Document Title	Specification of Diagnostic Event Manager
Document Category	Software specification (SWS)
Document UID	19
Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R26-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2026-11-26	R26-11	AUTOSAR Release Management	• Added J1939 individual-DTC clearing (DM22) and severity reporting (DM01); corrected DTC-specific lamp status (DM31). • Added OBD DTC internal data elements and clarified fault detection counter data. • Clarified PTO handling and added PID $1E reporting. • Removed event combination on retrieval and the TestFailed status-bit storage option. • Updated API, callback and notification specifications, including removal of obsolete version-information references. • Added and clarified configuration constraints for OBD relevance, event availability and diagnostic data elements. • Minor corrections and editorial changes.
2025-11-27	R25-11	AUTOSAR Release Management	• Added support for handling the initialization of Dem-related NvM data blocks. • Added support for 64-bit DiagnosticDataElements. • Enabled post-build configuration for DTC priority, aging, and OBD-related parameters. • Removed the 'Dem aging cycle counter threshold for TFSLC' feature.

△

			△ • Added support for J1939 DTC class filtering. • Added an overlay for Std_ReturnType using Dem literals. • Added configuration to freeze the monitor numerator if the denominator is inhibited. • Clarified the requirements for Monitor Status.
2024-11-27	R24-11	AUTOSAR Release Management	• Support of FunctionalGroupIdentifier 0x33 (OBD) • Configurability of the ControlDTCSetting action to specific event memories
2023-11-23	R23-11	AUTOSAR Release Management	• Remove draft API Dem_SetEventFailedWithSyncFreeze Frame / Dem_CallbackEventSync StorageProcessed • Introduce DemResetPendingBitOnOverflow • Introduce IUMPR Denominator for evaporative system purge flow monitor • Adapt range for EDR 0xFE
2022-11-24	R22-11	AUTOSAR Release Management	• Add Interface Intrusion Detection System Manager • Support J1979-02 use case • Rename DemDebounceTimeBaseClass to DemDebounceTimeBasedClass • Rename DemDebounceTimeBaseRef to DemDebounceTimeBasedClassRef • Add API Dem_ResetMonitorStatus

▽

△

2021-11-25	R21-11	AUTOSAR Release Management	• OBD on UDS supported • SAE J2012_4 DTCs and UDS DTCs seperated • Add Pending / Busy in Dem_GetDTCSuppression / Dem_SetDTCSuppression • Removal of the Mirror Memory following ISO 14229-1:2020 • Support subfunctions 1A and 56 for UDS services 0x19 • Limit OBD DTCs to primary fault memory
2020-11-30	R20-11	AUTOSAR Release Management	• Referenced DemEventFailureCycle-CounterThreshold removed • Remove Dem_J1939DcmClearDTC from SWS_Dem_01101 • Add Dem_ GetDTCSelectionResultForClearDTC to the list of API's in SWS_Dem_01253 • Minor corrections / clarifications / editorial changes
2019-11-28	R19-11	AUTOSAR Release Management	• Diagnostic SRS renamed from SRS_DIAG_xxxx to RS_DIAG_xxxx • DataType of size parameter increased for FF and ED APIs • Inconsistencies and contradictions within the AR4.3.1 specifications of Dem and Dcm has been removed • Minor corrections / clarifications / editorial changes , For details please refer to the Change Documentation • Changed Document Status from Final to published

▽

△

2018-10-31	4.4.0	AUTOSAR Release Management	• Clarified asynchronous API behavior • IUMPR denominators are locked at operation cycle start • Support of typed C/S interfaces • minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2017-12-08	4.3.1	AUTOSAR Release Management	• Update and optimized interaction between Dcm and Dem • Made many functions asynchronous • Shifted constraint handling to explicit requirements instead of informative text in ECUC tables • minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2016-11-30	4.3.0	AUTOSAR Release Management	• Removal of context workarounds and reworked callback behaviour. Introduced monitor status and updated point in time of callback processing. • Introduced client concept for multiple access to the Dem. Optimized APIs for better runtime performance and aligned return values to allow the Dcm mapping to return values according to ISO 14229-1 [1]. • Supporting event memories for multiple diagnostic servers • Clarified thresholds and operation cycle handling • minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation

▽

△

2015-07-31	4.2.2	AUTOSAR Release Management	• New APIs Dem_GetEventFreezeFrameDataEx and Dem_GetEventExtendedDataRecordEx with buffersize as parameter and corrected return value defintions. • Providing OBD FreezFrame for UDS service 0x19 0x05 • ISO 14229-1:2013[2] NRC handling for service 0x14 • Refined service interfaces for DataElements • minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2014-10-31	4.2.1	AUTOSAR Release Management	• Support of ISO 27145 (WWH-OBD / Euro VI)[3] • Update to support ISO 14229-1:2013[1] • Introduction of event dependencies • Refined DTC/Event suppression
2014-03-31	4.1.3	AUTOSAR Release Management	• Further clarification of event combination • Clarification of DTC groups • Editorial changes
2013-10-31	4.1.2	AUTOSAR Release Management	• Added API table for service interfaces • Clarification of event combination • Editorial changes • Removed chapter(s) on change documentation
2013-03-15	4.1.1	AUTOSAR Administration	• Incorporated new and reporting of several Dem internal data elements • Supported J1939 • Supported OBD • Reworked Dem/Dcm interface • Introduced new debouncing behaviour and debounce counter storage • Extended clear DTC functionality • Supported event suppression • Extended DTC storage behavior

▽

△

			△ • Incorporation of user-controlled warning indicator requestet bit • Incorporation of autostart behaviour for operation cycles
2011-12-22	4.0.3	AUTOSAR Administration	• Introduced multiple formats per DTC • Reworked Dem_ResetEventStatus behavior • Reworked Dlt interaction • Reworked Dem/Dcm interface • Corrected include-structure and RTE interfaces • Refined several aspects on features
2009-12-18	4.0.1	AUTOSAR Administration	• Reworked Dem/Dcm interface • Extended definition of ”Diagnostic Monitor” • Introduced "Event significance" and "DTC suppression" • Reworked OBD (esp. interface for service $02, readiness, and permanent memory) • Reworked file-structure • Finalization of issues on Revision 1
2009-12-04	3.1.4	AUTOSAR Administration	• Improved and extended of Dem functional description (especially: status bit handling, event displacement, debouncing, fault confirmation and indicator handling, event combination, enable- & storage conditions, event related data, operation cycle management) • Introduced the approach for functional diagnostics of SW-Cs • Added Dlt interaction and debugging interface • Document structure reworked and extended • Legal disclaimer revised

▽

△

2008-08-13	3.1.1	AUTOSAR Administration	• Document structure reworked and extended • Add APIs and configuration parameters for OBD support • Improve interaction between DCM and software components • Legal disclaimer revised
2007-12-21	3.0.1	AUTOSAR Administration	• Improvement of RTE compliance • Improvement of configuration part • Improvement of document structure • Rework and tightening of data type useage • New API added • Document meta information extended • Small layout adaptations made
2007-01-24	2.1.115	AUTOSAR Administration	• “Advice for users” revised • “Revision Information” added
2006-11-28	2.1	AUTOSAR Administration	• Complete rework and extension of debouncing part • Dem_ClearGroupOfDTC and Dem_ClearSingleDTC replaced by Dem_SingleDTC • Dem_GetNextFilteredDTCAndFDC, Dem_SetDTCFilterForRecords, Dem_GetSizeOfFreezeFrame, Dem_SetValueByOemId, Dem_SetEnableCondition, Xxx_DemGetFaultDetectionCounter added • DTCTranslationType replaced by DTCKind in several APIs • Chapter "Service DEM" added • Function IDs reworked • File Structure reworked and extended • Configuration chapter reworked and extended • Dem_GetNextFilteredDTC reworked • Legal disclaimer revised

▽

△

2006-05-16	2.0	AUTOSAR Administration	• Layout Adaptations

Disclaimer

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

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

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

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Introduction and functional overview 24

2 Acronyms and Abbreviations 25
2.1 Acronyms . . . 25
2.2 Abbreviations . . . 28

3 Related documentation 30
3.1 Input documents & related standards and norms . . . 30
3.2 Related specification . . . 31

4 Constraints and assumptions 32
4.1 Limitations . . . 32
4.2 Applicability to car domains . . . 33

5 Dependencies to other modules 34
5.1 File structure . . . 35
5.1.1 Code file structure . . . 35
5.1.2 Header file structure . . . 35
5.2 Integration Hints . . . 35

6 Requirements traceability 37

7 Functional specification 44
7.1 Startup behavior . . . 44
7.2 Monitor re-initialization . . . 44
7.3 Diagnostic event definition . . . 47
7.3.1 Event priority . . . 51
7.3.2 Event occurrence . . . 51
7.3.3 Event kind . . . 52
7.3.4 Event destination . . . 52
7.3.5 Diagnostic monitor definition . . . 53
7.3.6 Event dependencies . . . 53
7.3.7 Component availability . . . 55
7.4 Diagnostic trouble code definition . . . 56
7.4.1 DTC kind . . . 58
7.4.2 DTC format . . . 58
7.4.3 DTC groups . . . 60
7.4.4 DTC severity . . . 61
7.4.5 Functional unit . . . 62
7.4.6 DTC significance . . . 62
7.4.7 Suppress DTC output . . . 63
7.4.8 Availability of events (visibility and computation) . . . 65
7.5 Monitored Component Definition . . . 69
7.5.1 Definition of components and dependencies . . . 70

7.6 Operation cycle management . . . 71
7.6.1 Operation Cycle Counters . . . 74
7.6.1.1 Cycles since last failed . . . 74
7.6.1.2 Cycles since first failed . . . 75
7.6.1.3 Failed cycles . . . 76
7.7 Event memory description . . . 77
7.7.1 Event status management . . . 78
7.7.1.1 Status bit support . . . 82
7.7.1.2 Monitor status and UDS status update . . . 82
7.7.1.3 Status bit transitions . . . 86
7.7.1.4 Active/Passive status . . . 90
7.7.1.5 Notification of status bit changes . . . 90
7.7.2 Event memory management . . . 92
7.7.2.1 Event retention . . . 94
7.7.2.2 Clearing event memory entries . . . 97
7.7.2.3 Event memory overflow indication . . . 101
7.7.2.4 Event displacement . . . 102
7.7.2.5 Reporting order of event memory entries . . . 105
7.7.3 Debouncing of diagnostic events . . . 106
7.7.3.1 Counter based debounce algorithm . . . 107
7.7.3.2 Time based debounce algorithm . . . 113
7.7.3.3 Monitor internal debounce algorithm . . . 118
7.7.3.4 Debounce algorithm initialization and reset conditions . . . 118
7.7.3.5 Fault detection counter retrieval . . . 119
7.7.3.6 Fault detection counter reporting . . . 120
7.7.4 Fault confirmation . . . 122
7.7.4.1 Method for grouping of association of events for OBD purpose . . . 123
7.7.5 Event Combination . . . 123
7.7.5.1 Combination On Storage . . . 125
7.7.6 Enable and storage conditions of diagnostic events . . . 126
7.7.7 Event related data . . . 130
7.7.7.1 Storage of freeze frame data . . . 131
7.7.7.2 Pre-storage of freeze frame data . . . 136
7.7.7.3 Storage of extended data . . . 139
7.7.7.4 Configuration of Event related data . . . 141
7.7.7.5 Notification of data changes . . . 151
7.7.8 Aging of diagnostic events . . . 153
7.7.9 Healing of diagnostic events . . . 157
7.7.9.1 Warning indicator handling . . . 158
7.7.9.2 User controlled WarningIndicatorRequested-bit . . . 159
7.7.9.3 Handling of the warning indicator lamp (MIL) . . . 161
7.7.9.4 Notification and Set of the warning indicator status . . . 162
7.8 BSW Error Handling . . . 162

7.9 OBD-specific functionality . . . 164
7.9.1 General overview and restrictions . . . 164
7.9.2 PIDs provided by Dem . . . 167
7.9.2.1 Centralized PID $21 / $31 / $4D / $4E handling . . . 168
7.9.2.2 PID $1E handling . . . 170
7.9.3 Readiness status . . . 171
7.9.4 In-Use-Monitor Performance Ratio (IUMPR) Support . . . 172
7.9.5 OBD for light duty (OBD2) . . . 178
7.9.5.1 Service $01 Read Current Powertrain Diagnostic Data . . . 178
7.9.5.2 Service $02 Read Powertrain Freeze Frame Data . . . 178
7.9.5.3 Service $03 - Read Emission-Related Powertrain Diagnostic Trouble Codes . . . 180
7.9.5.4 Service $04 - Clear Reset Emission-Related Diagnostic Information . . . 180
7.9.5.5 Service $06 - Support of central DTR handling . . . 181
7.9.5.5.1 DTR visibility . . . 185
7.9.5.6 Service $07 - Read emission-related diagnostic trouble codes detected during during current or last completed driving cycle . 185
7.9.5.7 Read Emission-Related Diagnostic Trouble Codes with Permanent Status . . . 186
7.9.5.8 MIL Handling . . . 186
7.9.6 UDS based OBD protocols . . . 186
7.9.6.1 DTC WWHOBD class . . . 187
7.9.6.2 Service $19 42 - Read WWH-OBD DTC By Mask Record . . . 187
7.9.6.3 Service $14 FFFF33 - Clear Emission Related DTCs . . . 188
7.9.6.4 WWH-OBD . . . 188
7.9.6.4.1 Continuous-MI counter . . . 188
7.9.6.4.2 Cumulative Continuous-MI counter (Master ECU) . . . 189
7.9.6.4.3 Class B1 counter . . . 189
7.9.6.4.4 Activation Mode . . . 190
7.9.6.4.5 Freeze Frame 0x00 . . . 191
7.9.6.4.6 Extended Data Record 0x90 . . . 191
7.9.6.4.7 Aging . . . 192
7.9.6.5 OBDonUDS . . . 192
7.9.6.5.1 Legislated FreezeFrame storage for OBDonUDS . . . 192
7.10 J1939 specific functionality . . . 193
7.10.1 Read DTC . . . 193
7.10.1.1 Composite Malfunction Indicator Lamp Status . . . 194
7.10.1.2 Composite Red Stop Lamp Status . . . 195
7.10.1.3 Composite Amber Warning Lamp Status . . . 196
7.10.1.4 Composite Protect Lamp Status . . . 196
7.10.1.5 DTC data acqusition . . . 197
7.10.2 Clear DTCs . . . 200
7.10.3 DM31 . . . 201

7.10.3.1 Malfunction Indicator Lamp . . . 202
7.10.3.2 Red Stop Lamp . . . 202
7.10.3.3 Amber Warning Lamp . . . 203
7.10.3.4 Protect Lamp . . . 204
7.10.3.5 J1939 lamp status . . . 205
7.10.4 FreezeFrame . . . 205
7.10.4.1 SPNs in ExpandedFreezeFrame . . . 207
7.10.5 Diagnostic Readiness . . . 207
7.10.6 Monitor Performance Ratio . . . 209
7.11 Interaction with other Software Modules . . . 210
7.11.1 Interaction with Software Components (SW-C) . . . 210
7.11.2 Interaction with Diagnostic Clients . . . 211
7.11.2.1 Parallel event memory access . . . 212
7.11.2.2 Accessing diagnostic fault memory . . . 213
7.11.2.3 Access DTCs and Status Information . . . 216
7.11.2.4 Access event related data . . . 223
7.11.2.5 Clear diagnostic information . . . 228
7.11.2.6 Control DTC setting . . . 228
7.11.2.7 Asynchronous Dcm operations . . . 229
7.11.3 Interaction with J1939 Diagnostic Manager (J1939) . . . 229
7.11.4 Interaction with Function Inhibition Manager (FIM) . . . 229
7.11.5 Interaction with NVRAM Manager (NvM) . . . 230
7.11.5.1 Initialize NvM Data . . . 232
7.11.6 Interaction with Default Error Tracer (Det) . . . 233
7.11.7 Required data by the Dem module . . . 233
7.11.8 Scaling information on Service Interfaces . . . 234
7.12 Error classification . . . 236
7.12.1 Development Errors . . . 237
7.12.2 Runtime Errors . . . 238
7.12.3 Production Errors . . . 238
8 API specification 239
8.1 Imported Types . . . 241
8.2 Type definitions . . . 242
8.2.1 Dem data types . . . 242
8.2.1.1 Dem_ComponentIdType . . . 242
8.2.1.2 Dem_ConfigType . . . 242
8.2.1.3 Dem_EventIdType . . . 243
8.2.1.4 Dem_EventStatusType . . . 243
8.2.1.5 Dem_DebouncingStateType . . . 243
8.2.1.6 Dem_DebounceResetStatusType . . . 243
8.2.1.7 Dem_UdsStatusByteType . . . 243
8.2.1.8 Dem_IndicatorStatusType . . . 243
8.2.1.9 Dem_MonitorDataType . . . 243

8.2.1.10 Dem_MonitorStatusType . . . 244
8.2.1.11 Dem_DTCKindType . . . 244
8.2.1.12 Dem_DTCFormatType . . . 244
8.2.1.13 Dem_DTCOriginType . . . 244
8.2.1.14 Dem_DTCRequestType . . . 245
8.2.1.15 Dem_DTCTranslationFormatType . . . 245
8.2.1.16 Dem_DTCSeverityType . . . 246
8.2.1.17 Dem_RatioIdType . . . 246
8.2.1.18 Dem_DTRControlType . . . 246
8.2.1.19 Dem_InitMonitorReasonType . . . 246
8.2.1.20 Dem_IumprDenomCondIdType . . . 247
8.2.1.21 Dem_IumprDenomCondStatusType . . . 247
8.2.1.22 Dem_J1939DcmDTCStatusFilterType . . . 247
8.2.1.23 Dem_J1939DcmDTCSeverityType . . . 247
8.2.1.24 Dem_J1939DcmSetClearFilterType . . . 248
8.2.1.25 Dem_J1939DcmSetFreezeFrameFilterType . . . 248
8.2.1.26 Dem_J1939DcmLampStatusType . . . 249
8.2.1.27 Dem_J1939DcmDiagnosticReadiness1Type . . . 249
8.2.1.28 Dem_J1939DcmDiagnosticReadiness2Type . . . 250
8.2.1.29 Dem_J1939DcmDiagnosticReadiness3Type . . . 251
8.2.1.30 Dem_EventOBDReadinessGroupType . . . 252
8.2.2 Extension to Std ReturnType . . . 252
8.3 Function definitions . . . 253
8.3.1 Interface ECU State Manager <=> Dem . . . 253
8.3.1.1 Dem_PreInit . . . 253
8.3.1.2 Dem_Init . . . 253
8.3.1.3 Dem_Shutdown . . . 254
8.3.2 Interface BSW modules / SW-Components <=> Dem . . . 254
8.3.2.1 Dem_ClearDTC . . . 254
8.3.2.2 Dem_ClearPrestoredFreezeFrame . . . 255
8.3.2.3 Dem_GetComponentFailed . . . 256
8.3.2.4 Dem_GetDTCSelectionResult . . . 256
8.3.2.5 Dem_GetDTCSelectionResultForClearDTC . . . 257
8.3.2.6 Dem_GetEventUdsStatus . . . 258
8.3.2.7 Dem_GetMonitorStatus . . . 258
8.3.2.8 Dem_GetDebouncingOfEvent . . . 259
8.3.2.9 Dem_GetDTCOfEvent . . . 259
8.3.2.10 Dem_GetDTCSuppression . . . 260
8.3.2.11 Dem_GetEventAvailable . . . 261
8.3.2.12 Dem_GetFaultDetectionCounter . . . 261
8.3.2.13 Dem_GetIndicatorStatus . . . 262
8.3.2.14 Dem_GetEventFreezeFrameDataEx . . . 262
8.3.2.15 Dem_GetEventExtendedDataRecordEx . . . 263

8.3.2.16 Dem_GetEventMemoryOverflow . . . 264
8.3.2.17 Dem_GetNumberOfEventMemoryEntries . . . 264
8.3.2.18 Dem_ResetEventDebounceStatus . . . 265
8.3.2.19 Dem_ResetEventStatus . . . 266
8.3.2.20 Dem_ResetMonitorStatus . . . 266
8.3.2.21 Dem_RestartOperationCycle . . . 267
8.3.2.22 Dem_PrestoreFreezeFrame . . . 267
8.3.2.23 Dem_SelectDTC . . . 268
8.3.2.24 Dem_SetComponentAvailable . . . 268
8.3.2.25 Dem_SetDTCSuppression . . . 269
8.3.2.26 Dem_SetEnableCondition . . . 269
8.3.2.27 Dem_SetEventAvailable . . . 270
8.3.2.28 Dem_SetEventConfirmationThresholdCounter . . . 271
8.3.2.29 Dem_SetEventStatus . . . 271
8.3.2.30 Dem_SetEventStatusWithMonitorData . . . 272
8.3.2.31 Dem_SetStorageCondition . . . 272
8.3.2.32 Dem_SetWIRStatus . . . 273
8.3.2.33 Dem_InitAllNvmBlocks . . . 274
8.3.2.34 Dem_InitAllNvmBlocksGetResult . . . 274
8.3.3 Interface Dcm <=> Dem . . . 275
8.3.3.1 Access DTCs and Status Information . . . 275
8.3.3.1.1 Dem_GetTranslationType . . . 275
8.3.3.1.2 Dem_GetDTCStatusAvailabilityMask . . . 275
8.3.3.1.3 Dem_GetStatusOfDTC . . . 276
8.3.3.1.4 Dem_GetSeverityOfDTC . . . 277
8.3.3.1.5 Dem_GetFunctionalUnitOfDTC . . . 277
8.3.3.1.6 Dem_SetDTCFilter . . . 278
8.3.3.1.7 Dem_GetNumberOfFilteredDTC . . . 279
8.3.3.1.8 Dem_GetNextFilteredDTC . . . 280
8.3.3.1.9 Dem_GetNextFilteredDTCAndFDC . . . 281
8.3.3.1.10 Dem_GetNextFilteredDTCAndSeverity . . . 281
8.3.3.1.11 Dem_SetFreezeFrameRecordFilter . . . 282
8.3.3.1.12 Dem_GetNextFilteredRecord . . . 283
8.3.3.1.13 Dem_GetDTCByOccurrenceTime . . . 283
8.3.3.1.14 Dem_SetDTCFilterByExtendedDataRecordNumber . . . 284
8.3.3.1.15 Dem_SetDTCFilterByReadinessGroup . . . 285
8.3.3.2 Access extended data records and FreezeFrame data . . . 285
8.3.3.2.1 Dem_DisableDTCRecordUpdate . . . 285
8.3.3.2.2 Dem_EnableDTCRecordUpdate . . . 286
8.3.3.2.3 Dem_GetSizeOfExtendedDataRecordSelection . . . 287
8.3.3.2.4 Dem_GetSizeOfFreezeFrameSelection . . . 287
8.3.3.2.5 Dem_GetNextExtendedDataRecord . . . 288
8.3.3.2.6 Dem_GetNextFreezeFrameData . . . 289

