0** entry.

Step 5

In the **Data Element Class List** table, add an entry.

Step 6

In the **Data Element Class** drop-down list box, select **DemDataElementClass_0**.

You assigned the data element class to the extended data record.

Step 7

Navigate back to the **Extended Data** tab of the **Dem** page.

Step 8

In the **Extended Data Class List** table, open the **DemExtendedDataClass_0** entry.

Step 9

In the **Extended Data Record** table, add an entry.

Step 10

In the **Extended Data Record** drop-down list, select **EXTDATAREC_0**.

You added the extended data record to the extended data class list. This extended data class list is used in the event list.



Configuring a freeze frame

In this step, you create a freeze frame. The freeze frame contains error information.

Step 1

In the top right corner of the editor view, click **Home**.

Step 2

Click the **Freeze Frames** tab.

Step 3

In the **Data Id (DID)** table, add an entry.

A **DID_0** entry is added to the table. This entry is a data-ID.

Step 4

Open the **DID_0** entry.

Step 5

In the **Did Data Element Class List** table, add an entry.

Step 6

In the **Did Data Element Class** drop-down list box, select **DemDataElementClass_0**.

You assigned the data element class to the extended data record.



Step 7

Navigate back to the **Freeze Frames** tab of the **Dem** page.

Step 8

In the **Freeze Frame Class List** table, add an entry and open it.

Step 9

In the **Did Class List** table, add an entry.

Step 10

In the **Did Class** drop-down list box, select **DID_0**.

You added the data-ID to the freeze frame class list. This freeze frame class list is used in the event list.



Creating an event list

In this step, you create an event list. The event list, among others, contains the configuration of the DTCs, extended data record, and freeze frame.

Step 1

Click **Home**.

Step 2

Click the **Configuration Set List** tab.

Step 3

In the **Multiple Configuration: Configuration Set List** table, add an entry and open it.

Step 4

In the **DTC Class List** table, add an entry.

Step 5

Click the **Event List** tab.

Step 6

In the **Event List** table, add an entry.

An **APPL_E_0** entry is added to the table. This entry is the event list.



Configuring the DTCs and events

In the configured **APPL_E_0** event list, you create a DTC class and reference the extended data class and freeze frame class that you created in the previous step. Furthermore, you enable aging and configure the event failure cycle.

Step 1

In the **Event List** table, open the **APPL_E_0** entry.

Step 2

In the **DTC Class** drop-down list box, select **DemDTCCClass_0**.



You referenced the DTC configuration.

Step 3

In the **Extended Data Class** drop-down list box, select **DemExtendedDataClass_0**.

You referenced the extended data class that contains the extended data records.

Step 4

In the **Freeze Frame Class** drop-down list box, select **DemFreezeFrameClass_0**.

You referenced the freeze frame class that contains the data-IDs.

Step 5

Click the **Event Class** tab.

Step 6

Select the **Aging Allowed** check box.

Step 7

Select the **Event Failure Cycle Counter Threshold (1 -> 254)** check box.

Step 8

Select the **Event Failure Cycle** check box.

Step 9

In the **Event Failure Cycle** drop-down list box, select **DEM_OPCYC_POWER**.

Step 10

In the **Operation Cycle Reference** drop-down list box, select **DEM_OPCYC_POWER**.



Configuring the non-volatile memory block

The **NvM** module enables the **Dem** module to persistently store diagnostic events on non-volatile memory. The events are stored in NVRAM blocks. In this step, you create the required NVRAM blocks in the **NvM** module and link the NVRAM blocks to the **Dem** module.

In this tutorial, only the **Dem** part of the NVRAM blocks configuration is performed. The **NvM** module configuration is not part of this tutorial.

Step 1

In the **Dem** editor view, click **Home**.

Step 2

Click the **NVRAM Storage** tab.

Step 3

In the **NVRAM Block List** table, add an entry.

Step 4

Rename the **DEM_NVM_BLOCK_ID_0** entry to **DEM_NVM_BLOCK_ID_DEFAULT**. This enables you to automatically configure the required NVRAM blocks by using the **Calculate Service Needs** wizard.



Step 5

In the menu bar of EB tresos Studio, select **Project → Unattended Wizards → Calculate Service Needs(SvcAs_Trigger)**.

The **Calculate Service Needs** wizard creates the required NVRAM blocks in the `NvM` module and references them in the `Dem` module.