8.3.3.2.7 Dem_SelectExtendedDataRecord . . . 289
8.3.3.2.8 Dem_SelectFreezeFrameData . . . 290
8.3.3.2.9 Dem_GetNumberOfFreezeFrameRecords . . . 291
8.3.3.3 DTC storage . . . 291
8.3.3.3.1 Dem_DisableDTCSetting . . . 291
8.3.3.3.2 Dem_EnableDTCSetting . . . 292
8.3.4 OBD-specific Dcm <=> Dem Interfaces . . . 293
8.3.4.1 Dem_DcmGetInfoTypeValue08 . . . 293
8.3.4.2 Dem_DcmGetInfoTypeValue0B . . . 293
8.3.4.3 Dem_DcmReadDataOfPID01 . . . 294
8.3.4.4 Dem_DcmReadDataOfPID1C . . . 295
8.3.4.5 Dem_DcmReadDataOfPID1E . . . 295
8.3.4.6 Dem_DcmReadDataOfPID21 . . . 296
8.3.4.7 Dem_DcmReadDataOfPID30 . . . 296
8.3.4.8 Dem_DcmReadDataOfPID31 . . . 297
8.3.4.9 Dem_DcmReadDataOfPID41 . . . 297
8.3.4.10 Dem_DcmReadDataOfPID4D . . . 298
8.3.4.11 Dem_DcmReadDataOfPID4E . . . 299
8.3.4.12 Dem_DcmReadDataOfPID91 . . . 299
8.3.4.13 Dem_DcmReadDataOfOBDFreezeFrame . . . 300
8.3.4.14 Dem_DcmGetDTCOfOBDFreezeFrame . . . 300
8.3.4.15 Dem_DcmGetAvailableOBDMIDs . . . 301
8.3.4.16 Dem_DcmGetNumTIDsOfOBDMID . . . 302
8.3.4.17 Dem_DcmGetDTRData . . . 302
8.3.4.18 Dem_DcmGetInfoTypeValue79 . . . 303
8.3.4.19 Dem_DcmReadDataOfPIDF501 . . . 304
8.3.5 Interface J1939Dcm <=> Dem . . . 304
8.3.5.1 Access DTCs and Status Information . . . 304
8.3.5.1.1 Dem_J1939DcmSetDTCFilter . . . 304
8.3.5.1.2 Dem_J1939DcmGetNumberOfFilteredDTC . . . 305
8.3.5.1.3 Dem_J1939DcmGetNextFilteredDTC . . . 306
8.3.5.1.4 Dem_J1939DcmGetNextFilteredDTCWithSeverity . . . 306
8.3.5.1.5 Dem_J1939DcmFirstDTCwithLampStatus . . . 307
8.3.5.1.6 Dem_J1939DcmGetNextDTCwithLampStatus . . . 308
8.3.5.2 DTC storage . . . 308
8.3.5.2.1 Dem_J1939DcmClearDTC . . . 308
8.3.5.2.2 Dem_J1939DcmClearSingleDTC . . . 309
8.3.5.2.3 Dem_J1939DcmSetFreezeFrameFilter . . . 310
8.3.5.2.4 Dem_J1939DcmGetNextFreezeFrame . . . 311
8.3.5.2.5 Dem_J1939DcmGetNextSPNInFreezeFrame . . . 312
8.3.5.3 Reporting . . . 312
8.3.5.3.1 Dem_J1939DcmSetRatioFilter . . . 312
8.3.5.3.2 Dem_J1939DcmGetNextFilteredRatio . . . 313

8.3.5.3.3 Dem_J1939DcmReadDiagnosticReadiness1 314
8.3.5.3.4 Dem_J1939DcmReadDiagnosticReadiness2 314
8.3.5.3.5 Dem_J1939DcmReadDiagnosticReadiness3 315
8.3.6 OBD-specific Interfaces 315
8.3.6.1 Dem_SetEventDisabled 315
8.3.6.2 Dem_RepIUMPRFaultDetect 316
8.3.6.3 Dem_SetIUMPRDenCondition 316
8.3.6.4 Dem_GetIUMPRDenCondition 317
8.3.6.5 Dem_RepIUMPRDenRelease 318
8.3.6.6 Dem_SetPtoStatus 318
8.3.6.7 Dem_ReadDataOfPID01 319
8.3.6.8 Dem_GetDataOfPID21 319
8.3.6.9 Dem_SetDataOfPID21 320
8.3.6.10 Dem_SetDataOfPID31 320
8.3.6.11 Dem_SetDataOfPID4D 321
8.3.6.12 Dem_SetDataOfPID4E 321
8.3.6.13 Dem_GetCycleQualified 322
8.3.6.14 Dem_SetCycleQualified 322
8.3.6.15 Dem_GetDTCSeverityAvailabilityMask 323
8.3.6.16 Dem_GetB1Counter 323
8.3.6.17 Dem_SetDTR 324
8.4 Scheduled functions 324
8.4.1 Dem_MainFunction 324
8.4.2 Runnable Entity MainFunction 325
8.5 Expected interfaces 325
8.5.1 Mandatory interfaces 325
8.5.2 Optional interfaces 326
8.5.3 Callout interfaces 326
8.5.4 Configurable interfaces 326
8.5.4.1 Interface BSW modules / SW-Components <=> Dem 327
8.5.4.1.1 InitMonitorForEvent 328
8.5.4.1.2 DemTriggerOnComponentStatus 328
8.5.4.2 ClearDtcNotification 329
8.5.4.3 DemGeneralTriggerOnMonitorStatus 329
8.5.4.4 DemGeneralTriggerOnEventUdsStatus 330
8.5.4.5 DemTriggerOnEventUdsStatus 330
8.5.4.6 DemTriggerOnDTCStatus 331
8.5.4.7 DemTriggerOnMonitorStatus 331
8.5.4.8 EventDataChanged 332
8.5.4.9 ClearEventAllowed 332
8.5.4.10 ReadDataElement 333
8.5.4.11 GetFaultDetectionCounter 334
8.6 Service Interfaces 335

8.6.1 Implementation Data Types . . . 335
8.6.1.1 Dem_DataType . . . 335
8.6.1.2 Dem_DataPrimitiveType . . . 336
8.6.1.3 Dem_DataArrayType . . . 337
8.6.1.4 Dem_DTCOriginType . . . 338
8.6.1.5 Dem_DebouncingStateType . . . 339
8.6.1.6 Dem_DebounceResetStatusType . . . 339
8.6.1.7 Dem_DTRControlType . . . 340
8.6.1.8 Dem_EventIdType . . . 340
8.6.1.9 Dem_EventStatusType . . . 341
8.6.1.10 Dem_DTCFormatType . . . 341
8.6.1.11 Dem_InitMonitorReasonType . . . 342
8.6.1.12 Dem_IumprDenomCondIdType . . . 342
8.6.1.13 Dem_IumprDenomCondStatusType . . . 343
8.6.1.14 Dem_MaxDataValueType . . . 344
8.6.1.15 Dem_MonitorDataType . . . 344
8.6.1.16 Dem_MonitorStatusType . . . 344
8.6.1.17 Dem_IndicatorStatusType . . . 345
8.6.1.18 Dem_PID21valueType . . . 345
8.6.1.19 Dem_PID31valueType . . . 346
8.6.1.20 Dem_RatioIdType . . . 346
8.6.1.21 Dem_UdsStatusByteType . . . 347
8.6.1.22 Dem_PID4DvalueType . . . 347
8.6.1.23 Dem_PID4EvalueType . . . 348
8.6.2 Sender-Receiver-Interfaces . . . 348
8.6.2.1 DataServices_{Data} . . . 348
8.6.3 Client-Server-Interfaces . . . 348
8.6.3.1 CallbackClearEventAllowed . . . 348
8.6.3.2 CallbackComponentStatusChanged . . . 349
8.6.3.3 CallbackDTCStatusChange . . . 350
8.6.3.4 CallbackEventDataChanged . . . 350
8.6.3.5 CallbackEventUdsStatusChanged . . . 351
8.6.3.6 CallbackGetFaultDetectCounter . . . 352
8.6.3.7 CallbackInitMonitorForEvent . . . 352
8.6.3.8 CallbackMonitorStatusChange . . . 353
8.6.3.9 ClearDtcNotification . . . 353
8.6.3.10 ClearDTC . . . 354
8.6.3.11 DTCSuppression . . . 355
8.6.3.12 DataServices_{Data} . . . 357
8.6.3.13 DTRCentralReport . . . 358
8.6.3.14 DiagnosticInfo . . . 358
8.6.3.15 DiagnosticMonitor . . . 362
8.6.3.16 DiagnosticMonitor_MonitorData . . . 364

8.6.3.17 EnableCondition . . . 364
8.6.3.18 EventAvailable . . . 365
8.6.3.19 EventFailureCycleCounterThreshold . . . 366
8.6.3.20 EvMemOverflowIndication . . . 366
8.6.3.21 EventStatus . . . 367
8.6.3.22 GeneralCallbackEventDataChanged . . . 368
8.6.3.23 GeneralCallbackEventUdsStatusChanged . . . 368
8.6.3.24 GeneralCallbackMonitorStatusChanged . . . 369
8.6.3.25 GeneralDiagnosticInfo . . . 370
8.6.3.26 GetDataOfPID21 . . . 373
8.6.3.27 IndicatorStatus . . . 374
8.6.3.28 IUMPRDenominator . . . 375
8.6.3.29 IUMPRDenominatorCondition . . . 375
8.6.3.30 IUMPRNumerator . . . 376
8.6.3.31 OperationCycle . . . 376
8.6.3.32 PowerTakeOff . . . 377
8.6.3.33 SetDataOfPID21 . . . 378
8.6.3.34 SetDataOfPID31 . . . 378
8.6.3.35 SetDataOfPID4D . . . 379
8.6.3.36 SetDataOfPID4E . . . 380
8.6.3.37 StorageCondition . . . 380
8.6.4 Ports . . . 381
8.6.4.1 CBClrEvt . . . 381
8.6.4.2 CBDataEvt . . . 381
8.6.4.3 CBFaultDetectCtr . . . 382
8.6.4.4 CBInitEvt . . . 382
8.6.4.5 CBStatusDTC . . . 383
8.6.4.6 CBEventUdsStatusChanged . . . 383
8.6.4.7 CBMonitorStatusChanged . . . 384
8.6.4.8 CBStatusComp . . . 384
8.6.4.9 ClearDTC . . . 384
8.6.4.10 ClearDtcNotification . . . 385
8.6.4.11 ControlDTCSuppression . . . 385
8.6.4.12 EventAvailable . . . 385
8.6.4.13 ControlEventFailureCycleCounterThreshold . . . 386
8.6.4.14 DataServices_{Data} . . . 386
8.6.4.15 DTR . . . 386
8.6.4.16 EnableCond . . . 387
8.6.4.17 Event . . . 387
8.6.4.18 DiagnosticMonitor_MonitorData . . . 387
8.6.4.19 EventStatus . . . 388
8.6.4.20 EventInfo . . . 388
8.6.4.21 GeneralCBDataEvt . . . 388

8.6.4.22 GeneralCBMonitorStatusChanged . . . 389
8.6.4.23 GeneralCBStatusEvt . . . 389
8.6.4.24 GeneralEvtInfo . . . 389
8.6.4.25 IndStatus . . . 389
8.6.4.26 IUMPRDenominator . . . 390
8.6.4.27 IUMPRDenominatorCondition . . . 390
8.6.4.28 IUMPRNumerator . . . 391
8.6.4.29 OpCycle . . . 391
8.6.4.30 OverflowIndPrimaryMemory . . . 391
8.6.4.31 OverflowIndUserDefinedMemory . . . 392
8.6.4.32 PowerTakeOffStatus . . . 392
8.6.4.33 GetDataOfPID21 . . . 393
8.6.4.34 SetDataOfPID21 . . . 393
8.6.4.35 SetDataOfPID31 . . . 393
8.6.4.36 SetDataOfPID4D . . . 394
8.6.4.37 SetDataOfPID4E . . . 394
8.6.4.38 StorageCond . . . 394

9 Sequence Diagrams 395
9.1 ControlDTCSetting . . . 395
9.2 Dem_ClearDTC . . . 395
9.3 Dem_GetDTCByOccurrenceTime . . . 397
9.4 Dem_GetNextExtendedDataRecord . . . 397
9.5 Dem_DcmGetStatusOfDTC . . . 398
9.6 Retrieving freeze frames . . . 398
9.7 GetOBDFaultInformation . . . 399
9.8 J1939DcmGetNextFilteredDTC . . . 400
9.9 ReportDTCByStatusMask . . . 400
9.10 FiM_DemTriggerOnMonitorStatus . . . 401
9.11 ProcessEvent (Example) . . . 402
9.12 Retrieving DTCs based on supported extended data records . . . 403
9.13 Retrieving DTCs assigned to a DTC readiness group . . . 403
9.14 Retrieving stored snapshot records for all DTCs . . . 404

10 Configuration specification 405
10.1 How to read this chapter . . . 405
10.2 Containers and configuration parameters . . . 405
10.2.1 Dem . . . 405
10.2.2 General . . . 406
10.2.2.1 DemGeneral . . . 406
10.2.2.2 DemConfigSet . . . 418
10.2.2.3 DemClient . . . 420
10.2.2.4 DemDTCAttributes . . . 423
10.2.2.5 DemEventParameter . . . 429

10.2.2.6 DemMultiEventTriggering . . . 441
10.2.2.7 DemComponent . . . 442
10.2.2.8 DemDTC . . . 444
10.2.2.9 DemGroupOfDTC . . . 450
10.2.2.10 DemOperationCycle . . . 451
10.2.2.11 DemIndicator . . . 452
10.2.2.12 DemIndicatorAttribute . . . 453
10.2.2.13 DemNvRamBlockId . . . 456
10.2.3 OBD . . . 457
10.2.3.1 DemGeneralOBD . . . 457
10.2.3.2 DemObdDTC . . . 467
10.2.3.3 DemRatio . . . 471
10.2.3.4 DemDtrs . . . 477
10.2.3.5 DemDtr . . . 477
10.2.3.6 DemPidClass . . . 482
10.2.3.7 DemPidDataElement . . . 483
10.2.4 J1939 . . . 484
10.2.4.1 DemGeneralJ19139 . . . 484
10.2.4.2 DemJ1939FreezeFrameClass . . . 490
10.2.4.3 DemSPNClass . . . 491
10.2.5 Conditions . . . 493
10.2.5.1 DemEnableCondition . . . 493
10.2.5.2 DemEnableConditionGroup . . . 494
10.2.5.3 DemStorageCondition . . . 495
10.2.5.4 DemStorageConditionGroup . . . 497
10.2.6 Event Memory . . . 498
10.2.6.1 DemEventMemorySet . . . 498
10.2.6.2 DemPrimaryMemory . . . 503
10.2.6.3 DemUserDefinedMemory . . . 507
10.2.7 Debouncing . . . 511
10.2.7.1 DemDebounceAlgorithmClass . . . 511
10.2.7.2 DemDebounceCounterBased . . . 512
10.2.7.3 DemDebounceCounterBasedClass . . . 513
10.2.7.4 DemDebounceTimeBased . . . 519
10.2.7.5 DemDebounceTimeBasedClass . . . 520
10.2.7.6 DemDebounceMonitorInternal . . . 522
10.2.8 Callbacks . . . 523
10.2.8.1 DemCallbackClearEventAllowed . . . 523
10.2.8.2 DemCallbackDTCStatusChanged . . . 524
10.2.8.3 DemCallbackJ1939DTCStatusChanged . . . 525
10.2.8.4 DemCallbackOBDDTCStatusChanged . . . 526
10.2.8.5 DemCallbackGetFDC . . . 527
10.2.8.6 DemCallbackEventDataChanged . . . 528

10.2.8.7 DemCallbackEventUdsStatusChanged . . . 529
10.2.8.8 DemCallbackMonitorStatusChanged . . . 530
10.2.8.9 DemCallbackInitMForE . . . 531
10.2.8.10 DemClearDTCNotification . . . 532
10.2.9 Event related data . . . 533
10.2.9.1 DemFreezeFrameClass . . . 533
10.2.9.2 DemFreezeFrameRecordClass . . . 535
10.2.9.3 DemFreezeFrameRecNumClass . . . 538
10.2.9.4 DemExtendedDataClass . . . 539
10.2.9.5 DemExtendedDataRecordClass . . . 541
10.2.10 Data elements . . . 543
10.2.10.1 DemDataElementClass . . . 543
10.2.10.2 DemDataElementInstance . . . 544
10.2.10.3 DemDidClass . . . 545
10.2.10.4 DemInternalDataElementClass . . . 546
10.2.10.5 DemExternalCSDataElementClass . . . 550
10.2.10.6 DemExternalSRDataElementClass . . . 553
10.2.10.7 DemSRDataElementClass . . . 556
10.2.10.8 DemSubElementInDataElementInstance . . . 556
10.2.10.9 DemDiagnosisScaling . . . 557
10.2.10.10 DemAlternativeDataInterface . . . 558
10.2.10.11 DemAlternativeDiagnosticDataElement . . . 559
10.2.10.12 DemAlternativeDataType . . . 560
10.2.10.13 DemTextTableMapping . . . 561
10.3 Published information . . . 562
A Not Applicable Requirements 563
B Change history of AUTOSAR traceable items 564
B.1 Traceable item history of this document according to AUTOSAR Release R26-11 . . . 564
B.1.1 Added Specification Items in R26-11 . . . 564
B.1.2 Changed Specification Items in R26-11 . . . 564
B.1.3 Deleted Specification Items in R26-11 . . . 574
B.1.4 Added Constraints in R26-11 . . . 574
B.1.5 Changed Constraints in R26-11 . . . 575
B.1.6 Deleted Constraints in R26-11 . . . 575
B.2 Traceable item history of this document according to AUTOSAR Release R25-11 . . . 575
B.2.1 Added Specification Items in R25-11 . . . 575
B.2.2 Changed Specification Items in R25-11 . . . 576
B.2.3 Deleted Specification Items in R25-11 . . . 577
B.2.4 Added Constraints in R25-11 . . . 578
B.2.5 Changed Constraints in R25-11 . . . 578
B.2.6 Deleted Constraints in R25-11 . . . 578

B.3 Traceable item history of this document according to AUTOSAR Release R24-11 . . . 579
B.3.1 Added Specification Items in R24-11 . . . 579
B.3.2 Changed Specification Items in R24-11 . . . 579
B.3.3 Deleted Specification Items in R24-11 . . . 580
B.3.4 Added Constraints in R24-11 . . . 580
B.3.5 Changed Constraints in R24-11 . . . 580
B.3.6 Deleted Constraints in R24-11 . . . 581
B.4 Traceable item history of this document according to AUTOSAR Release R23-11 . . . 581
B.4.1 Added Specification Items in R23-11 . . . 581
B.4.2 Changed Specification Items in R23-11 . . . 581
B.4.3 Deleted Specification Items in R23-11 . . . 582
B.4.4 Added Constraints in R23-11 . . . 582
B.4.5 Changed Constraints in R23-11 . . . 582
B.4.6 Deleted Constraints in R23-11 . . . 582

1 Introduction and functional overview

The service component Diagnostic Event Manager (Dem) is responsible for processing and storing diagnostic events (errors) and associated data. Further, the Dem provides fault information to the Dcm (e.g. read all stored DTCs from the event memory). The Dem offers interfaces to the application layer and to other BSW modules.

The basic target of the Dem specification document is to define the ability for a common approach of a "diagnostic fault memory" for automotive manufacturers and component suppliers

This specification defines the functionality, API and the configuration of the AUTOSAR basic software module Diagnostic Event Manager (Dem). Parts of the internal behavior are manufacturer specific and described in the Limitations chapter.

2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the Dem module that are not included in the [4, AUTOSAR glossary].

2.1 Acronyms

Acronym	Description
Activation Mode 1	Absence of malfunction - The MIL shall blink for one flash.
Activation Mode 2	"on-demand-MI" - The MIL shall show blink for two flashes if the OBD system would command an on-demand-MI according to the discriminatory display strategy.
Activation Mode 3	"short-MI" - The MIL shall blink for three flashes if the OBD system would command a short-MI according to the discriminatory display strategy.
Activation Mode 4	"continuous-MI" - The MIL shall remain continuously ON ("continuous-MI") if the OBD system would command a continuous-MI according to the discriminatory display strategy.
Aging	Unlearning/deleting of a no longer failed event/DTC after a defined number of operation cycles from event memory.
Aging Counter	The "Aging Counter" or "Aging Cycle Counter" or "DTC Aging Counter" specifies the counter which is used to perform Aging. It counts the number of operation cycles until an event/DTC is removed from event memory.
Class B1 counter	Number of engine hours during which a Class B1 malfunction has been Confirmed and TestFailed.
Combined DTC	Normal DTC, but referenced by multiple events reported by several monitors (e.g. ECU Defect, consisting of different HW defects).
Continuous-MI counter	Hours run by the engine while a continuous MI is commanded.
Cumulative Continuous-MI counter	Number of engine hours during which MI has been continuously commanded to be on during its lifetime.
Debounce counter	Internal counter for counter-based debouncing algorithm(s).
DemComponent / Monitored Component	A monitored component is a part of the system, which is checked for proper operation by one or several monitorings. (see chapter 7.5)
Dem-internal data value	Some data values (e.g. the occurrence counter) are calculated by the Dem module itself internally.
Denominator	The denominator of a specific monitor m (Denominatorm) is a counter indicating the number of vehicle driving events, taking into account conditions specific to that specific monitor.
Dependent / Secondary ECUs	Dependent / Secondary (or dep. / sec.) ECUs are always related to a Master or a Primary ECU.
Directed acyclic graph	Dependency graph without circular dependencies.
Displacement	Replacing the the most insignificant event memory entry by a more significant event memory entry which needs to be stored.

Acronym	Description
DTC group	Uniquely identifies an set of DTCs. A DTC group is mapped into the range of valid DTCs. By providing a group of DTCs it is expressed that a certain operation is requested on all DTCs of that group. The DTC group definition is provided by ISO 14229-1[2] and OEM/supplier specific.
DtcGroupAllDtcs	Grouping of all configured DTCS (representation is 0xFFFFFF).
Event combination	Event combination is a method to merge several events to one specific combined DTC. It is used to adapt different monitor results to one significant fault, which is clearly evaluable in a service station.
Event debouncing	Debouncing is a specific mechanism (e.g. counter-based) to evaluate, if the diagnostic event gets qualified. This works on top of potential signal debouncing and can be done within the SW-C or inside the Dem.
Event confirmation	A diagnostic event is confirmed in case of repeated detection of qualified events over cycles or time evaluated by means of fault confirmation counters. Therefore, also the UDS status bit 3 (ConfirmedDTC) is set.
Event memory	An event memory (e.g. Primary memory) consists of several event memory entries.
Event memory entry	The event memory entry is a single storage container for an event and its event related data. The event memory entry is dynamically assigned to specific events.
Event memory overflow indication	The event memory overflow indication indicates, if this specific event memory is full and the next event occurs to be stored in this event memory.
Event qualification	A diagnostic event is qualified in case of a passed or a failed result is set (Dem-internal or reported from another BSW module or SW-C).
Event related data	Event related data is additional data, e.g. sensor values or time stamp/mileage, which is stored with an event in an event memory. ISO defines two types of event related data: freeze frames (snapshot records) and extended data.
Event status byte	Status byte as defined in ISO 14229-1 [1], based on event level.
Extended data record	An extended data record is a record to store specific information assigned to a fault.
Failure counter	The Failure counter represents the Trip Counter according to ISO14229-1 [2]. The Trip Counter counts the number of operation cycles (driving cycles) where a malfunction occurred. If the counter reaches the threshold (e.g., 2 driving cycles) the confirmed bit changes from 0 to 1.
Fault Detection Counter	sint8 value as used in ISO and FDC-APIs.
Freeze frame	Freeze frame is defined as a record of data (DIDs/PIDs). Freeze frames are the same as SnapShotRecords in ISO-14229-1[2].
General Denominator	The general denominator is a counter indicating the number of times a vehicle has been operated, taking into account general conditions.
Healing	Switching off the warning indicator including the handling of reported passed results over a period of time / several operation cycles
In-Use performance ratio	The in-use performance ratio (IUPR) of a specific monitor m of the OBD system is: IUPRm = Numeratorm / Denominatorm

Acronym	Description
Master ECU	As a primary ECU a Master ECU stores “it’s own” and “reported errors” of related dep. / sec ECUs in it’s event memory. Beside this a Master has to fulfill special Master tasks as MIL Master or provision of “general nominator” information.
Monitor	A diagnostic monitor is a routine entity determining the proper functionality of a component. Alternatively the term “diagnostic function” can be used.
Numerator	The numerator of a specific monitor m (Numeratorm) is a counter indicating the number of times a vehicle has been operated such that all monitoring conditions necessary for that specific monitor to detect a malfunction have been encountered.
NvM is marked for NvM_WriteAll	The Dem has called NvM_SetRamBlockStatus() to set the according NvM Block to be written by NvM_WriteAll()
Operating cycle	An ‘operation cycle’ is the base of the event qualifying and also Dem scheduling (e.g. ignition key off-on cycles, driving cycles, etc.)
OBD	On-Board Diagnostics, or OBD is a generic term referring to a vehicle’s self-diagnostic and reporting capability. OBD systems give the vehicle owner or a repair technician access to state of health information for various vehicle sub-systems.
OBD ECUs	"In a vehicle there can be 3 different types of OBD ECUs: • Master ECU (one per vehicle) • Primary ECUprimary ECUs (several per vehicle) • Dependent / Secondary ECUs (several per vehicle)
OBD UDS DTC separation	The feature will report a different 3 byte DTC number for SAE J1979-2 based UDS communication than for none J1979-2 UDS communication
P-Code	Power train code
PFC cycle	Permanent fault code - driving cycle (OBD Term)
Positive Callback from NvM	The Dem module shall use the APIs NvM_WriteBlock and NvM_GetErrorStatus of the NVRAMManager [5], if there is the necessity to store data between Dem_Init and Dem_Shutdown. Furthermore the API NvM_GetErrorStatus shall wait for positive response if writing of block completed successfully.
PossibleErrors	PossibleErrors means the ApplicationErrors as defined in meta model
Primary ECU	A primary ECU stores “it’s own” and "reported errors" of related dep. / sec ECUs in it’s event memory
RBM cycle	OBD Term: General Nominator / Rate-based monitoring - driving cycle (OBD Term)
Readiness	The readiness refers to the tested bits TestNotCompletedSinceLastClear (bit 4) and TestNotCompleteThisOperationCycle (bit 6) of the UDS status byte.
Triggered to NvM	The Dem module shall use the API NvM_WriteBlock of the NVRAMManager [5], if there is the necessity to trigger the storage of data between Dem_Init and Dem_Shutdown. Furthermore the Dem module shall wait for positive response of NvM_WriteBlock if request has been accepted.
UDS status bit 0	testFailed bit of the UDS status byte. Indicates the result of the most recently performed test.

Acronym	Description
UDS status bit 1	testFailedThisOperationCycle bit of the UDS status byte. Indicates whether or not a diagnostic test has reported a testFailed result at any time during the current operation cycle.
UDS status bit 2	pendingDTC bit of the UDS status byte. Indicates whether or not a diagnostic test has reported a testFailed result at any time during the current or last completed operation cycle.
UDS status bit 3	confirmedDTC bit of the UDS status byte. Indicates whether a malfunction was detected enough times to warrant that the DTC is desired to be stored in long-term memory.
UDS status bit 4	testNotCompletedSinceLastClear bit of the UDS status byte. Indicates whether a DTC test has ever run and completed since the last time a call was made to ClearDiagnosticInformation.
UDS status bit 5	testFailedSinceLastClear bit of the UDS status byte. Indicates whether a DTC test has completed with a failed result since the last time a call was made to ClearDiagnosticInformation.
UDS status bit 6	testNotCompletedThisOperationCycle bit of the UDS status byte. Indicates whether a DTC test has ever run and completed during the current operation cycle.
UDS status bit 7	warningIndicatorRequested bit of the UDS status byte. Report the status of any warning indicators associated with a particular DTC.
UDS status byte	Status byte as defined in ISO 14229-1 [1], based on DTC level.

2.2 Abbreviations

Abbreviation	Description
API	Application Programming Interface
BSW	Basic Software
CDD	Complex Device Driver
CRC	Cyclic Redundancy Check
Dcm	Diagnostic Communication Manager
Dem	Diagnostic Event Manager
Det	Default Error Tracer
DID	Data Identifier
DTC	Diagnostic Trouble Code
DTR	Diagnostic Test Result
DYC	OBD Term: Driving Cycle (OBD Term)
ECU	Electronic Control Unit
EcuM	Electronic Control Unit Manager
FDC	Fault Detection Counter
Fim	Function Inhibition Manager
FMI	Failure Mode Indicator (SAE J1939)
FTB	Failure Type Byte
HW	Hardware
ID	Identification/Identifier
ISO	International Standardization Organization
IUMPR	In Use Monitoring Performance Ratio (OBD Term)
J1939Dcm	SAEJ1939 Diagnostic Communication Manager
MIL	Malfunction Indicator Light (SAE J1979) or Lamp (SAE J1939)
NVRAM	Non volatile RAM
OBD	On-Board-Diagnostics

Abbreviation	Description
OC	Occurrence Count (SAE J1939)
OEM	Original Equipment Manufacturer (Automotive Manufacturer)
OS	Operating System
PID	Parameter Identification (SAE J1587 or SAE J1979)
PTO	Power Take Off
RAM	Random Access Memory
ROM	Read-only Memory
RTE	Runtime Environment
SPN	Suspect Parameter Number (SAE J1939)
SSCP	synchronous server call point
SW	Software
SW-C	Software Component
UDS	Unified Diagnostic Services
VOBD	Vehicle On-Board-Diagnostic
WUC	OBD Term: Warm up cycle (OBD Term)
WIR	Warning Indicator Request
WWH-OBD	World Wide Harmonized On-Board-Diagnostic

3 Related documentation

3.1 Input documents & related standards and norms

[1] ISO 14229-1 – Unified diagnostic services (UDS) – Part 1: Specification and requirements (Release 2006-12)
https://www.iso.org

[2] ISO 14229-1(2013) – Unified diagnostic services (UDS) – Part 1: Application layer (Release 2013-03)
https://www.iso.org

[3] ISO 27145-1:2012 – Road vehicles – Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements – Part 1: General information and use case definition
https://www.iso.org/standard/46273.html

[4] Glossary
AUTOSAR_FO_TR_Glossary

[5] Specification of NVRAM Manager
AUTOSAR_CP_SWS_NVRAMManager

[6] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[7] Specification of a Diagnostic Communication Manager for SAE J1939
AUTOSAR_CP_SWS_SAEJ1939DiagnosticCommunicationManager

[8] Requirements on Function Inhibition Manager
AUTOSAR_CP_RS_FunctionInhibitionManager

[9] Specification of Diagnostic Communication Manager
AUTOSAR_CP_SWS_DiagnosticCommunicationManager

[10] Requirements on Diagnostics
AUTOSAR_FO_RS_Diagnostics

[11] General Requirements on Basic Software Modules
AUTOSAR_CP_RS_BSWGeneral

[12] ISO 15031-5 – Road vehicles – Communication between vehicle and external equipment for emission-related diagnostic – Part 5: Emission-related diagnostic services.
https://www.iso.org/standard/66368.html

[13] SAE J2012-DA Digital

[14] SAE J1939-73 Application Layer – Diagnostics

[15] ISO 11992-4 – Road vehicles – Interchange of digital information on electrical connections between towing and towed vehicles – Part 4: Diagnostic communica-

tion
https://www.iso.org

[16] ISO 14229-1(2020) – Unified diagnostic services (UDS) – Part 1: Application layer (Release 2020-02)
https://www.iso.org

[17] SAE J1979 – E/E Diagnostic Test Modes

[18] SAE J1979-DA Digital Annex of E/E Diagnostic Test Modes

[19] Title 13, California Code Regulations, Section 1971.1, On-Board Diagnostic System Requirements for 2013 and Subsequent Model-Year Heavy-Duty Engines (HD OBD)

[20] Title 13, California Code Regulations, Section 1968.2, Malfunction and Diagnostic System Requirements for 2004 and Subsequent Model-Year Passenger Cars, Light-Duty Trucks, and Medium-Duty Vehicles and Engines (OBD II) (Biennial Review MY08-11)
http://www.arb.ca.gov/regact/obdii06/19682clean.pdf

[21] ISO 27145-3:2012 – Road vehicles – Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements – Part 3: Common message dictionary
https://www.iso.org/standard/46277.html

[22] Software Component Template
AUTOSAR_CP_TPS_SoftwareComponentTemplate

[23] Diagnostic Extract Template
AUTOSAR_CP_TPS_DiagnosticExtractTemplate

[24] Basic Software Module Description Template
AUTOSAR_CP_TPS_BSWModuleDescriptionTemplate

3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules [6, SWS BSW General], which is also valid for Diagnostic Event Manager.

Thus, the specification SWS BSW General shall be considered as additional and required specification for Diagnostic Event Manager.

4 Constraints and assumptions

Some of the synchronous API calls defined within the Dem might take variable time to complete. Thus, this variable time must be considered in the system configuration.

[SWS_Dem_00126]

Upstream requirements: RS_Diag_04002

⌈There shall only be one Dem module available per ECU.⌋

The Dem can have multiple different sections of `event memory`. The mapping of a DTC to the respective section is done with the parameter DTC Origin. A specific ECU's Dem is only accessible by software components located inside the same ECU.

4.1 Limitations

Timing constrains have to be considered for the whole ECU. If there are explicit needs for faster responses from the Dem than the Dem basic cycle time, special measures have to be implemented, that are not specified in this AUTOSAR document. This is especially the case in ECUs with many events.

The handling of infrastructure errors reported by the RTE during Dem <=> SW-C interactions is missing from the SWS and might have to be taken into account by implementers if they need it.

The Dem is able to support additional event memories (permanent memory and used defined memory), but the specific `event memory` processing is not defined in detail.

Some details on the interaction between Dem and specific emission-related SW-C are not specified in this specification, since they are dependent on the SW-C implementation. The following functionality is not defined:

- Malfunction Indicator Lamp (MIL)-activation (MIL handler interaction to Dem, MIL-bulb check, readiness blinking, blinking in case of catalyst damaging misfire, etc.)
- misfire fault handling (debouncing over all cylinders, filtering single / multiple misfire faults)
- support of similar conditions for the specific `healing` of misfire and fuel system faults

Note: For OBD2, it is required that misfire and fuel system fault shall only be healed (yielding leaving Service $07) under the similar conditions as they have been detected. The "similarity" is derived from a "window" spanned by ranges on engine speed, engine load and temperature conditions being present at the time of fault detection.

For the handling of similar conditions it is assumed that the SW-C modules of misfire detection / fuelsystem diagnostics carry out the necessary computations themselves. That is, it is not globally solved within the Dem.

However, based on specific interface requirements of the respective misfire detection / fuelsystem diagnostics this may result in an extension of the Dem. Any further implementations on that need to be defined for a particular engine control unit project depending on the diagnostics and the Dem implementation.

The DTCStoredDataRecordNumber (absolute freeze frame record addressing functionality) is limited to 0x00 (OBD freeze frame, refer to Chapter 7.11.2.4).

The structure of a specific extended data record identified by its record number is unique per ECU.

The Dcm may lock the event memory update (refer to [SWS_Dem_00270]) while processing the “read diagnostic data” service, due to architectural design.

This specification covers a subset of SAE J1939 related diagnostic requirements (see Table 1: Supported DMx messages in [7, SWS J1939 Dcm]). The SPN Conversion Method is limited to Version 4 (CM = 0).

AUTOSAR only supports user defined memories with IDs from 0x10 to 0xFF as proposed to ISO 14229:1[2] by AUTOSAR.

Indicators are DemEventMemorySet local only. The Dem does not support configurations where indicators are used on more than one DemEventMemorySet.

4.2 Applicability to car domains

The Dem is designed to fulfill the design demands for ECUs with OBD requirements as well as for ECUs without OBD requirements. The immediate domains of applicability are currently body, chassis and powertrain ECUs. However, there is no reason why the Dem cannot be used to implement ECUs for other car domains like infotainment.

5 Dependencies to other modules

The AUTOSAR **Diagnostic Event Manager (Dem)** has interfaces and dependencies to the following Basic software modules and Software Components:

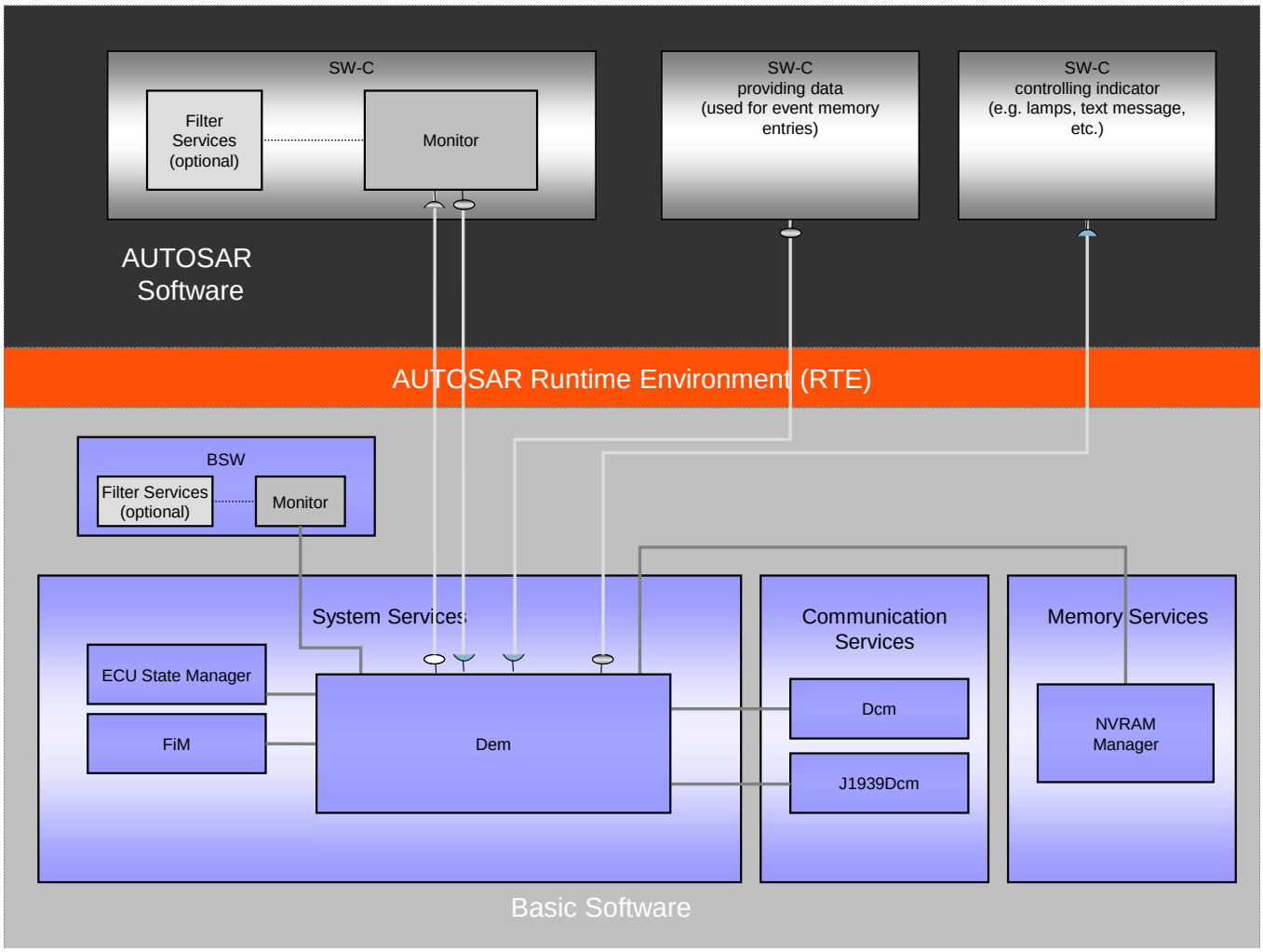


Figure 5.1: Dependencies of the Dem to other software modules

- The **Function Inhibition Manager (FiM)** (refer to [8, SWS FiM]) stands for the evaluation and assignment of events to the required actions for Software Components (e.g. inhibition of specific “Monitors”). The Dem informs and updates the Function Inhibition Manager (FiM) upon changes of the monitor status in order to stop or release function entities according to assigned dependencies.
- The **Diagnostic Communication Manager (Dcm)** (refer to [9, SWS Dcm]) is in charge of the UDS and SAE J1979 communication path and execution of diagnostic service resulting in the processing of diagnostic requests from an external tester or onboard test system. It forwards requests coming from an external diagnostic scan tool and is further responsible for assembly of response messages (DTC, status information, etc.) which will be transferred to the external diagnostic scan tool afterwards.

- The **Diagnostic Communication Manager for J1939 (J1939Dcm)** (refer to [7, SWS J1939 Dcm]) is in charge of the SAE J1939-73 diagnostics communication protocol.
- **Software-Components (SW-C)** and **Basic Software (BSW) modules** can access the Dem to update and/or retrieve current monitor status and UDS status information. SW-Cs and BSW modules can retrieve data from the Dem e.g. to turn the indicator lamps on or off. The monitor is a sub-component of a SW-C / BSW module.
- **Data Provider** SW-Cs and/or BSW modules will provide data (i.e. event related data) required by the Dem, for example, to be able to create event memory entries.
- The **NVRAM Manager (NvM)** (refer to [5]) provides mechanisms to store data blocks in NVRAM. NVRAM blocks (maximum size is a matter of configuration) are assigned to the Dem and used by the Dem to achieve permanent storage of UDS status information and associated data (e.g. over power-on reset).
- The **ECU State Manager (EcuM)** is responsible for the basic initialization and de-initialization of basic software components including Dem.
- The **RTE** implements scheduling mechanisms for BSW, e.g. assigns priority and memory protection to each BSW module used in an ECU.

5.1 File structure

5.1.1 Code file structure

For details refer to [6] Chapter 5.1.6 *"Code file structure"*, including subchapters.

5.1.2 Header file structure

The symbolic names (refer to TPS_ECUC_02108) are generated for configuration containers containing an identifier parameter, like event Id symbols, operation cycles, indicators, enable/storage conditions, etc.

Note, that also SW-C event Ids are published especially for complex device drivers, which may access on the status of specific SW-C events. SW-Cs use different ports to distinguish between different events.

5.2 Integration Hints

Due to cross-module communication it is recommend to map the Dem and J1939Dcm to the same Task to have for the RTE the correct call-context configured. There might be other BSW vendor-specific solutions possible allowing a more flexible mapping (e.g.

by describing this cross-module communication in the BSW internal-behavior or by a similar mechanism with the Dcm <-> Dem interaction).

6 Requirements traceability

The following table references the requirements specified in [10] as well as [11] and links to the fulfillment of these.