Step 6

In the **NVRAM Block** drop-down list box, select **NVM_BLOCK_DEM_DEFAULT**.

6.5. Configuring the data routing and communication mode (EcuC, PduR, ComM)

For the data routing and communication mode, the following modules are configured:

- ▶ EcuC:
 - ▶ 2 PDUs are created.
- ▶ PduR:
 - ▶ The 2 PDUs are specified as the routing source and the routing destination.
 - ▶ The `Dcm` interface is enabled.
- ▶ ComM:
 - ▶ 1 CAN communication channel.

For more information on the `EcuC` module, see the EB tresos AutoCore Generic 8 Base documentation.

For more information on the `PduR` module, see the EB tresos AutoCore Generic 8 COM Services documentation.

For more information on the `ComM` module, see the EB tresos AutoCore Generic 8 Mode Management documentation.



Adding the EcuC, PduR, and ComM modules to the project

In this step, you add the modules to the project that are required for the data routing and the communication mode handling.

The data routing is managed by the `PduR` module. You add this module to the project with a *recommended configuration*. The recommended configuration that you use in this step contains a specific configuration of the `PduR` module. It is configured for use within a CAN network.



Step 1

In the **Project Explorer**, right-click the **DiagStack_ECU** configuration node.

Step 2

In the context menu, click **Module Configurations....**

Step 3

Add the following modules to your project:

- ▶ EcuC
- ▶ PduR
- ▶ ComM

Step 4

In the **Module Configurations** table area, click the **PduR** entry.

Step 5

In the **Recommended Configuration** drop-down list box in the **Details of** view, select **PduRRecConfigurationCanEcu**.

Step 6

Click **OK**.



Creating the PDUs in the EcuC

Protocol data units (PDUs) represent data packages that are used for data transmission. In this step, you create two PDUs. One PDU is used to send diagnostic service requests from a tester to the ECU. The second PDU is used to send back the requested response from the ECU to the tester.

Step 1

In the **Project Explorer**, double-click the **EcuC** configuration node.

In the editor view, the **EcuC** editor opens up.

Step 2

Click the **Pdu** tab.

Step 3

Select the **EcucPduCollection** check box.

Step 4

In the **Pdu** table, add two entries.

A **Pdu_0** and **Pdu_1** entry is added to the table.

Step 5

In the **PduLength** text box, enter 8 for both PDU entries.

You created two PDUs with 8 bytes.



Configuring interfaces with other modules in the PduR

The PDU Router module (`PduR`) specifies the data flow between two modules, e.g. between the bus interface module `CanTp` and the `Dcm` module. You already added the `Dcm` with a recommended configuration for a CAN interface. This recommended configuration contains configured interfaces to the `Dcm` and other modules of the Communication Stack. Since this tutorial is focused on the Diagnostic Stack modules, you disable all interfaces except the `Dcm` interface in this step.

Step 1

In the **Project Explorer**, double-click the **PduR** configuration node.

In the editor view, the **PduR** editor opens up.

Step 2

Click the **PduRBswModules** tab.

Step 3

In the **PduRBswModules** table, clear the **Enable Module** check box for the following entries:

- ▶ Com
- ▶ CanIf
- ▶ CanTp
- ▶ IpduM

NOTE

Configuration of Communication Stack modules



In a real project, you would configure the modules that are listed above. However, since these modules are not part of this tutorial, you can disable them to avoid errors.



Configuring the PDU routing in the PduR

In this step, you define the routing path between the `Dcm` and another module. To do so, you specify one of the PDUs that you created in the `EcuC` module as the routing source and the other as the routing destination. This enables the `PduR` to send a service request from a lower layer module to the `Dcm`.

Step 1

Click the **PduRRoutingTables** tab.

Step 2

In the **Multiple Configuration: PduRRoutingTables** table, add an entry and open it.

Step 3

Click the **PduRRoutingTable** tab.



Step 4

In the **PduRRoutingTable** table, add an entry and open it. The routing table contains the routing paths.

Step 5

Click the **PduRRoutingPath** tab.

Step 6

In the **PduRRoutingPath** table, add an entry and open it.

Step 7

In the **PduRSourcePduHandleId (0 -> 65535)** text box, enter 0.

Step 8

In the **PduRSrcPduRef** drop-down list box, select **Pdu_0**.

The **Pdu_0** is configured as the routing source.

Step 9

Click the **PduRDestPdu** tab.