Requirement	Description	Satisfied by
[RS_Diag_04002]	The Diagnostic event (fault) management shall be established as Basic SW Module	[SWS_Dem_00126]
[RS_Diag_04024]	Access and handle specific data elements and data element groups if requested by an external scan tool	[SWS_Dem_00479] [SWS_Dem_01194]
[RS_Diag_04031]	Notify the Function Inhibition Manager (FIM) upon changes of the event status in order to process them according to the SW components dependencies	[SWS_Dem_00029] [SWS_Dem_01189] [SWS_Dem_01325] [SWS_Dem_01326] [SWS_Dem_01419]
[RS_Diag_04057]	Classification of events for series production, OBD and expert usage	[SWS_Dem_00057] [SWS_Dem_00518] [SWS_Dem_00649] [SWS_Dem_00906] [SWS_Dem_01066] [SWS_Dem_01362] [SWS_Dem_01363] [SWS_Dem_01364]
[RS_Diag_04058]	Ability to access different event memories	[SWS_Dem_00171]
[RS_Diag_04067]	Provide the diagnostic status information according to ISO 14229-1	[SWS_Dem_00006] [SWS_Dem_00011] [SWS_Dem_00016] [SWS_Dem_00051] [SWS_Dem_00059] [SWS_Dem_00060] [SWS_Dem_00187] [SWS_Dem_00212] [SWS_Dem_00213] [SWS_Dem_00330] [SWS_Dem_00385] [SWS_Dem_00386] [SWS_Dem_00389] [SWS_Dem_00390] [SWS_Dem_00391] [SWS_Dem_00392] [SWS_Dem_00393] [SWS_Dem_00394] [SWS_Dem_00395] [SWS_Dem_00409] [SWS_Dem_00421] [SWS_Dem_00431] [SWS_Dem_00523] [SWS_Dem_00524] [SWS_Dem_00529] [SWS_Dem_00566] [SWS_Dem_00638] [SWS_Dem_00823] [SWS_Dem_00936] [SWS_Dem_01186] [SWS_Dem_01276] [SWS_Dem_01277] [SWS_Dem_01278] [SWS_Dem_01280] [SWS_Dem_01281] [SWS_Dem_01282] [SWS_Dem_01283] [SWS_Dem_01284] [SWS_Dem_01285] [SWS_Dem_01286] [SWS_Dem_01287] [SWS_Dem_01342] [SWS_Dem_01395] [SWS_Dem_01426] [SWS_Dem_91008]



△

Requirement	Description	Satisfied by
[RS_Diag_04068]	The diagnostic in AUTOSAR shall support event specific debounce counters to improve signal quality internally (According to ISO 14229-1 Appendix D)	[SWS_Dem_00019] [SWS_Dem_00204] [SWS_Dem_00264] [SWS_Dem_00343] [SWS_Dem_00344] [SWS_Dem_00413] [SWS_Dem_00414] [SWS_Dem_00416] [SWS_Dem_00417] [SWS_Dem_00418] [SWS_Dem_00419] [SWS_Dem_00420] [SWS_Dem_00421] [SWS_Dem_00422] [SWS_Dem_00423] [SWS_Dem_00424] [SWS_Dem_00425] [SWS_Dem_00426] [SWS_Dem_00427] [SWS_Dem_00428] [SWS_Dem_00429] [SWS_Dem_00431] [SWS_Dem_00432] [SWS_Dem_00433] [SWS_Dem_00434] [SWS_Dem_00435] [SWS_Dem_00438] [SWS_Dem_00439] [SWS_Dem_00526] [SWS_Dem_00527] [SWS_Dem_00643] [SWS_Dem_00684] [SWS_Dem_00685] [SWS_Dem_00730] [SWS_Dem_00772] [SWS_Dem_00779] [SWS_Dem_00818] [SWS_Dem_00844] [SWS_Dem_00985] [SWS_Dem_01213] [SWS_Dem_01279] [SWS_Dem_01408]
[RS_Diag_04071]	Process events according to their defined importance like priority and/or severity	[SWS_Dem_00232] [SWS_Dem_00382] [SWS_Dem_00383] [SWS_Dem_00692] [SWS_Dem_01291] [SWS_Dem_01292] [SWS_Dem_01293] [SWS_Dem_01294] [SWS_Dem_01434] [SWS_Dem_01435]
[RS_Diag_04077]	Uses standard mechanisms provided by persistency modules	[SWS_Dem_00164] [SWS_Dem_00329]
[RS_Diag_04093]	Memory overflow indication	[SWS_Dem_00397] [SWS_Dem_00398] [SWS_Dem_00399] [SWS_Dem_00559] [SWS_Dem_01023]
[RS_Diag_04105]	Event memory management	[SWS_Dem_00580] [SWS_Dem_00607] [SWS_Dem_00683] [SWS_Dem_00781] [SWS_Dem_00783] [SWS_Dem_00784] [SWS_Dem_00785] [SWS_Dem_00786] [SWS_Dem_00922] [SWS_Dem_00923] [SWS_Dem_01000]
[RS_Diag_04107]	Provide defensive behavior	[SWS_Dem_00339]
[RS_Diag_04109]	Provide an interface to retrieve the number of event memory entries	[SWS_Dem_00651] [SWS_Dem_00652]
[RS_Diag_04110]	SAE J1939 lamp status	[SWS_Dem_00546] [SWS_Dem_00858] [SWS_Dem_00859] [SWS_Dem_00860] [SWS_Dem_00861] [SWS_Dem_00862] [SWS_Dem_00863] [SWS_Dem_00866] [SWS_Dem_00867] [SWS_Dem_00868] [SWS_Dem_00869] [SWS_Dem_00870] [SWS_Dem_00872] [SWS_Dem_00873] [SWS_Dem_00883] [SWS_Dem_00884] [SWS_Dem_00885] [SWS_Dem_00887] [SWS_Dem_00888] [SWS_Dem_00889] [SWS_Dem_00891] [SWS_Dem_00892] [SWS_Dem_00893] [SWS_Dem_00894] [SWS_Dem_00895] [SWS_Dem_00897] [SWS_Dem_00898] [SWS_Dem_00948] [SWS_Dem_00974] [SWS_Dem_01361]

▽

△

Requirement	Description	Satisfied by
[RS_Diag_04111]	SAE J1939 Expanded-FreezeFrame	[SWS_Dem_00877] [SWS_Dem_00899] [SWS_Dem_00900] [SWS_Dem_00901] [SWS_Dem_00902] [SWS_Dem_00904] [SWS_Dem_00905] [SWS_Dem_00906] [SWS_Dem_00907] [SWS_Dem_01384] [SWS_Dem_01385] [SWS_Dem_01386] [SWS_Dem_01387] [SWS_Dem_01388] [SWS_Dem_01433]
[RS_Diag_04112]	The DEM module shall support DTCs according to SAE J1939	[SWS_Dem_00645] [SWS_Dem_00845] [SWS_Dem_00855] [SWS_Dem_00856] [SWS_Dem_00864] [SWS_Dem_00874] [SWS_Dem_00875] [SWS_Dem_00910] [SWS_Dem_00911] [SWS_Dem_00912] [SWS_Dem_00933] [SWS_Dem_00945] [SWS_Dem_00946] [SWS_Dem_00970] [SWS_Dem_00971] [SWS_Dem_00976] [SWS_Dem_00977] [SWS_Dem_00978] [SWS_Dem_00979] [SWS_Dem_00981] [SWS_Dem_01432]
[RS_Diag_04113]	Support a set of SAE J1939 DM-messages	[SWS_Dem_00770] [SWS_Dem_00880] [SWS_Dem_00881] [SWS_Dem_00882] [SWS_Dem_00908] [SWS_Dem_00909] [SWS_Dem_00913] [SWS_Dem_00949] [SWS_Dem_00950] [SWS_Dem_00951] [SWS_Dem_00982] [SWS_Dem_00983] [SWS_Dem_01180] [SWS_Dem_01397] [SWS_Dem_01412] [SWS_Dem_01413] [SWS_Dem_01414] [SWS_Dem_01415] [SWS_Dem_01416] [SWS_Dem_01417] [SWS_Dem_01418] [SWS_Dem_01430]
[RS_Diag_04115]	The optional parameter DTCSetting ControlOptionRecord as part of UDS service ControlDTCSetting shall be limited to GroupOfDTC	[SWS_Dem_00080] [SWS_Dem_00626]
[RS_Diag_04117]	Configurable behavior for DTC deletion	[SWS_Dem_00343] [SWS_Dem_00620] [SWS_Dem_00670] [SWS_Dem_00679] [SWS_Dem_00680] [SWS_Dem_00879]
[RS_Diag_04118]	Optionally support event displacement	[SWS_Dem_00382] [SWS_Dem_00383] [SWS_Dem_00400] [SWS_Dem_00401] [SWS_Dem_00402] [SWS_Dem_00403] [SWS_Dem_00404] [SWS_Dem_00405] [SWS_Dem_00406] [SWS_Dem_00407] [SWS_Dem_00408] [SWS_Dem_00409] [SWS_Dem_00443] [SWS_Dem_00692] [SWS_Dem_01186] [SWS_Dem_01395]
[RS_Diag_04124]	Store the current debounce counter value nonvolatile to over a powerdown cycle	[SWS_Dem_00674] [SWS_Dem_00675] [SWS_Dem_00676]
[RS_Diag_04125]	Event debounce counter shall be configurable	[SWS_Dem_00625] [SWS_Dem_00654] [SWS_Dem_00655] [SWS_Dem_00656] [SWS_Dem_00677] [SWS_Dem_00678] [SWS_Dem_00681] [SWS_Dem_00682] [SWS_Dem_00788] [SWS_Dem_00790] [SWS_Dem_00791] [SWS_Dem_00792] [SWS_Dem_00794] [SWS_Dem_00795] [SWS_Dem_01004]

▽

△

Requirement	Description	Satisfied by
[RS_Diag_04126]	Configurable suppression of events	[SWS_Dem_00608] [SWS_Dem_00687] [SWS_Dem_01010] [SWS_Dem_01047] [SWS_Dem_01080] [SWS_Dem_01103] [SWS_Dem_01104] [SWS_Dem_01106] [SWS_Dem_01108] [SWS_Dem_01109] [SWS_Dem_01110] [SWS_Dem_01111] [SWS_Dem_01302] [SWS_Dem_01335] [SWS_Dem_01336] [SWS_Dem_91025]
[RS_Diag_04127]	Configurable record numbers and trigger options for DTCSnapshot Records and DTCExtendedData Records	[SWS_Dem_00002] [SWS_Dem_00261] [SWS_Dem_00334] [SWS_Dem_00337] [SWS_Dem_00461] [SWS_Dem_00468] [SWS_Dem_00471] [SWS_Dem_00581] [SWS_Dem_00797] [SWS_Dem_00798] [SWS_Dem_00799] [SWS_Dem_00800] [SWS_Dem_00801] [SWS_Dem_00802] [SWS_Dem_00803] [SWS_Dem_00804] [SWS_Dem_00808] [SWS_Dem_00809] [SWS_Dem_00810] [SWS_Dem_00811] [SWS_Dem_00812] [SWS_Dem_00813] [SWS_Dem_00814] [SWS_Dem_00815] [SWS_Dem_00816] [SWS_Dem_00817] [SWS_Dem_00819] [SWS_Dem_00820] [SWS_Dem_00991] [SWS_Dem_01068] [SWS_Dem_01069] [SWS_Dem_01070] [SWS_Dem_01081] [SWS_Dem_01308] [SWS_Dem_01351] [SWS_Dem_CONSTR_06166]
[RS_Diag_04129]	Provide OBD-specific configuration capabilities	[SWS_Dem_00752] [SWS_Dem_00932] [SWS_Dem_01220] [SWS_Dem_01221] [SWS_Dem_01222] [SWS_Dem_01234] [SWS_Dem_01235]
[RS_Diag_04131]	Consistent event management mechanisms	[SWS_Dem_01063]
[RS_Diag_04133]	Aging for event memory entries	[SWS_Dem_00019] [SWS_Dem_00442] [SWS_Dem_00489] [SWS_Dem_00490] [SWS_Dem_00492] [SWS_Dem_00493] [SWS_Dem_00498] [SWS_Dem_00698] [SWS_Dem_00824] [SWS_Dem_00985] [SWS_Dem_01054] [SWS_Dem_01185] [SWS_Dem_01214] [SWS_Dem_01215] [SWS_Dem_01219] [SWS_Dem_01221]
[RS_Diag_04137]	Definition of replacement failure	[SWS_Dem_01086] [SWS_Dem_01089] [SWS_Dem_CONSTR_06167]
[RS_Diag_04148]	Provide capabilities to inform applications about diagnostic data changes	[SWS_Dem_00259] [SWS_Dem_01426]
[RS_Diag_04150]	Support the primary fault memory defined by ISO 14229-1	[SWS_Dem_00242] [SWS_Dem_00243] [SWS_Dem_01202] [SWS_Dem_01203] [SWS_Dem_01206]
[RS_Diag_04151]	Event status handling	[SWS_Dem_00153] [SWS_Dem_00154] [SWS_Dem_00942] [SWS_Dem_01034] [SWS_Dem_01037] [SWS_Dem_01208] [SWS_Dem_01209]
[RS_Diag_04155]	Notify applications and BSW modules about updates of event related data	[SWS_Dem_00475] [SWS_Dem_00618] [SWS_Dem_00619] [SWS_Dem_01003]
[RS_Diag_04156]	Support DTCFunctionalUnit	[SWS_Dem_00594]
[RS_Diag_04159]	Control of DTC storage	[SWS_Dem_00243]
[RS_Diag_04160]	ResponseOnEvent according to ISO 14229-1	[SWS_Dem_00562]

▽

△

Requirement	Description	Satisfied by
[RS_Diag_04161]	Provide support for the ASMIP algorithm	[SWS_Dem_01242] [SWS_Dem_01243] [SWS_Dem_01244] [SWS_Dem_01245] [SWS_Dem_01246]
[RS_Diag_04162]	Parallel fault memory access	[SWS_Dem_01251] [SWS_Dem_01252] [SWS_Dem_01263]
[RS_Diag_04164]	Independent event memories for multiple diagnostic server instances (virtual ECUs)	[SWS_Dem_01247]
[RS_Diag_04165]	Triggering of multiple events upon a master event is reported	[SWS_Dem_01250] [SWS_Dem_01339]
[RS_Diag_04178]	Support operation cycles according to ISO 14229-1	[SWS_Dem_00019] [SWS_Dem_00673] [SWS_Dem_00693] [SWS_Dem_00698] [SWS_Dem_00773] [SWS_Dem_00826] [SWS_Dem_00968] [SWS_Dem_00985] [SWS_Dem_01224] [SWS_Dem_01225] [SWS_Dem_01316] [SWS_Dem_01317] [SWS_Dem_01318]
[RS_Diag_04179]	Provide interfaces for monitoring application.	[SWS_Dem_00331] [SWS_Dem_00756] [SWS_Dem_00759]
[RS_Diag_04183]	Notify interested parties about event status changes	[SWS_Dem_01426]
[RS_Diag_04185]	Notify applications about the clearing of an event	[SWS_Dem_01240] [SWS_Dem_01241]
[RS_Diag_04189]	Support a fine grained configuration for SnapshotRecords and Extended DataRecords	[SWS_Dem_00469] [SWS_Dem_00779] [SWS_Dem_00821] [SWS_Dem_00995] [SWS_Dem_01216]
[RS_Diag_04190]	Usage of internal data elements in SnapshotRecords and ExtendedData Records	[SWS_Dem_00469] [SWS_Dem_00470] [SWS_Dem_00471] [SWS_Dem_00472] [SWS_Dem_00473] [SWS_Dem_00592] [SWS_Dem_00643] [SWS_Dem_00673] [SWS_Dem_00775] [SWS_Dem_00779] [SWS_Dem_00817] [SWS_Dem_00818] [SWS_Dem_00819] [SWS_Dem_00820] [SWS_Dem_00821] [SWS_Dem_00822] [SWS_Dem_00984] [SWS_Dem_01043] [SWS_Dem_01044] [SWS_Dem_01045] [SWS_Dem_01084] [SWS_Dem_01310] [SWS_Dem_01311] [SWS_Dem_01350] [SWS_Dem_01365] [SWS_Dem_01366] [SWS_Dem_01367] [SWS_Dem_01402] [SWS_Dem_01403] [SWS_Dem_01404] [SWS_Dem_01405] [SWS_Dem_01406] [SWS_Dem_01407] [SWS_Dem_01427] [SWS_Dem_01428]
[RS_Diag_04192]	Provide the ability to handle event specific enable conditions	[SWS_Dem_00201] [SWS_Dem_00233] [SWS_Dem_00243] [SWS_Dem_00447] [SWS_Dem_00449] [SWS_Dem_00450] [SWS_Dem_00604] [SWS_Dem_00733] [SWS_Dem_01038]
[RS_Diag_04194]	ClearDTC shall be accessible for applications	[SWS_Dem_00515] [SWS_Dem_00516] [SWS_Dem_00570] [SWS_Dem_00571] [SWS_Dem_00572] [SWS_Dem_00573] [SWS_Dem_00661] [SWS_Dem_00667] [SWS_Dem_00668] [SWS_Dem_00669] [SWS_Dem_01295] [SWS_Dem_01332]

▽

△

Requirement	Description	Satisfied by
[RS_Diag_04195]	Chronological reporting order of the DTCs located in the configured event memory	[SWS_Dem_00161] [SWS_Dem_00219] [SWS_Dem_00221] [SWS_Dem_00410] [SWS_Dem_00411] [SWS_Dem_00412] [SWS_Dem_00477] [SWS_Dem_00695] [SWS_Dem_00696] [SWS_Dem_00787] [SWS_Dem_00935]
[RS_Diag_04200]	Support event combination	[SWS_Dem_00536]
[RS_Diag_04201]	Support a configuration to assign specific events to a customer specific DTC	[SWS_Dem_00198] [SWS_Dem_00231] [SWS_Dem_00269]
[RS_Diag_04204]	Provide the current status of each warning indicator.	[SWS_Dem_01361]
[RS_Diag_04205]	Support of SnapshotRecords	[SWS_Dem_00074] [SWS_Dem_00208] [SWS_Dem_00650] [SWS_Dem_01190] [SWS_Dem_01312] [SWS_Dem_01389]
[RS_Diag_04206]	Support of ExtendedDataRecords	[SWS_Dem_01398] [SWS_Dem_01399] [SWS_Dem_01400]
[RS_Diag_04211]	Persistent storage of DTC status and environmental data	[SWS_Dem_00341] [SWS_Dem_01237] [SWS_Dem_01238] [SWS_Dem_01420] [SWS_Dem_01421] [SWS_Dem_01422] [SWS_Dem_01423] [SWS_Dem_01424] [SWS_Dem_01425]
[RS_Diag_04214]	Support the user defined fault memories defined by ISO 14229-1	[SWS_Dem_01202] [SWS_Dem_01203] [SWS_Dem_01217]
[RS_Diag_04219]	Provide the ability to handle event specific storage conditions	[SWS_Dem_00455] [SWS_Dem_00459] [SWS_Dem_00556] [SWS_Dem_00605] [SWS_Dem_01015] [SWS_Dem_01046] [SWS_Dem_01113] [SWS_Dem_01309]
[RS_Diag_04220]	Support DTCs suppression	[SWS_Dem_00915] [SWS_Dem_01100] [SWS_Dem_01101] [SWS_Dem_01102] [SWS_Dem_01307] [SWS_Dem_01315]
[RS_Diag_04244]	Support sub-function 0x04 of UDS service 0x19.	[SWS_Dem_01348]
[RS_Diag_04245]	Support sub-function 0x06 of UDS service 0x19.	[SWS_Dem_01349]
[RS_Diag_04250]	Support subfunction 0x1A and 0x56 of UDS service 0x19	[SWS_Dem_01362] [SWS_Dem_01363] [SWS_Dem_01364] [SWS_Dem_91192]
[RS_Diag_04252]	Support of SAE J1979-2	[SWS_Dem_01297] [SWS_Dem_01365] [SWS_Dem_01366] [SWS_Dem_01367] [SWS_Dem_01368] [SWS_Dem_01369] [SWS_Dem_01409] [SWS_Dem_CONSTR_06175] [SWS_Dem_CONSTR_06176]
[RS_Diag_04253]	Support separated DTCs for UDS and OBD based on J1979-2	[SWS_Dem_01372] [SWS_Dem_01373] [SWS_Dem_01374] [SWS_Dem_01375] [SWS_Dem_01376] [SWS_Dem_01377] [SWS_Dem_01378]
[RS_Diag_04254]	Independent CP Software Cluster development	[SWS_Dem_91060]
[RS_Diag_04270]	Notify applications about the changes of enable conditions	[SWS_Dem_00256] [SWS_Dem_00682]
[RS_Diag_04274]	Variant handling at runtime	[SWS_Dem_00687] [SWS_Dem_01410] [SWS_Dem_01411]

▽

△

Requirement	Description	Satisfied by
[SRS_BSW_00101]	The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function	[SWS_Dem_00180] [SWS_Dem_00181] [SWS_Dem_00256] [SWS_Dem_00340]
[SRS_BSW_00171]	Optional functionality of a Basic-SW component that is not required in the ECU shall be configurable at pre-compile-time	[SWS_Dem_00255] [SWS_Dem_00579]
[SRS_BSW_00300]	Unique Name of Basic Software Modules	[SWS_Dem_00512]
[SRS_BSW_00301]	All AUTOSAR Basic Software Modules shall only import the necessary information	[SWS_Dem_00176]
[SRS_BSW_00310]	API naming convention	[SWS_Dem_00256]
[SRS_BSW_00336]	Basic SW module shall be able to shutdown	[SWS_Dem_00102] [SWS_Dem_00182] [SWS_Dem_00341]
[SRS_BSW_00337]	Classification of development errors	[SWS_Dem_00368]
[SRS_BSW_00339]	Reporting of production relevant error status	[SWS_Dem_00167] [SWS_Dem_00207] [SWS_Dem_01079] [SWS_Dem_01212] [SWS_Dem_01289]
[SRS_BSW_00369]	All AUTOSAR Basic Software Modules shall not return specific development error codes via the API	[SWS_Dem_01261] [SWS_Dem_01270] [SWS_Dem_01271] [SWS_Dem_01274] [SWS_Dem_01299] [SWS_Dem_01300] [SWS_Dem_01313]
[SRS_BSW_00373]	The main processing function of each AUTOSAR Basic Software Module shall be named according the defined convention	[SWS_Dem_00125]
[SRS_BSW_00377]	A Basic Software Module can return a module specific types	[SWS_Dem_91201]
[SRS_BSW_00406]	API handling in uninitialized state	[SWS_Dem_00124] [SWS_Dem_00169] [SWS_Dem_00364]
[SRS_BSW_00457]	Callback functions of Application software components shall be invoked by the Basis SW	[SWS_Dem_00003] [SWS_Dem_00284] [SWS_Dem_00613] [SWS_Dem_00986] [SWS_Dem_00987] [SWS_Dem_01005]
[SRS_Mem_08549]	The NVRAM manager shall provide functionality to automatically initialize RAM data blocks after a software update	[SWS_Dem_00578]
[SWS_BSW_00050]	No description	[SWS_Dem_00173]
[SWS_BSW_00212]	No description	[SWS_Dem_00173]

Table 6.1: Requirements Tracing

7 Functional specification

The Diagnostic Event Manager (Dem) handles and stores the events detected by diagnostic monitors in both Software Components (SW-Cs) and Basic software (BSW) modules. The stored event information is available via an interface to other BSW modules or SW-Cs.

DemGeneral contains the global part of the configuration and DemConfigSet contains the multiple configuration part.

7.1 Startup behavior

[SWS_Dem_00169]

Upstream requirements: SRS_BSW_00406

⌈The Dem module shall distinguish between a pre-initialization mode and a full-initialized mode (operation mode).⌋

[SWS_Dem_00180]

Upstream requirements: SRS_BSW_00101

⌈The function Dem_PreInit shall initialize the internal states of the Dem module necessary to process events and reset debounce counters reported by SW-C or BSW modules by using Dem_SetEventStatus and Dem_ResetEventDebounceStatus.⌋

The function Dem_PreInit is called by the ECU State Manager during the startup phase of the ECU before the NVRAM Manager is initialized. The BSW modules can start reporting of related events via Dem_SetEventStatus (refer to [SWS_Dem_00167]).

The function Dem_Init (refer also to [SWS_Dem_00340]) is called during the startup phase of the ECU, after the NVRAM Manager has finished the restore of NVRAM data. SW-Components including monitors are initialized afterwards. The function Dem_Init is also used to reinitialize the Dem module after the Dem_Shutdown was called.

Caveats of Dem_Init: The Dem module is not functional until the Dem module's environment has called the function Dem_Init.

7.2 Monitor re-initialization

The primary initialization of the monitors in the application is done via Rte_Start. The initialization of the event-specific part of the monitor can be triggered by the Dem.

[SWS_Dem_00003]

Upstream requirements: SRS_BSW_00457

⌈The Dem module shall provide the interface `InitMonitorForEvent` to trigger the initalization of a diagnostic monitor (refer also to [SWS_Dem_00573]).⌋

For details see also chapter 7.7.3.

The function parameter `InitMonitorReason` indicates the reason / trigger of initialization.

Note: The Dem module does not evaluate the return value (e.g. if other than E_OK) of this callback function.

Note: The configuration container `DemCallbackInitMForE` is used to specify the related port or c-callback per event.

[SWS_Dem_00679]

Upstream requirements: RS_Diag_04117

⌈A call to `Dem_RestartOperationCycle` shall re-initialize the diagnostic monitors of all related events by calling their respective callbacks `InitMonitorForEvent` with parameter `InitMonitorReason` set to `DEM_INIT_MONITOR_RESTART`.⌋

[SWS_Dem_00680]

Upstream requirements: RS_Diag_04117

⌈The `Dem_ClearDTC` API shall trigger the callback function `InitMonitorForEvent` of the related event(s) in case of clearing the event(s). The `InitMonitorReason` parameter shall be set to `DEM_INIT_MONITOR_CLEAR`.⌋

[SWS_Dem_00681]

Upstream requirements: RS_Diag_04125

⌈The API `Dem_SetEnableCondition` shall trigger the callback function `InitMonitorForEvent` of the related event(s) in case an enable condition of the event(s) is changed to fulfilled and thus all enable conditions of the event(s) are fulfilled. The `InitMonitorReason` parameter shall be set to `DEM_INIT_MONITOR_REENABLED`.⌋

[SWS_Dem_00682]

Upstream requirements: RS_Diag_04125, RS_Diag_04270

⌈The API `Dem_EnableDTCSetting` shall trigger the callback function `InitMonitorForEvent` of the related event(s) in case ControlDTCSetting of the event(s) is re-enabled. The `InitMonitorReason` parameter shall be set to `DEM_INIT_MONITOR_REENABLED`.⌋

[SWS_Dem_01113]

Upstream requirements: RS_Diag_04219

⌈The API Dem_SetStorageCondition shall trigger the callback function InitMonitorForEvent of the related event(s) in case a storage condition of the event(s) is changed to fulfilled and thus all storage conditions of the event(s) are fulfilled. Furthermore the callback shall only be called, if the event did report FAILED or PASSED while the storage condition was disabled. The InitMonitorReason parameter shall be set to DEM_INIT_MONITOR_STORAGE_REENABLED.⌋

The following figures show two examples:

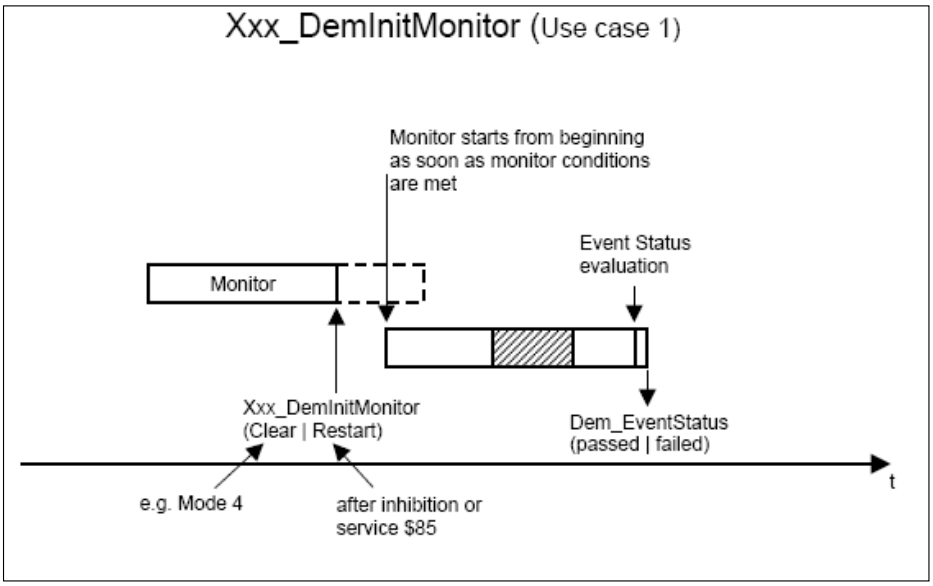


Figure 7.1: Use-case of the interface InitMonitorForEvent for a specific event

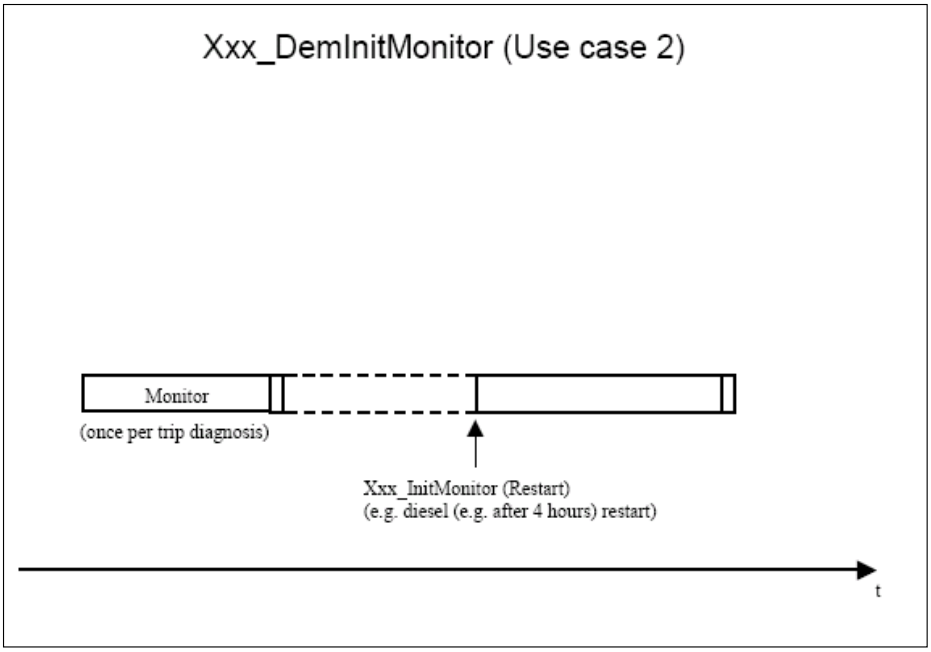


Figure 7.2: Use-case of the interface InitMonitorForEvent for a specific event

The initialization of any function (which may relate to the monitor) can also be triggered by the Dem.

Note: The Dem module does not evaluate the return value (e.g. if other than E_OK) of this callback function.

Example: Adaptations may be initialized in case of clearing the Dem module (on service 04/ISO 15031-5[12] request).

[SWS_Dem_01046]

Upstream requirements: RS_Diag_04219

⌈In case multiple triggers for `InitMonitorForEvent` occur simultaneously, the InitMforE shall only be called once. The reason shall be selected by following priority: `DEM_INIT_MONITOR_CLEAR` (highest priority), `DEM_INIT_MONITOR_RESTART`, `DEM_INIT_MONITOR_REENABLED`, `DEM_INIT_MONITOR_STORAGE_REENABLED` (lowest priority) (refer to `Dem_InitMonitorReasonType`).⌋

7.3 Diagnostic event definition

A 'Diagnostic Event' defines the atomic unit that can be handled by the Dem module. The status of a 'Diagnostic Event' represents the result of a monitor (refer to chapter 7.3.5). The Dem receives the result of a monitor from SW-C via the RTE or other BSW modules.

The Dem module uses the EventId to manage the status of the 'Diagnostic Event' of a system and performs the required actions for individual test results, e.g. stores the freeze frame.

[SWS_Dem_00153]

Upstream requirements: RS_Diag_04151

⌈The Dem module shall represent each Diagnostic Event by an EventId and the related EventName.⌋

All monitors and BSW modules use the EventId as a symbolic EventName. The Dem configuration tool replaces the symbolic names by numbers.

[SWS_Dem_00154]

Upstream requirements: RS_Diag_04151

⌈The EventId and the related EventName shall be unique per Dem module represented by the ECU configuration (refer to [SWS_Dem_00126]).⌋

The Dem is not designed to be able to handle the case where more than one monitor shares a single EventId.

The Dem module uses an internal monitor status to store the status of reported events. To the Dcm the UDS status is reported

The Dem module supports several event-specific configuration parameters as shown in the following figures. For a detailed description, refer to chapter 10 Configuration specification.

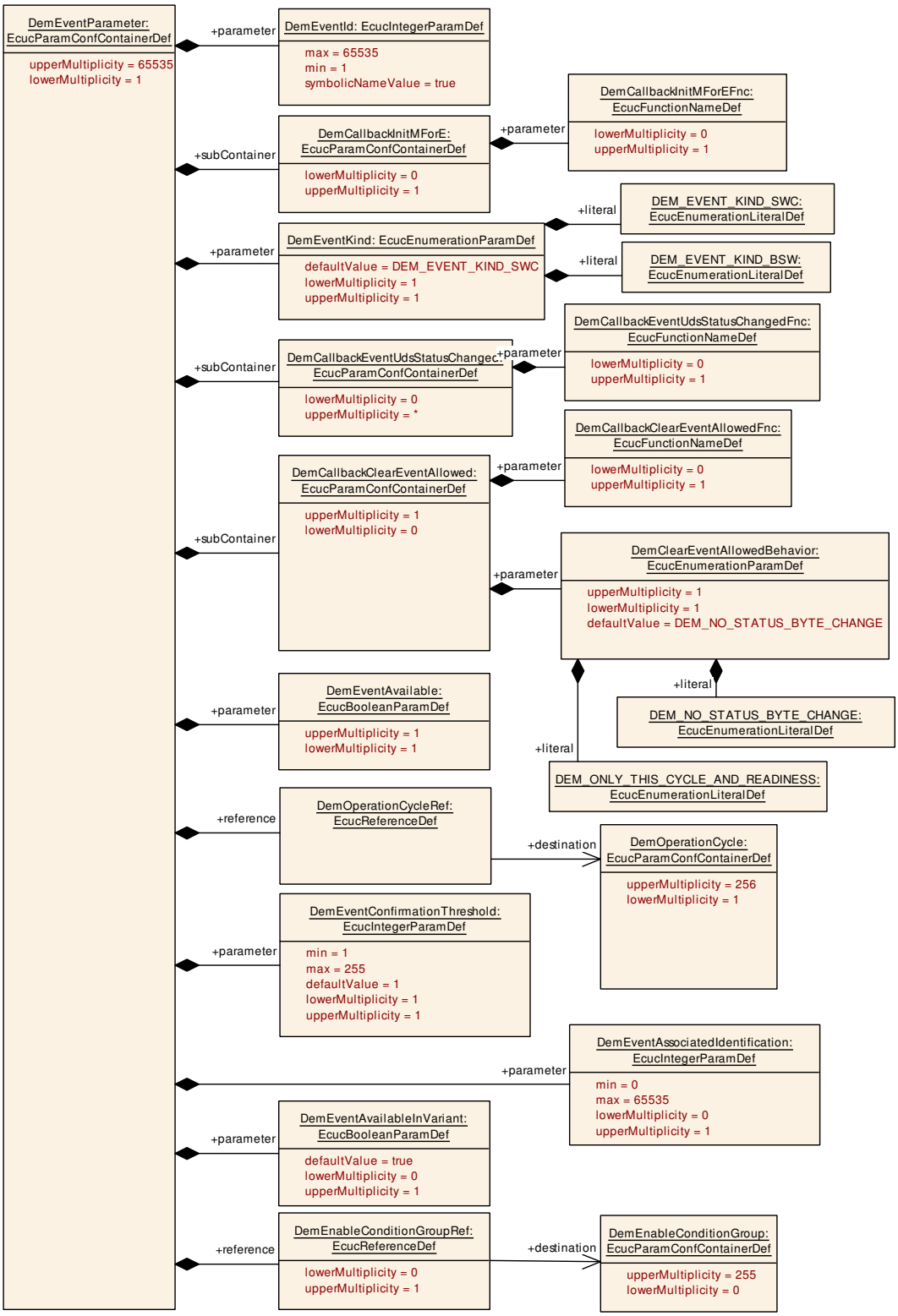


Figure 7.3: Event parameter configuration (part 1)

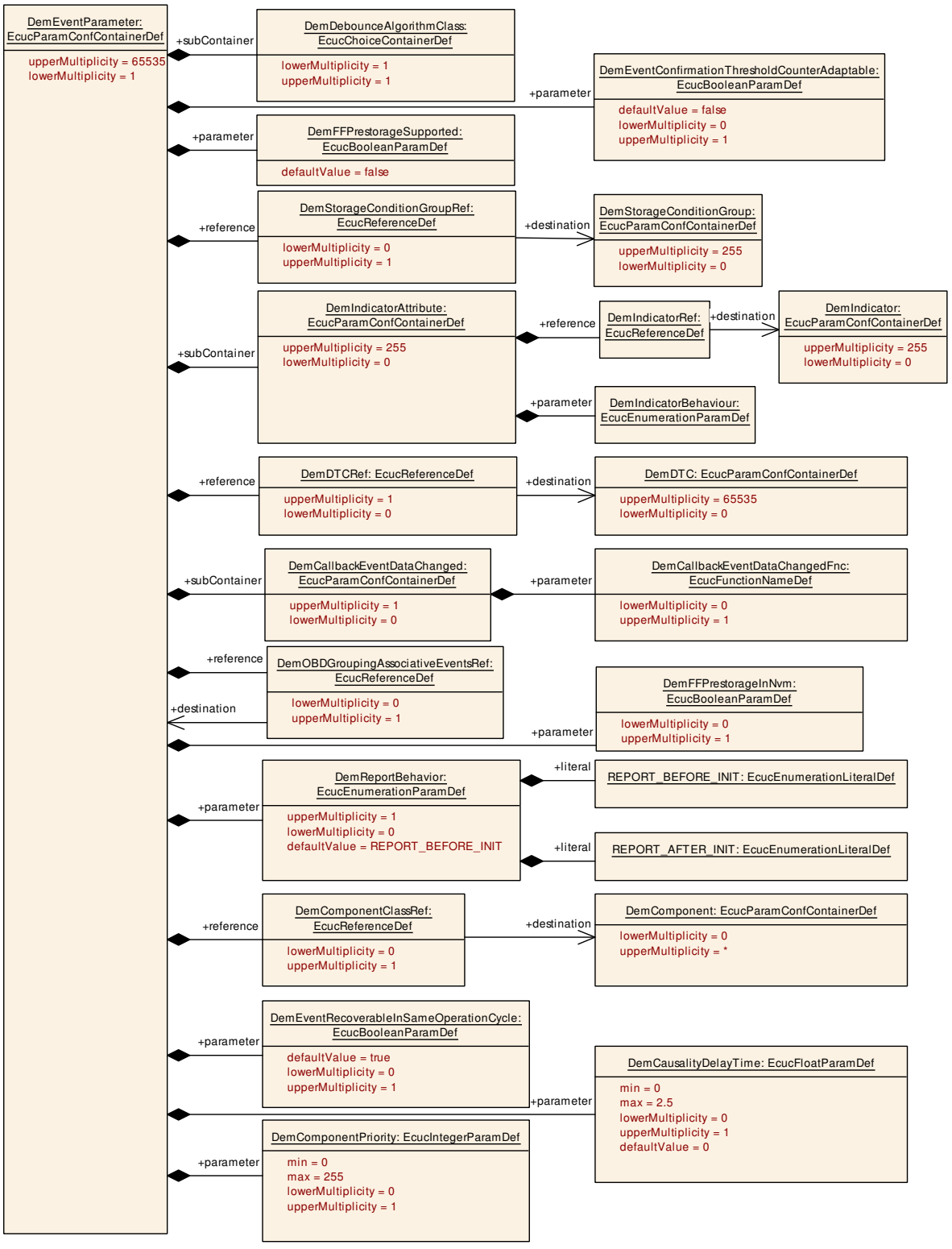


Figure 7.4: Event parameter configuration (part 2)

7.3.1 Event priority

Event priority is defined as a ranking of events based upon level of importance. It is used to determine which fault entries may be removed from the event memory in case the number of stored events exceeds the maximum number of memory entries (event memory is full).

[SWS_Dem_00382]

Upstream requirements: RS_Diag_04118, RS_Diag_04071

⌈Each supported event shall have a priority assigned to it (refer to parameter DemDTCPriority in DemDTCAttributes).⌋

[SWS_Dem_00383]

Upstream requirements: RS_Diag_04118, RS_Diag_04071

⌈A priority value of 1 shall be the highest priority. Larger priority value shall define lower importance.⌋

7.3.2 Event occurrence

[SWS_Dem_00011]

Upstream requirements: RS_Diag_04067

⌈The Dem module shall provide an occurrence counter per event memory entry.⌋

[SWS_Dem_00523]

Upstream requirements: RS_Diag_04067

⌈The Dem module shall initialize the occurrence counter with the value one if the related event is entered in the respective event memory.⌋

[SWS_Dem_00524]

Upstream requirements: RS_Diag_04067

⌈If the configuration parameter DemOccurrenceCounterProcessing is set to DEM_PROCESS_OCCCTR_TF (refer to DemPrimaryMemory or DemUserDefinedMemory), the Dem module shall increment the occurrence counter by one, triggered by each UDS status bit 0 (TestFailed) transition from 0 to 1, if the related event is already stored in the event memory.⌋

[SWS_Dem_00580]

Upstream requirements: RS_Diag_04105

⌈If the configuration parameter DemOccurrenceCounterProcessing is set to DEM_PROCESS_OCCCTR_CDTC (refer to DemPrimaryMemory or DemUserDefinedMemory), the Dem module shall increment the occurrence counter by one,

triggered by each UDS status bit 0 transition from 0 to 1, if the related event is already stored in the event memory and the UDS status bit 3 is equal to 1.⌋

[SWS_Dem_00625]

Upstream requirements: RS_Diag_04125

⌈The Dem module shall not increment the event-specific occurrence counter anymore, if it has reached its maximum value (255, refer to [SWS_Dem_00471]).⌋

7.3.3 Event kind

There are two different types of events:

- BSW-related events (reported via C-API - Dem_SetEventStatus)
- SW-C-related events (reported via RTE operation - SetEventStatus)

This kind is configurable per event (refer to DemEventKind in DemEventParameter).

This is necessary because BSW-events may be reported prior to full Dem initialization and need to be buffered (refer to chapter 7.8).

7.3.4 Event destination

The configuration parameter DemMemoryDestinationRef (refer to DemDTCAttributes) defines the dedicated storage location(s) of the event and its related data (refer to chapter 7.7.7).

The "permanent event memory" assignment is implicitly derived from the related DTC kind (refer to chapter 7.4.1). Emission-related events are automatically assigned to the permanent event memory, since the storage of an event as "permanent DTC" is dynamically derived from its current status (handling is described in chapter 7.9.5.8). In this context the term "permanent" relates to an attribute of emission-related events and does not relate only to persistent storage via NvM, which is done for each event memory type anyway.

The definition and use of the different memory types is OEM specific.

For the Dcm-Dem interface the parameter DTCOrigin is used to distinguish between the different memory areas. The intention is to allow specific operations on the different memory areas (primary, user defined and permanent).

[SWS_Dem_CONSTR_06114] Limitations on DemMemoryDestinationRef ⌈A DTC can only reference the event memories via DemMemoryDestinationRef to the event memories of the same DemEventMemorySet. The scenario that a DTC references event memories via DemMemoryDestinationRef on different DemEventMemorySet is not supported.⌋

7.3.5 Diagnostic monitor definition

A diagnostic monitor is a routine entity determining the proper functionality of a component. This monitoring function identifies a specific fault type (e.g. short to ground, open load, etc.) for a monitoring path. A monitoring path represents the physical system or a circuit, that is being monitored (e.g. sensor input). Each monitoring path is associated with exactly one diagnostic event.