Step 10

In the **PduRDestPdu** table, add an entry.

Step 11

In the **Dest PDU Ref.** drop-down list box, select **Pdu_1**.

The **Pdu_1** is configured as the routing destination.



Configuring the communication mode in the ComM

In this step, you enable the **ComM** module to provide the **Dcm** module with a communication mode.

Step 1

In the **Project Explorer**, double-click the **ComM** configuration node.

In the editor view, the **ComM** editor opens up.

Step 2

Click the **ComMConfigSet** tab.

Step 3

In the **Multiple Configuration: ComMConfigSet** table, add an entry and open it.

Step 4

Click the **Users** tab.

Step 5

In the **Users** table, add an entry.

A **ComMUser_0** entry is added to the table.

Step 6

Click the **Network Channel** tab.



Step 7

In the **Network Channel** table, add an entry.

A **ComMChannel_0** entry is added to the table. This entry is a communication channel.

Step 8

Open the **ComMChannel_0** entry.

Step 9

In the **Network Management Variant** drop-down list box, select **NONE**.

Step 10

Click the **User Per Channel** tab.

Step 11

In the **User Per Channel** table, add an entry.

Step 12

In the **User Reference** drop-down list box, select **ComMUser_0**.

6.6. Configuring the Diagnostic Communication Manager (Dcm)

At the beginning of this tutorial, you added the **Dcm** with a recommended configuration. This recommended configuration already contains most of the regular UDS-based diagnostic services, e.g. the *ReadDTCInformation* service. The configuration also contains diagnostic sessions and DIDs. Therefore, you do not need to configure diagnostic services, diagnostic sessions, or DIDs in this tutorial. If you need information on these **Dcm** components, see [Section 4.4.3, “Configuring the Dcm module”](#).

In this chapter, you configure the communication channel of the **Dcm**, i.e. what PDU is used to receive a diagnostic service request or to transmit a response.

For more information on the **Dcm** module, see [Section 4.4.2, “Background information”](#).



Configuring the reception and transmission channels of the Dcm

Step 1

In the **Project Explorer**, double-click the **Dcm** configuration node.

In the editor view, the **Dcm** editor opens up.

Step 2

Click the **Configuration Set List** tab.

Step 3

Open the **DcmConfigSet_0** entry.



Step 4

Click the **Diagnostic Service Layer** tab.

Step 5

Open the **DcmDsIProtocolRow_0** entry.

Step 6

Open the **DcmDsIConnection_0** entry. This entry is the main connection. The main connection contains the communication channels.

Step 7

In the **Reference to Tx PdulId** drop-down list box, select **Pdu_0**.

The transmission channel of the `Dcm` is configured.

Step 8

In the **Receive Channels** table, delete the **DcmDsIProtocolRx_1** entry. In this tutorial, only the physical reception channel is used.

Step 9

In the **Reference to the ComMChannel** drop-down list box, select **ComMChannel_0**.

You added a CAN communication channel to the reception channel.

Step 10

In the **Reference to Rx PdulId** drop-down list box, select **Pdu_1**.

The reception channel of the `Dcm` is configured.

6.7. Configuring the Function Inhibition Manager (Fim)

In this chapter, you enable the `Fim` module to react to diagnostic events and to disable functions that depend on the diagnostic events.

For more information on the `Fim` module, see [Section 4.6.2, “Background information”](#).



Configuring the function inhibition

Step 1

In the **Project Explorer**, double-click the **Fim** configuration node.

In the editor view, the **Fim** editor opens up.

Step 2

Click the **Configuration Set List** tab.

**Step 3**

In the **Multiple Configuration: Configuration Set List** table, add a new entry and open it.

Step 4

Click the **FID List** tab.

Step 5

In the **FID List** table, add a new entry.

A **FID_0** entry is added to the table. This entry represents 1 function that is monitored by the **Fim** module.

Step 6

Click the **Inhibition Configuration List** tab.

Step 7

In the **Inhibition Configuration List** table, add a new entry and open it.

Step 8

In the **Inhibition Function Id** drop-down list box, select created **FID_0**.

Step 9

In the **Inhibition Dem Event** drop-down list box, select **APPL_E_0**.

You referenced the event list that triggers the function inhibition.

You configured a Diagnostic Stack with some of its basic functions. You learned some basics of the Diagnostic Stack and how to configure a project in EB tresos Studio.