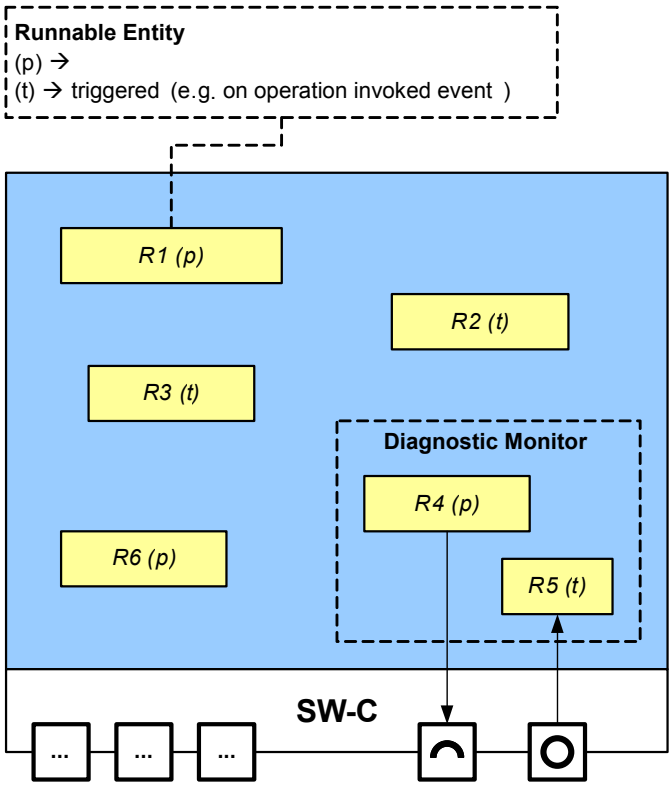


Figure 7.5: Example for a monitor embedded within a SW-C

If the monitor debounces on its own, the reporting API is called only after a qualified result (passed or failed) is available. A report is necessary at least if the result changes. However, usually it is computationally more efficient for the monitor to always call the Dem and should therefore be preferred. Hence, it is implementation specific for the Dem to deal with reports of unchanged results.

If the monitor uses the Dem-internal debouncing mechanism (refer to chapter 7.7.3), the reporting API is called whenever the code with the functional check is executed.

7.3.6 Event dependencies

The priority of events within the assigned `DemComponent` and furthermore the dependency relation between the `DemComponent`s is used for filtering the storage of error reports to the failure memory.

[SWS_Dem_01126] ⌈Whenever an event reports FAILED, it shall be considered as CONSECUTIVE FAULT, if any other event with higher priority at the same DemComponent already is FAILED.⌋

[SWS_Dem_01127] ⌈Whenever an event reports FAILED, it shall be considered as CONSECUTIVE FAULT, if any parent DemComponent is FAILED.⌋

[SWS_Dem_01128] ⌈The Dem shall allow to ignore the priority of events of a DemComponentIgnoresPriority). In this case, the event shall only be considered as CONSECUTIVE FAULT if any parent component is FAILED (other events FAILED status at the same component shall be ignored).]⌋

[SWS_Dem_01129] ⌈If a reported failure is not considered as consecutive fault, it shall be considered as CAUSAL FAULT. Causal faults shall be handled normally.⌋

[SWS_Dem_01218] ⌈If a reported failure is not considered as consecutive fault and it has configured a DemCausalityDelayTime, it shall be considered as PRELIMINARY-CAUSAL FAULT. Starting from the point in time where the report of the event occurs, until the DemCausalityDelayTime is elapsed, the event can be reconsidered as being a consecutive fault. If another failure is reported during this time which has higher priority at the same DemComponent or at any parent DemComponent, the event shall be reconsidered as consecutive fault. After the time is elapsed, the failure will not be reconsidered.⌋

[SWS_Dem_01130] ⌈A failure which is considered as consecutive fault shall not be stored to the failure memory. (The handling shall be analogous to storage condition not fulfilled).⌋

[SWS_Dem_01211] ⌈A report of DEM_EVENT_STATUS_FDC_THRESHOLD_REACHED (e.g. either via interface call or when reaching the configured threshold in debounce algorithm) which is considered as consecutive report of a Failed event shall not be stored to the failure memory. The handling shall be analogous to storage condition not fulfilled.⌋

Note: Other reports of DEM_EVENT_STATUS_FDC_THRESHOLD_REACHED will not be considered as consecutive.

[SWS_Dem_01131] ⌈The Dem shall provide the interface Dem_GetComponentFailed, which allows querying a DemComponents FAILED status.⌋

[SWS_Dem_01132] ⌈If DemTriggerFiMReports (refer to DemGeneral) is enabled, the Dem shall notify the FiM module [8] on each change of DemComponent failed status by calling the function DemTriggerOnComponentStatus.⌋

[SWS_Dem_01133] ⌈If the component status is changing and DemComponentFailedCallbackFnc is configured or DemComponentFailedCallbackUsePort is set to TRUE, the Dem shall trigger the callback DemTriggerOnComponentStatus.⌋

[SWS_Dem_CONSTR_06157] ⌈Setting DemComponentFailedCallbackUsePort to TRUE is only allowed, if DemComponentFailedCallbackFnc is not configured.⌋

7.3.7 Component availability

[SWS_Dem_01134] ⌈The Dem shall support the availability of DemComponent. Components which are not available shall be treated as if they are not included in the system (e.g. Dem_GetComponentFailed will return E_NOT_OK).⌋

[SWS_Dem_01135] ⌈The interface Dem_SetComponentAvailable shall be available to set the availability state of a component.⌋

[SWS_Dem_01136] ⌈With setting a DemComponent to not available, all assigned events shall also be set as not available.⌋

[SWS_Dem_01239] ⌈After startup all DemComponents shall be available.⌋

[SWS_Dem_01226] ⌈If a DemComponent is set to 'not available' via Dem_SetComponentAvailable, the Dem shall treat all the dependent components (children) as if they are set as 'not available'.⌋

[SWS_Dem_01227] ⌈If a DemComponent is set to 'not available' via Dem_SetComponentAvailable, the Dem shall set all assigned events to 'not available', including the events of all child nodes. The behavior for the events will be analogous to setting each event individually to 'not available' (e.g. setting the event status).⌋

[SWS_Dem_01228] ⌈If a DemComponent is set to 'not available' via Dem_SetComponentAvailable and any individual event assigned to it (see [SWS_Dem_01227]) is already failed, the individual event will be kept as 'available'. Dem_SetComponentAvailable still shall return E_OK.⌋

[SWS_Dem_01229] ⌈If a DemComponent is set to available via Dem_SetComponentAvailable, all assigned events are also set to 'available', including the events of all child nodes (if those nodes are not still set to 'not available'). The behavior for the individual events will be analogous to setting them individually to 'available' (e.g. triggering InitMonitorForEvent).⌋

[SWS_Dem_01231] ⌈If the function Dem_SetEventAvailable is called for an event to set the event to 'available', the function shall return E_NOT_OK, in case its related node is current 'not available'.⌋

[SWS_Dem_01232] ⌈If an event is set to available using Dem_SetEventAvailable and its component is 'not available', Dem_SetEventAvailable shall return E_NOT_OK.()⌋

7.4 Diagnostic trouble code definition

A ‘Diagnostic trouble code’ defines a unique identifier (shown to the diagnostic tester) mapped to a ‘Diagnostic event’ of the Dem module. The Dem provides the status of ‘Diagnostic trouble codes’ to the Dcm module (refer to chapter 7.11.2).

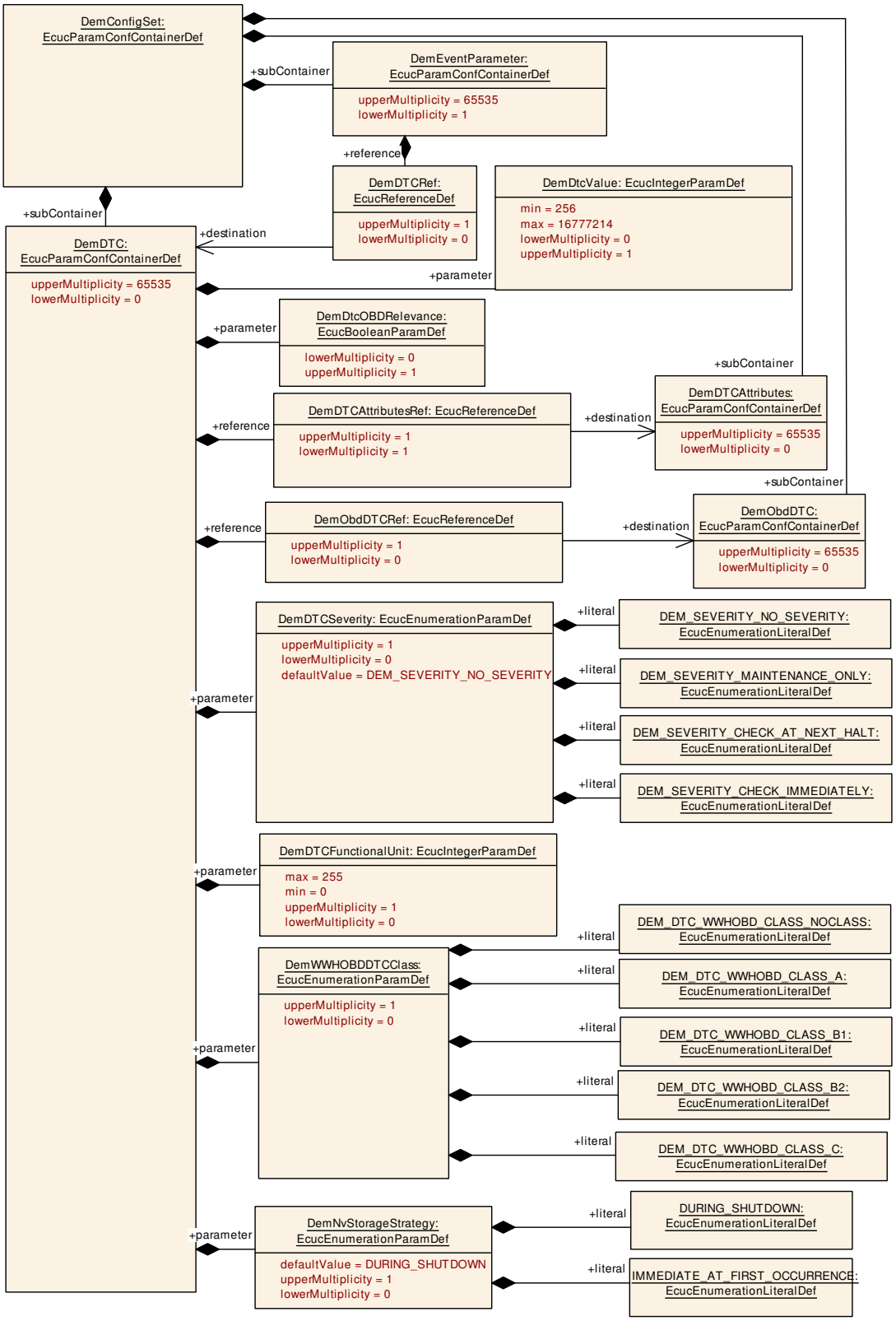


Figure 7.6: DTC configuration

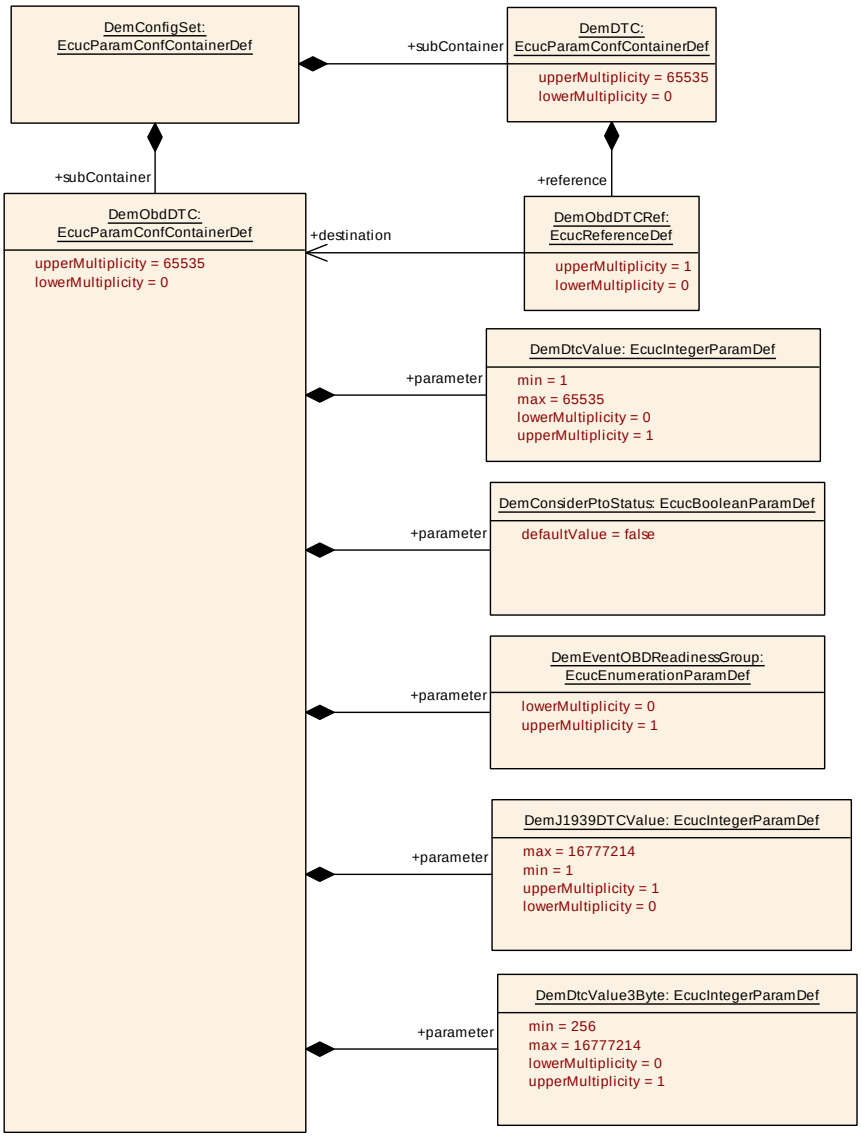


Figure 7.7: OBD-DTC configuration

7.4.1 DTC kind

No content.

7.4.2 DTC format

[SWS_Dem_00013] ⌈The Dem module shall support DTC formats for DemDTC according to:

- ISO-14229-1[2]
- SAE J2012 OBD DTC (aka 2-byte DTC) [13]

- SAE J1939-73[14]
- ISO 11992-4[15]
- SAE J2012 WWH-OBD DTC (aka 3-byte DTC) [13]

⌋

The configuration parameter DemTypeOfDTCSupported is used to specify one of the supported DTC formats defined in [SWS_Dem_00013] of the ECU (refer to Dem_GetTranslationType). This is required to determine the DTC format value to be reported for ISO-14229-1[2] service Read DTC Information (0x19).

[SWS_Dem_00645]

Upstream requirements: RS_Diag_04112

⌈The Dem module shall support different DTC numbers for UDS, OBD, and J1939 DTCs based on separate configuration parameters (refer to parameters DemDTC and DemObdDTC in DemConfigSet).⌋

For DTCs multiple combinations of the supported DTC formats (UDS, OBD, OBD3Byte and J1939) are possible. The Dem internally handles one of the two OBD format in combination with UDS and J1939 DTCs.

Some manufacturers don't want to fully change to SAE J1979-2 support with UDS and will only support the J1979-2 functionality on the limited UDS subset defined by J1979-2, wherefore the feature OBD UDS DTC separation was added.

[SWS_Dem_00277] ⌈The Dem shall report all three byte DTC values as a uint32 with byte 0 = Low-Byte, byte 1 = MiddleByte, byte 2 = HighByte and byte 3 is unused. For 2 Byte OBD DTC format there are only two bytes (HighByte, LowByte) used. The Dem services shall report these DTCs as a uint32 with byte 1 = LowByte, byte 2 = HighByte, byte 3 is unused and byte 0 = 0x00.⌋

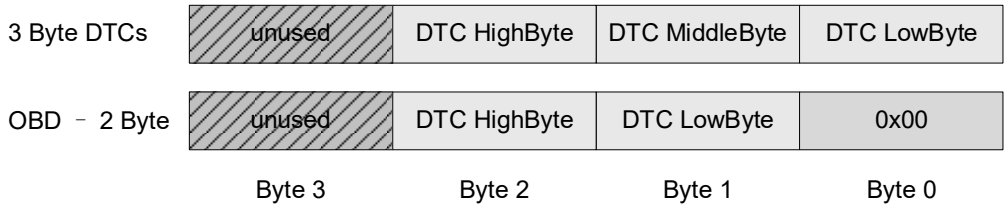


Figure 7.8: DTC Byte Order

Note: The Dem expects the SPN and FMI values of the J1939DTC to be configured according to the J1939 standard. The Dem will not do any encoding/transcoding. Refer chapter 5.7.1.11 SPN Conversion Method in [14] for details.

[SWS_Dem_01180]

Upstream requirements: RS_Diag_04113

⌈The Dem shall report DTC values as a uint32 with byte 0 = LowByte, byte 1 = MiddleByte, byte 2 = HighByte and byte 3 is unused. For WWH-OBD DTC format there are only three bytes (HighByte, MiddleByte and LowByte) used. The Dem services shall report these DTCs as a uint32 with byte 1 = LowByte, byte 2 = HighByte, byte 0 is used as FTB (as of SAE J2012-DA) [13].⌋

[SWS_Dem_CONSTR_06146] Dependency for DemDtcValue ⌈The OBD DTC DemDtcValue shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06172] Limit OBD DTCs to primary fault memory ⌈A DemDTC shall only have a configured DemObdDTCRef if the DTC has a DemMemoryDestinationRef to DemPrimaryMemory.⌋

[SWS_Dem_00269]

Upstream requirements: RS_Diag_04201

⌈The function Dem_GetDTCOfEvent shall get the DTC which is mapped to EventId by the Dem configuration.⌋

[SWS_Dem_CONSTR_06178] 3 Byte OBD DTC configuration only for OBD UDS DTC separation ⌈A 3 byte OBD DTC DemDtcValue3Byte shall be only configured if DemSupportedObdUdsDtcSeparation is set to TRUE and if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06179] 2 Byte OBD DTC configuration without DTC separation ⌈A 2 byte OBD DTC DemDtcValue shall be only configured if DemSupportedObdUdsDtcSeparation is not configured or set to FALSE.⌋

7.4.3 DTC groups

In addition to single DTC values, groups of DTCs can be configured (refer to DemGroupOfDTC), as defined by ISO 14229-1 [2] - Annex D.1. Each DTC group has its own DTC group value assigned (which must be unique to any other DTC and DTC group value). When requesting operations on DTC groups (like ClearDTC and Disable/Enable DTC), the DTCGroup is selected by the DTC value.

[SWS_Dem_01059] ⌈The DemGroupOfDTC shall represent the values of the DTC Group boundaries.⌋

Note: A group with the global boundaries 0x000100 and 0xfffe00 does not need to be configured, as the group ALL is automatically defined.

[SWS_Dem_01061] ⌈All the DTCs in the range between the dtc-code of the requested group and the next higher dtcgroup-code shall be treated as belonging to the group.⌋

Note: DTC groups are relevant for the diagnostic service ClearDiagnosticInformation (0x14, refer to Dem_ClearDTC).

[SWS_Dem_01337] ⌈The Dem shall support the predefined DTC group GroupOfAllDTCs (0xFFFFFF) within the parameter DTC of Dem_SelectDTC.⌋

[SWS_Dem_01338] ⌈The Dem shall support the predefined DTC group emission related DTCs for WWH-OBD and J1979-2 (0xFFFF33) within the parameter DTC of Dem_SelectDTC if WWH-OBD or J1979-2 is supported.⌋

Note, the DTC group 'all DTCs' will not be configured in DemGroupOfDTC, because it has always to be provided by the Dem module

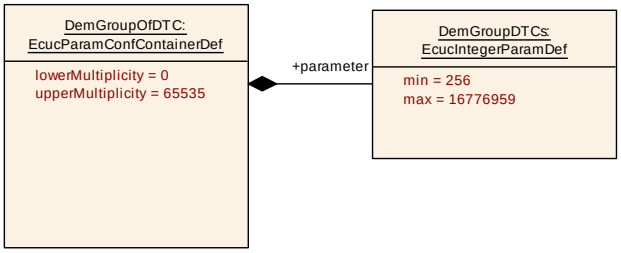


Figure 7.9: DTC group configuration

Taking into consideration that 0xFFFFFF is an unconfigurable value meaning "all DTCs group", each of the DTCs configured for a group "X" let's say, shall have their ids in the open interval (ID_of_group_X, ID_of_next_configured_group) e.g.:

powertrain DTC group –> 0x3FFF00	DTC1 –> 0x3FFF04 DTC2 –> 0x3FFF06 . . . DTCn –> 0x3FFF0E
body DTC group –> 0x7FFF0F	... DTCs in the body DTC group ...

As a configuration constraint of the above mentioned approach on DTC groups, if there is one or more DTC groups configured, there is no way to configure group independent DTCs.

7.4.4 DTC severity

The DTC severity is used to provide information regarding the importance of the specific events according to ISO-14229-1[2], Annex D.3 "DTC severity and class definition". The DemDTCSeverity is only available for UDS DTCs.

[SWS_Dem_01291]

Upstream requirements: RS_Diag_04071

⌈If the API Dem_GetSeverityOfDTC or Dem_GetNextFilteredDTCAndSeverity is called and for the selected DTC a severity value in DemDTCSeverity is configured, the Dem shall set the value configured in DemDTCSeverity as DTCSeverity.⌋

[SWS_Dem_01292]

Upstream requirements: RS_Diag_04071

⌈If the API Dem_GetSeverityOfDTC or Dem_GetNextFilteredDTCAndSeverity is called and for the selected DTC no severity value in DemDTCSeverity is configured, the Dem shall set DEM_SEVERITY_NO_SEVERITY as DTCSeverity.⌋

7.4.5 Functional unit

[SWS_Dem_01293]

Upstream requirements: RS_Diag_04071

⌈If the API Dem_GetFunctionalUnitOfDTC is called and for the selected DTC a functional unit value in DemDTCFunctionalUnit is configured, the Dem shall set the value configured in DemDTCFunctionalUnit in the out parameter DTCFunctionalUnit.⌋

[SWS_Dem_01294]

Upstream requirements: RS_Diag_04071

⌈If the API Dem_GetFunctionalUnitOfDTC is called and for the selected DTC no functional unit in DemDTCFunctionalUnit is configured, the Dem shall set a value of 0x00 in the out parameter DTCFunctionalUnit.⌋

7.4.6 DTC significance

There are two different significance levels of DTCs:

- fault: classifies a failure, which relates to the component/ECU itself (and requires for example a repair action)
- occurrence: classifies an issue, which indicates additional information concerning insufficient system behavior (and relates for example to a condition out of the ECU's control)

This significance level is configurable per event (refer to DemDTCSignificance in DemDTCAttributes) and can be mapped as a data element (refer to DEM_SIGNIFICANCE).

7.4.7 Suppress DTC output

This chapter describes the dynamic suppression of events/ DTCs during runtime via API call. A suppressed DTC behaves in a way that it is not visible for the tester but can be processed continuously by the diagnostic monitor.

An exemplary use-case could be that the decision to hide a DTC in a certain market is done by the SW-C at runtime.

The configuration parameter DemSuppressionSupport (refer to DemGeneral) controls the availability of the DTC suppression functionality

[SWS_Dem_00915]

Upstream requirements: RS_Diag_04220

⌈The Dem shall suppress a DTC if all related events of this DTC are not available.⌋

For details see chapter 7.4.8.

If there is a one to one relationship between DTC and event, the DTC is suppressed if the related event is not available. If the DTC is a combined DTC, the DTC is suppressed if all combined events are not available.

Concerning output and query functions, suppressed DTCs are treated as if they are not configured in the system.

[SWS_Dem_01100] Behavior of DTC query functions for suppressed DTCs

Upstream requirements: RS_Diag_04220

⌈In case any of the following APIs is called

- Dem_GetStatusOfDTC
- Dem_GetSeverityOfDTC
- Dem_GetFunctionalUnitOfDTC
- Dem_ClearDTC

on a single selected DTC which is suppressed, the Dem shall return DEM_WRONG_DTC.⌋

[SWS_Dem_01101]

Upstream requirements: RS_Diag_04220

⌈A suppressed DTC shall not be visible for the following Dcm query-functions; therefore the following functions shall treat the DTC as if filter is not matching:

- Dem_DcmGetDTCOfOBDFreezeFrame
- Dem_DcmReadDataOfOBDFreezeFrame
- Dem_DcmGetInfoTypeValue08

- Dem_DcmGetInfoTypeValue0B
- Dem_DcmReadDataOfPID01
- Dem_GetDTCByOccurrenceTime
- Dem_GetNextFilteredDTC
- Dem_GetNextFilteredDTCAndFDC
- Dem_GetNextFilteredDTCAndSeverity
- Dem_GetNextFilteredRecord
- Dem_J1939DcmFirstDTCwithLampStatus
- Dem_J1939DcmGetNextDTCwithLampStatus
- Dem_J1939DcmGetNextFreezeFrame
- Dem_J1939DcmSetDTCFilter
- Dem_J1939DcmGetNumberOfFilteredDTC
- Dem_J1939DcmGetNextFilteredDTC
- Dem_J1939DcmGetNextFilteredDTCWithSeverity
- Dem_J1939DcmSetRatioFilter
- Dem_J1939DcmGetNextFilteredRatio
- Dem_J1939DcmReadDiagnosticReadiness1
- Dem_J1939DcmReadDiagnosticReadiness2
- Dem_J1939DcmReadDiagnosticReadiness3
- Dem_SelectFreezeFrameData
- Dem_SetDTCFilter
- Dem_SelectExtendedDataRecord
- Dem_SetDTCFilterByExtendedDataRecordNumber
- Dem_SetDTCFilterByReadinessGroup

⌋

Note: This means, the (external) reporting (e.g. DTC number, UDS status, DTC extended data records, DTC statistical data, IUMPR, ...) of the DTC is suppressed.

[SWS_Dem_01102]

Upstream requirements: RS_Diag_04220

⌈DTC suppression shall not stop event processing of the corresponding DTC.⌋

Note: This means that e.g. event qualification, update of environmental data and functional degradation is still processed within the Dem.

Note: If DTC/freeze frame should not be stored in the fault memory storage conditions should be used in addition.

[SWS_Dem_01307] Functionality of Dem_GetDTCSuppression

Upstream requirements: RS_Diag_04220

⌈If Dem_GetDTCSuppression is called, the Dem shall return the DTC suppression status in the output parameter SuppressionStatus. A value of TRUE means that the DTC was suppressed by a call of Dem_SetDTCSuppression with SuppressionStatus = TRUE, a value of FALSE means, that the DTC is currently not suppressed.⌋

[SWS_Dem_01315]

Upstream requirements: RS_Diag_04220

⌈If Dem_SetDTCSuppression is called the Dem shall set the suppression status of the selected DTC to value of the parameter SuppressionStatus.⌋

Also there are various reasons, why a DTC is not processed and thus is not visible for diagnostics, the suppression state itself is only changed by Dem_SetDTCSuppression. Dem_GetDTCSuppression only evaluates this suppression state.

7.4.8 Availability of events (visibility and computation)

This chapter describes a method to mark events as not available without removing them from the actual configuration. An event which is not available is treated as if it was not included in the Dem configuration.

An exemplary use-case could be that a control unit and its software support different kind of hardware to be controlled. Hardware variants vary in certain parts being available or not available. Still the same control unit and software shall be applied.

1. One calibration set is generated per hardware variant. The application engineers set up a post-build configuration to make sure the software behaves correctly for the hardware. For certain parts not included in a hardware variant, this means that related Events need to be shut down, suppressed. These events shall behave completely neutral, never set their LastFailed, not cause limp-home modes, not be visible to garage services.

   => ECUC parameter DemEventAvailableInVariant set to TRUE (enabled) or FALSE (disabled)

2. One calibration covers multiple hardware variants. During production, or even at run-time when hardware is changed (by the garage) some coding bits are written to the control unit, or some network message is received, to change the

behaviour of the software in accordance with the changes in hardware. As in 1, certain Events need to be suppressed.

=> ECUC parameter DemEventAvailableInVariant set to TRUE (enabled)

=> ECUC parameter DemEventAvailable defines the initial availability of an event

=> Interface Dem_SetEventAvailable is used to change the availability defined by DemEventAvailable during runtime

[SWS_Dem_01343] Enable/disable Events by configuration ⌈The configuration parameter DemEventAvailableInVariant shall be used as master to consider an Event as enabled or disabled. If the configuration parameter DemEventAvailableInVariant is set to FALSE, the Dem shall treat the Event as if it is not configured as DemEventParameter. If DemEventAvailableInVariant is not set or set to TRUE, the Dem shall consider the Event as enabled depending on other sources.⌋

Note: DemEventAvailableInVariant allows to disable an Event that is not needed, but included in the software release; even the related SW-C is prevented from making it available through Dem_SetEventAvailable.

[SWS_Dem_01344] Dealing with availability of Events ⌈The Dem shall treat an event as unavailable if:

- DemEventAvailableInVariant == FALSE or
- DemEventAvailable == FALSE or
- DemEventAvailable is overwritten by Dem_SetEventAvailable(FALSE) or
- The component referenced by DemComponentClassRef is not available

⌋

[SWS_Dem_00687]

Upstream requirements: RS_Diag_04126, RS_Diag_04274

⌈The Dem shall provide functionality during runtime to treat events as if they were not configured as DemEventParameter. The configuration parameter DemAvailabilitySupport is used to enable the functionality provided by Dem_SetEventAvailable().⌋

[SWS_Dem_01103]

Upstream requirements: RS_Diag_04126

⌈Unavailable events shall not be considered for computation of service $01 PID $41.⌋

[SWS_Dem_01104]

Upstream requirements: RS_Diag_04126

⌈IUMPR ratios referring to an unavailable event shall neither be computed nor reported.⌋

[SWS_Dem_01106]

Upstream requirements: RS_Diag_04126

⌈The Dem shall not provide the function Dem_SetEventAvailable in VARIANT-POST-BUILD. Rational: In VARIANT-POST-BUILD the initial values are not valid before Dem_Init (postbuild initialization time).⌋

[SWS_Dem_01108]

Upstream requirements: RS_Diag_04126

⌈If an event is set to unavailable, the corresponding event shall be treated as if it is not configured in the system (e.g. Dem_SetEventStatus and Dem_GetEventUdsStatus shall return E_NOT_OK). The following APIs are affected:

- Dem_SetEventStatus
- Dem_GetEventUdsStatus
- Dem_ResetEventDebounceStatus
- Dem_ResetEventStatus
- Dem_PrestoreFreezeFrame
- Dem_ClearPrestoredFreezeFrame
- Dem_GetDebouncingOfEvent
- Dem_GetDTCOfEvent
- Dem_GetFaultDetectionCounter
- Dem_GetEventFreezeFrameDataEx
- Dem_GetEventExtendedDataRecordEx
- Dem_ClearDTC
- Dem_DcmGetDTRData
- Dem_RepIUMPRFaultDetect
- Dem_RepIUMPRDenRelease
- Dem_SetWIRStatus
- Dem_SetEventStatusWithMonitorData
- Dem_GetMonitorStatus

- Dem_SetEventConfirmationThresholdCounter
- Dem_SetEventDisabled
- Dem_ResetMonitorStatus

⌋

[SWS_Dem_CONSTR_06186] Matching event availability parameters in post-build configuration ⌈When post-build configuration is enabled in the Dem, the configuration parameters DemEventAvailable and DemEventAvailableInVariant shall have the same value for each BSW event.⌋

Note: The availability of an event may also impact the suppression of a DTC (see [SWS_Dem_00915]).

[SWS_Dem_01109]

Upstream requirements: RS_Diag_04126

⌈When the API Dem_SetEventAvailable is called with AvailableStatus = FALSE, the Dem shall return E_NOT_OK if:

- for the event an event memory entry already exists, or
- any of the event status flags 'testFailed', 'pending', 'confirmed' or 'warningIndicatorRequested' is set

⌋

[SWS_Dem_01302] Asynchronous behavior of Dem_SetEventAvailable

Upstream requirements: RS_Diag_04126

⌈The Dem shall process a call of the Dem_SetEventAvailable asynchronously. This means that the final result is available at a later point in time.⌋

A later point in time is meant to be implementation specific, it could be the next main function and after Dem_SetEventAvailable has returned.

[SWS_Dem_01110]

Upstream requirements: RS_Diag_04126

⌈If Dem_SetEventAvailable is called with AvailableStatus == 'false', the Dem shall set the UDS status shall be set to 0x00 for this event.⌋

Note: If the UDS status changes, the corresponding callbacks are called to properly provide the information about a changed status to the users.

[SWS_Dem_01111]

Upstream requirements: RS_Diag_04126

⌈If Dem_SetEventAvailable is called with AvailableStatus == 'true', the Dem shall set the UDS status shall be set to 0x50 for this event.⌋

[SWS_Dem_01345] Reaction on calls related to disabled events ⌈If an Event is configured with DemEventAvailableInVariant set to FALSE and the API Dem_SetEventAvailable is called with AvailableStatus = TRUE for this EventId, the API Dem_SetEventAvailable shall return E_NOT_OK.⌋

[SWS_Dem_01230] ⌈If the function Dem_SetEventAvailable is called to set an event to 'available', the Dem shall trigger the event initialization using DEM_INIT_MONITOR_REENABLED as InitMonitorReason.⌋

[SWS_Dem_01346] ⌈If Dem_GetEventAvailable is called with a valid EventId, it shall set the output value AvailableStatus to the Event's availability and return E_OK.

The availability is determined by the configuration parameter DemEventAvailable (modifiable through Dem_SetEventAvailable) and DemEventAvailableCnf ([SWS_Dem_01344] ("Dem shall treat an Event as unavailable...").⌋

7.5 Monitored Component Definition

A monitored component is a part of the system, which is relevant for being checked for proper operation. Malfunctions of the component need to be reported and documented. For this, usually one or multiple monitorings are defined and assigned to the component.

Furthermore the status of such components is relevant and used for influencing the system behavior (status of components may be used in Fim) and error documentation (filtering of consecutive failure entries).

The Dem allows to explicitly define the monitored components as DemComponent, defining the relation between events and components and finally allows to define the relation in-between the components.

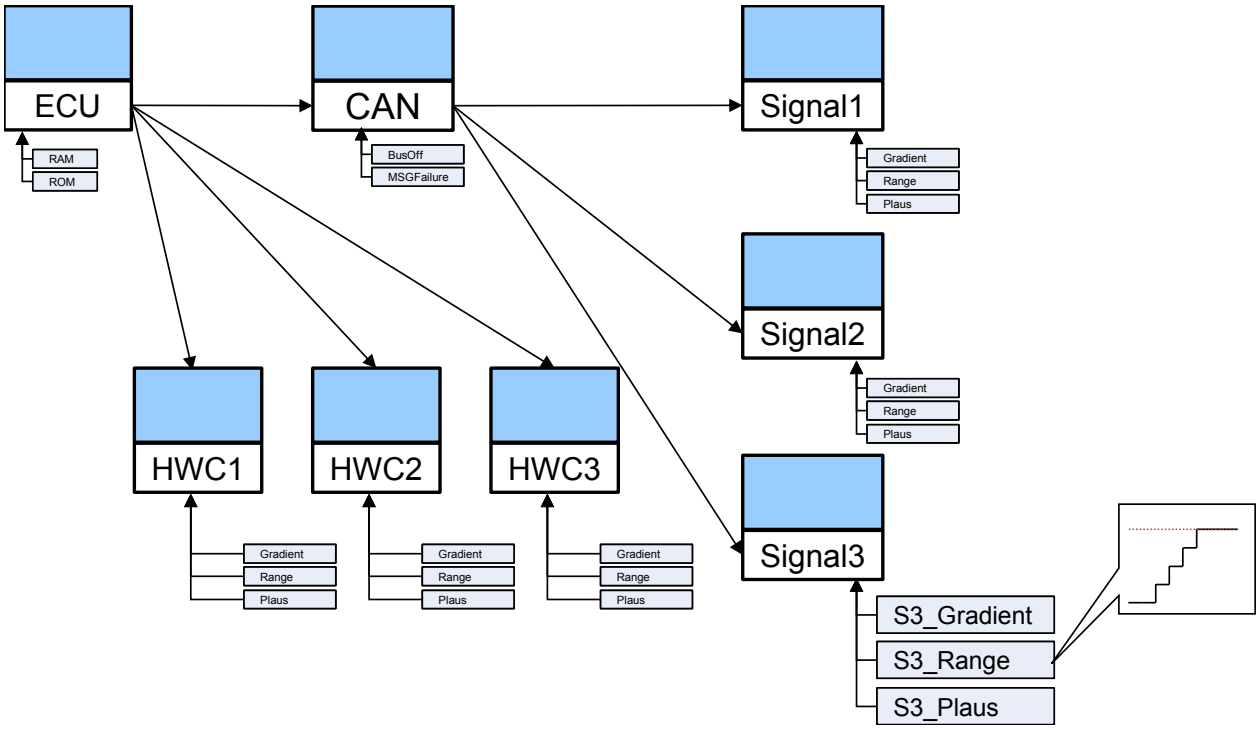


Figure 7.10: Event dependencies

7.5.1 Definition of components and dependencies

[SWS_Dem_01118] ⌈The Dem shall represent monitored components as DemComponent with a unique identifier, which is used in interfaces to identify the component.⌋

[SWS_Dem_01119] ⌈Each DemComponent shall have a boolean status FAILED, which represents the information whether the component has errors or not.⌋

[SWS_Dem_01120] ⌈The Dem shall allow defining relations between DemComponent. The relations are directed, which means one component is depending on another component (DemImmediateChildComponentRef). Any component which has a directed relation (also over multiple components) is considered as parent component.⌋

Example: a -> b -> c -> d
a and b are parents of c
c and d are depending on b

[SWS_Dem_CONSTR_06106] ⌈Only directed acyclic graph structures are supported for the dependencies of DemComponent.⌋

[SWS_Dem_01121] ⌈The Dem shall allow to assign events to DemComponents.⌋

[SWS_Dem_CONSTR_06107] ⌈Events may be assigned to exactly one DemComponent for which the monitoring is testing the error conditions. Multiple events may be assigned to the same component.⌋

[SWS_Dem_01122] ⌈The Dem shall allow sorting the events assigned to a DemComponent by priority. The priority is defined per event as DemComponentPriority.⌋

[SWS_Dem_01123] ⌈On reporting an error (DEM_MONITOR_STATUS_TF) for an event, the assigned DemComponent shall be set as FAILED in the context of the event report.⌋

[SWS_Dem_01124] ⌈On setting a DemComponent to FAILED, all dependent components shall be set as FAILED.⌋

[SWS_Dem_01125] ⌈A DemComponent shall be set to NOT FAILED, when allassigned events are NOT FAILED (DEM_MONITOR_STATUS_TF) and all parent components are NOT FAILED.⌋

7.6 Operation cycle management

The Dem module provides the possibility to define different operation cycles (ref. to ISO-14229-1[2]).

Examples of operation cycles are:

- ignition on/off cycle
- power up/power down cycle
- OBD driving cycle
- engine warm up cycle
- operation active/passive cycle
- accumulated operating time

The various operation cycles serve as the basis for calculating cycle-dependent status information such as testCompleteThisOperationCycle. In addition, the operation cycles are the basis for certain functionalities such as the confirmation of failures and the aging of failures. The restart or switch to the next operation cycle via Dem_RestartOperationCycle serves as the trigger for the calculation of these functionalities.

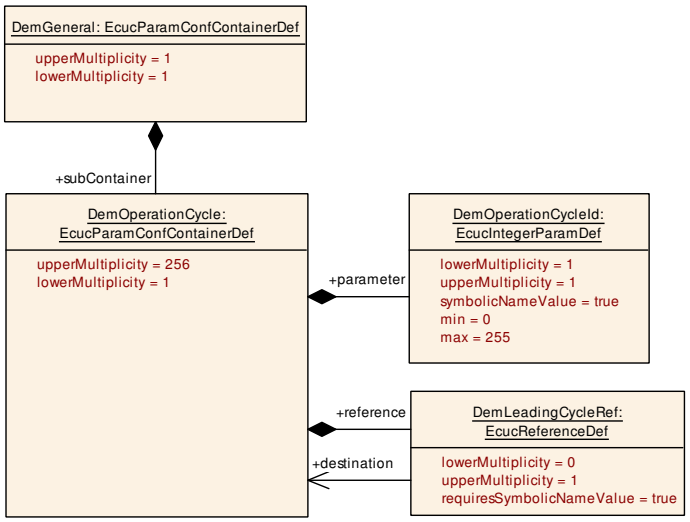


Figure 7.11: Operation cycle configuration

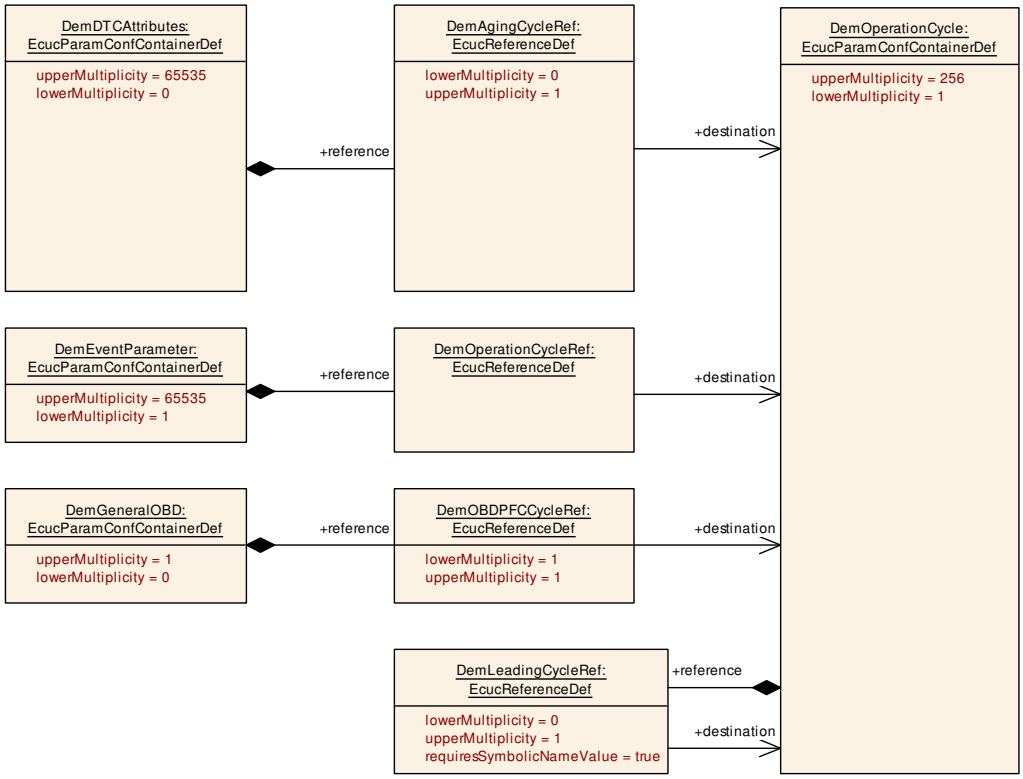


Figure 7.12: Operation and aging cycle assignment

[SWS_Dem_01318] Operation Cycle restart

Upstream requirements: RS_Diag_04178

⌈Calling the function `Dem_RestartOperationCycle` shall be allowed only after `Dem_PreInit`. In case an `operation cycle` is restarted before `Dem_Init`, the processing of the restart shall be done after `Dem_Init`.⌋

[SWS_Dem_01316] Operation Cycle restart

Upstream requirements: RS_Diag_04178

⌈The Dem shall only restart operation cycles after the previous restart has been fully processed. In other words this means that there is no Operation Cycle restart queue.⌋

[SWS_Dem_01221]

Upstream requirements: RS_Diag_04129, RS_Diag_04133

⌈The Dem allows definition of dependent operation cycles using DemLeadingCycleRef parameter. Whenever the parameter is set, the operation cycle is considered as a dependent operation cycle. Whenever the parameter is not set, the operation cycle is considered as "normal" operation cycle.⌋

[SWS_Dem_01222]

Upstream requirements: RS_Diag_04129

⌈If the operation cycle passed to Dem_SetCycleQualified is not configured at all, the Det error DEM_E_WRONG_CONFIGURATION shall be reported.⌋

[SWS_Dem_01317]

Upstream requirements: RS_Diag_04178

⌈The Dem shall store the qualification state of the dependent operation cycles over power cycles.⌋

[SWS_Dem_01224] Operation Cycle restart

Upstream requirements: RS_Diag_04178

⌈Whenever an operation cycle is requested to be restarted via Dem_RestartOperationCycle, all operation cycles which are depending on the selected operation cycle shall also be restarted, but only if they are qualified⌋

Note: Setting a dependent operation cycle as qualified will not start the operation cycle, even if the leading operation cycle has been started before.

[SWS_Dem_01225] Operation Cycle restart

Upstream requirements: RS_Diag_04178

⌈When restarting a dependent operation cycle, its operation cycle qualification shall be reset.⌋

[SWS_Dem_01234]

Upstream requirements: RS_Diag_04129

⌈The API Dem_GetCycleQualified shall provide the qualification state of the requested operation cycle in the out parameter "isQualified".⌋

[SWS_Dem_01235]

Upstream requirements: RS_Diag_04129

⌈If the operation cycle passed to `Dem_GetCycleQualified` is not configured at all, the `Det` error `DEM_E_WRONG_CONFIGURATION` shall be reported.⌋

[SWS_Dem_01220]

Upstream requirements: RS_Diag_04129

⌈If `DemOBDDelayedDCYConfirmedAndMIL` is set to TRUE, the behavior of events linked to the OBD driving cycle (`DemOBDDrivingCycleRef`) shall be changed:
If the operation cycle is not qualified, status reports shall be processed, except update of the confirmed status. At the moment the OBD driving cycle is set to qualified, the `Dem` shall execute computations required to reach the confirmed states of events collected during the "not qualified" phase of the cycle.⌋

OBD legislation requires specific implementation of operation cycles. For emission related ECUs it is mandatory to implement these accordingly.

7.6.1 Operation Cycle Counters

[SWS_Dem_01358] Initialization of Operation Cycle Counters ⌈If a new `event memory` entry is created for an event then the operation cycle counters 'Cycles since last failed', 'Cycles since first failed' and 'Failed cycles' shall be initialized to zero.⌋

[SWS_Dem_01359] Size of Operation Cycle Counters ⌈Each operation cycle counter defined in [SWS_Dem_01358] shall have the size of one byte.⌋

[SWS_Dem_01360] Overflow handling of Operation Cycle Counters ⌈If any count operation for an operation cycle counter occurs which would cause it to roll over past 0xFF then it shall instead be maintained at 0xFF.⌋

7.6.1.1 Cycles since last failed

The counter Cycles since last failed is representing the number of operation cycles since the DTC fault detection counter last reached its maximum value +127 (since DTC information was last cleared). All operation cycles, including those during which the test was not completed shall be included.

[SWS_Dem_00984]

Upstream requirements: RS_Diag_04190

⌈If the counter 'Cycles since last failed' is mapped to an extended data record (`DemInternalDataElement` set to `DEM_CYCLES_SINCE_LAST_FAILED`), it shall be available per 'event related data' record.⌋

[SWS_Dem_01402]

Upstream requirements: RS_Diag_04190

⌈If the counter 'Cycles since last failed excluding test not complete' is mapped to an extended data record (DemInternalDataElement set to DEM_CYCLES_TESTED_SINCE_LAST_FAILED), it shall be available per 'event related data' record.⌋

[SWS_Dem_00772]

Upstream requirements: RS_Diag_04068

⌈If the counter 'Cycles since last failed' is already stored in event memory, it shall be reset to zero when the internal debounce counter reaches DemDebounceCounterFailedThreshold.⌋

[SWS_Dem_00773]

Upstream requirements: RS_Diag_04178

⌈If the counter 'Cycles since last failed' is already stored in event memory, it shall be incremented at the end of the referenced operation cycle (refer to DemOperationCycleRef).⌋

7.6.1.2 Cycles since first failed

The counter Cycles since first failed is representing the number of operation cycles since the DTC fault detection counter first reached its maximum value of +127 (since DTC information was last cleared). All operation cycles, including those in which the test has not been completed shall be included.

[SWS_Dem_00775]

Upstream requirements: RS_Diag_04190

⌈If the counter 'Cycles since first failed' is mapped to an extended data record (DemInternalDataElement set to DEM_CYCLES_SINCE_FIRST_FAILED), it shall be available per 'event related data' record.⌋

[SWS_Dem_01403]

Upstream requirements: RS_Diag_04190

⌈If the counter 'Cycles since first failed excluding test not complete' is mapped to an extended data record (DemInternalDataElement set to DEM_CYCLES_TESTED_SINCE_FIRST_FAILED), it shall be available per 'event related data' record.⌋

[SWS_Dem_01404] Latching of Internal Data Element DEM_CYCLES_TESTED_SINCE_LAST_FAILED

Upstream requirements: RS_Diag_04190

⌈The Dem shall latch internal data element DEM_CYCLES_TESTED_SINCE_LAST_FAILED if it has reached 255 and no longer increment it.⌋

[SWS_Dem_01405] Latching of Internal Data Element DEM_CYCLES_TESTED_SINCE_FIRST_FAILED

Upstream requirements: RS_Diag_04190

⌈The Dem shall latch internal data element DEM_CYCLES_TESTED_SINCE_FIRST_FAILED if it has reached 255 and no longer increment it.⌋

[SWS_Dem_01406] Availability of internal data element Internal Data Element DEM_CYCLES_TESTED_SINCE_LAST_FAILED

Upstream requirements: RS_Diag_04190

⌈The Dem shall support the internal data element DEM_CYCLES_TESTED_SINCE_LAST_FAILED only for stored events.⌋

[SWS_Dem_01407] Availability of internal data element Internal Data Element DEM_CYCLES_TESTED_SINCE_FIRST_FAILED

Upstream requirements: RS_Diag_04190

⌈The Dem shall support the internal data element DEM_CYCLES_TESTED_SINCE_FIRST_FAILED only for stored events.⌋

[SWS_Dem_00777] ⌈If the counter 'Cycles since first failed' is already stored in event memory, it shall be incremented at the end of the referenced operation cycle (refer to DemOperationCycleRef).⌋

7.6.1.3 Failed cycles

The counter Failed cycles is representing the number of operation cycles during which the DTC fault detection counter reached its maximum value of +127 (since DTC information was last cleared)

[SWS_Dem_00779]

Upstream requirements: RS_Diag_04068, RS_Diag_04189, RS_Diag_04190

⌈If the counter 'Failed cycles' is mapped to an extended data record (DemInternalDataElement set to DEM_FAILED_CYCLES), it shall be available per 'event related data' record.⌋

[SWS_Dem_00781]

Upstream requirements: RS_Diag_04105

⌈If the counter 'Failed cycles' is already stored in event memory, it shall be incremented at the end of the referenced operation cycle (refer to DemOperationCycleRef) in case the UDS status bit 1 is set to 1.⌋

7.7 Event memory description

The 'event memory' is defined as a set of event records located in a dedicated memory block. The event record includes at least the UDS status and the event related data.

[SWS_Dem_00010] ⌈The Dem module shall support the primary event memory.⌋

[SWS_Dem_00548] ⌈If configured (refer to [SWS_Dem_00162]) the Dem module shall support the additional event memories (user defined, permanent).⌋

[SWS_Dem_01247]

Upstream requirements: RS_Diag_04164

⌈Configured via the container DemEventMemorySet, the Dem supports multiple independent event memories. Each API call with DemClient as parameter, shall be applicable only on the configured event memory (primary or user defined) for this client.⌋

[SWS_Dem_01217]

Upstream requirements: RS_Diag_04214

⌈In order to calculate the user defined memory ID from Dem_DTCOriginType, the Dem shall substract a value of 0x0100 from the obtained value if the value is in the range of [0x0100, 0x01FF].⌋

The size of the different event memories is configurable in the Dem configuration.

[SWS_Dem_00162] ⌈The Dem shall provide configuration parameters to adapt the fault memory size to the ECU memory space available (refer to configuration parameters DemMaxNumberEventEntryPrimary or DemMaxNumberEventEntryUserDefined).⌋

[SWS_Dem_CONSTR_06118] Unique DTC values within a single event memory ⌈The DemDtcValue shall be unique within all DTCs referencing the same event memory.⌋

[SWS_Dem_CONSTR_06170] Unique extended data record values within a single event memory ⌈All the DemExtendedDataRecordClass containers referenced from DTCs of one event memory shall have a distinct DemExtendedDataRecordNumber.⌋

[SWS_Dem_CONSTR_06171] Unique snapshot record values within a single event memory ⌈All the DemFreezeFrameRecordClass containers referenced from DTCs of one event memory shall have a distinct DemFreezeFrameRecordNumber.⌋

[SWS_Dem_00329]

Upstream requirements: RS_Diag_04077

⌈For storing to non-volatile memory the Dem shall use the NVRAM Manager.⌋

For details see chapter 7.11.5.

The following figure shows an example of a logical Dem `event memory` layout.

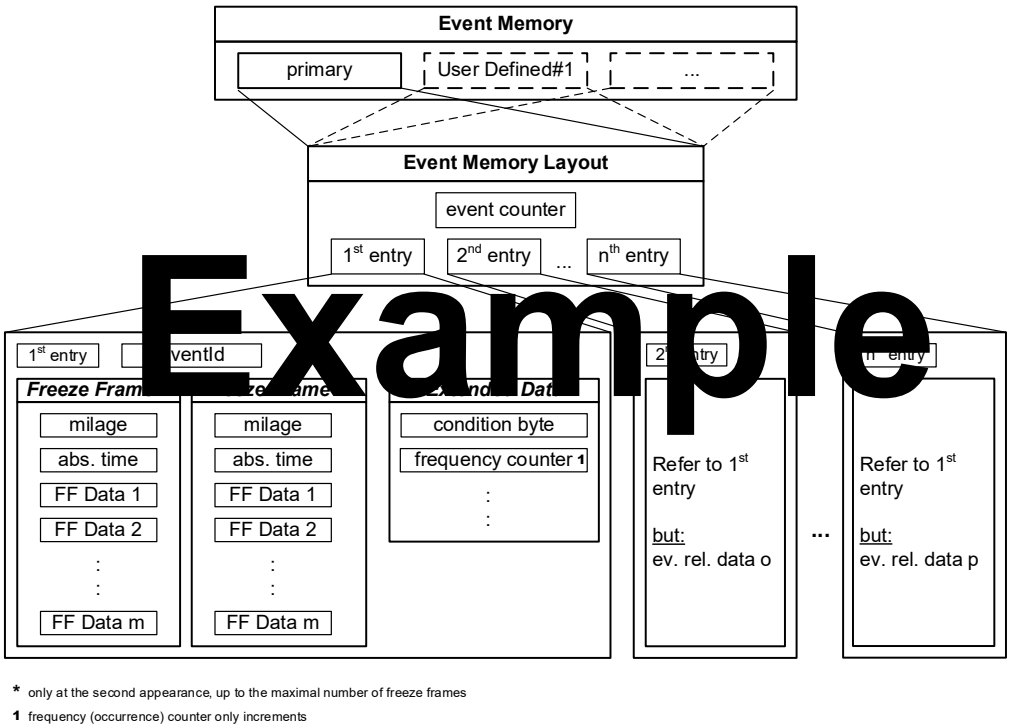


Figure 7.13: Example of a logical Dem `event memory` layout

If there are limitations of the memory size, it is necessary to provide overflow indication of the `event memory` and a displacement strategy (refer to chapter 7.7.2).

7.7.1 Event status management

The 'Event Status Management' is the Dem's ability to record and retain events, event status and associated data.

[SWS_Dem_00330]

Upstream requirements: RS_Diag_04067

⌈The Dem module shall provide the capability to report the status of an event allowing a diagnostic monitor to inform the Dem about the result of the internal diagnostic test (refer to `Dem_SetEventStatus`).⌋

[SWS_Dem_01333] ⌈The API Dem_SetEventStatusWithMonitorData shall have the same functionality as Dem_SetEventStatus with the only difference that it provides further parameters with data to store in an event related data.⌋

The monitors, which are located in the application, should call the function (Dem_SetEventStatus) to report an event status as soon as a new test result is available. This will be done independently of the current state of the Dem module (refer to Figure 7.14).

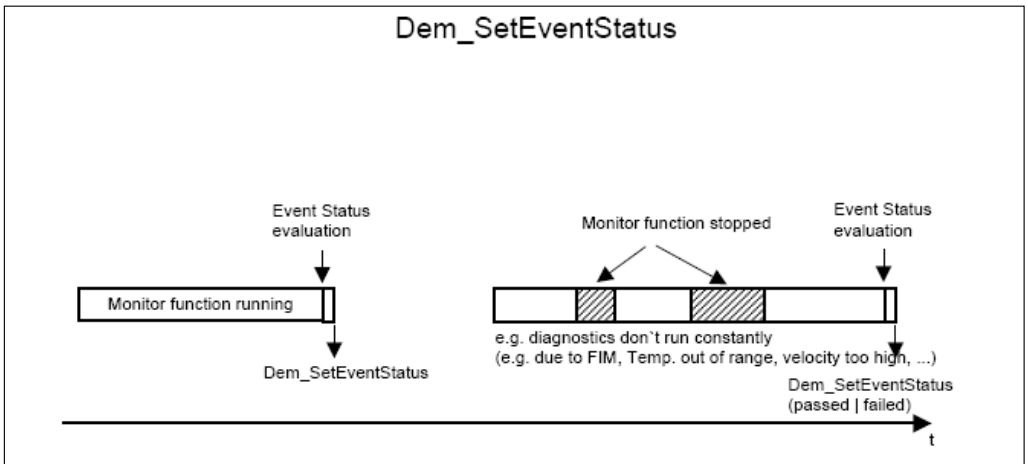


Figure 7.14: Example for using Dem_SetEventStatus

[SWS_Dem_00331]

Upstream requirements: RS_Diag_04179

⌈The Dem module shall provide the capability to reset the failed status of an event without reporting a passed result (refer to Dem_ResetEventStatus).⌋

Note: Dem_ResetEventStatus does not mean an event status report (like done by Dem_SetEventStatus). After the call, the monitor status is not qualified or tested.

Monitors will use the function Dem_ResetEventStatus in order to deactivate limp home and switch back to normal operation. At this point in time, the monitor has typically not been OK-tested and therefore Dem_SetEventStatus with tested and passed cannot be used.

[SWS_Dem_00187] Behavior of Dem_ResetEventStatus

Upstream requirements: RS_Diag_04067

⌈The function Dem_ResetEventStatus shall set the event status bit 0 (TestFailed) to 0 and reset the Dem-internal debounce algorithm to initial values if configured. The Dem shall execute this action asynchronously at a later point in time.⌋

A later point in time means that the exact moment is implementation specific. This could be the next main function and after Dem_ResetEventStatus has returned. Also a possible call of the status change callback is triggered asynchronously.

Note: The function Dem_ResetEventStatus does not change the monitor status and UDS status bit 6 and does not clear the pre-stored freeze frame.

[SWS_Dem_00638]

Upstream requirements: RS_Diag_04067

⌈The function Dem_ResetEventStatus shall return E_NOT_OK, if the event was already tested this operation cycle (UDS status bit 6 is set to 0).⌋

[SWS_Dem_01390] Behavior of Dem_ResetMonitorStatus ⌈The function Dem_ResetMonitorStatus shall set the monitor status bit 0 (TestFailed) to 0 and reset the Dem-internal debounce algorithm to initial values if configured.⌋

Note: The function Dem_ResetMonitorStatus does not change the UDS status byte and does not clear the pre-stored freeze frame.

[SWS_Dem_01401] ⌈The function Dem_ResetMonitorStatus shall return E_NOT_OK, if the monitoring was already tested this operation cycle (monitor status byte, bit 1 set to 0).⌋

[SWS_Dem_00051]

Upstream requirements: RS_Diag_04067

⌈If the API Dem_GetEventUdsStatus is called, the Dem shall provide the current event status byte of a specific event.⌋

Note: The function Dem_GetEventUdsStatus is provided to be used by SW-Cs or other BSW modules (e.g. FiM) on event-level. The Dcm uses the function Dem_GetStatusOfDTC on DTC-level instead.

[SWS_Dem_00844]

Upstream requirements: RS_Diag_04068

⌈The Dem shall provide a function Dem_GetDebouncingOfEvent () that reports the debounce status of an event. The outparameter 'DebouncingState' returns the debouncing incl. intermediate states. One particular OBD specific bit shall support the DTR update trigger if test is complete and debouncing is at its limit while the enable and storage conditions are met.⌋

The Dem supports a configuration per event to forbid to reset the event in the same operation cycle.

[SWS_Dem_01208]

Upstream requirements: RS_Diag_04151

⌈If the configuration parameter DemEventRecoverableInSameOperationCycle is set to FALSE, a report of PASSED (directly via Dem_SetEventStatus or indirectly after Debouncing) will be ignored if the event status byte has the "TestFailedThisOperationCycle" flag is set to 1.⌋

Note: Debouncing is done independently from the value DemEventRecoverableIn-SameOperationCycle. The FDC can reach -128 independently from the setting of DemEventRecoverableInSameOperationCycle.

[SWS_Dem_01209]

Upstream requirements: RS_Diag_04151

⌈The Dem shall perform the configured debouncing independently from the setting of the configuration parameter DemEventRecoverableInSameOperationCycle. I.e. if DemEventRecoverableInSameOperationCycle is set to FALSE and the monitor for this event reports PREPASSED the Dem shall still process the debouncing (e.g. decrement the counter).⌋

[SWS_Dem_01250]

Upstream requirements: RS_Diag_04165

⌈If DemMultiEventTriggering is configured and any of the APIs:

- Dem_SetEventStatus
- Dem_ResetEventStatus
- Dem_PrestoreFreezeFrame
- Dem_ClearPrestoredFreezeFrame
- Dem_ResetEventDebounceStatus

is called with an EventId that is referenced by DemMultiEventTriggering-MasterEventRef of a DemMultiEventTriggering, for each DemMultiEvent-TriggeringSlaveEventRef of this DemMultiEventTriggering the Dem shall trigger the same API calls with identical parameters only the EventId parameter shall be replaced with the event referenced from DemMultiEventTriggeringSlave-EventRef.⌋

[SWS_Dem_01339]

Upstream requirements: RS_Diag_04165

⌈The Dem shall not support calls of

- Dem_SetEventStatus
- Dem_ResetEventStatus
- Dem_PrestoreFreezeFrame
- Dem_ClearPrestoredFreezeFrame
- Dem_ResetEventDebounceStatus

with an EventId that is referenced by any of the DemMultiEventTriggering-SlaveEventRef in container DemMultiEventTriggering or by DemStorage-ConditionReplacementEventRef. These events are exclusively used for internal

triggering. The behavior of the Dem is undefined if any of those APIs are called in this situation.⌋

7.7.1.1 Status bit support

[SWS_Dem_00006]

Upstream requirements: RS_Diag_04067

⌈The Dem module shall by default implement the current set of all UDS status bits according to the definition in ISO-14229-1[2] for each diagnostic event.⌋

Note: For this specification, the UDS status byte of ISO-14229-1[2] for events is represented by the Dem data type Dem_UdsStatusByteType (defined in chapter 8.6.1.21). If particular UDS status bits do not need to be supported (refer also to [SWS_Dem_00060]), such optimizations or limitations may be implemented vendor-specific.

[SWS_Dem_01276]

Upstream requirements: RS_Diag_04067

⌈If the API Dem_GetEventUdsStatus is called, the event from the parameter EventId exists and is available, the Dem shall copy the UDS status byte into the out parameter UdsStatusByte and return E_OK.⌋

For details see chapter 7.4.8.

[SWS_Dem_01277]

Upstream requirements: RS_Diag_04067

⌈If the API Dem_GetEventUdsStatus is called and the event from the parameter EventId is unavailable, the Dem shall return E_NOT_OK.⌋

For details see chapter 7.4.8.

7.7.1.2 Monitor status and UDS status update

Status bit processing is an essential part of the Dem functionality. SWCs or BSW modules frequently report monitor results via the API Dem_SetEventStatus. A call of these functions will trigger the status bit processing of the Dem.

According to ISO 14229-1[2] monitor, event or DTC status cannot be used to control safety critical functionality. There are various situations e.g. enable conditions or the UDS service ControlDTCSetting that can block processing of these status information.

[SWS_Dem_01278]

Upstream requirements: RS_Diag_04067

⌈The Dem shall provide a monitor status per event to store current monitor state information. The monitor status contains the information as defined in Dem_MonitorStatusType.⌋

[SWS_Dem_01279]

Upstream requirements: RS_Diag_04068

⌈The Dem shall perform the counter based event debouncing synchronously within the context of the calling Dem_SetEventStatus function.⌋

[SWS_Dem_01280]

Upstream requirements: RS_Diag_04067

⌈Upon call of Dem_SetEventStatus and after debouncing, the Dem shall process the monitor status in the same manner as the UDS status:

- Bit0 (TestFailed) according to [SWS_Dem_00386]
- Bit1 (TestNotCompleteThisOperationCycle) according to [SWS_Dem_00394].

The monitor status byte processing shall be done synchronously within the context of the calling Dem_SetEventStatus function.⌋

[SWS_Dem_01281]

Upstream requirements: RS_Diag_04067

⌈If the parameter DemGeneralCallbackMonitorStatusChangedFnc is configured, the Dem shall call this configured C function with the signature of <Module>_DemGeneralTriggerOnMonitorStatus.⌋

[SWS_Dem_01282]

Upstream requirements: RS_Diag_04067

⌈Upon processing the monitor status, the Dem shall call the monitor status changed callback functions:

- <Module>_DemTriggerOnMonitorStatus

whenever the monitor status has changed.⌋

[SWS_Dem_01283]

Upstream requirements: RS_Diag_04067

⌈Upon processing the monitor status for reported event with DemEventKind = DEM_EVENT_KIND_SWC, the Dem shall call the monitor status changed callback C/S Interface CallbackMonitorStatusChange whenever the monitor status has changed and a C/S interface is configured for this event.⌋

[SWS_Dem_CONSTR_06120] Dependency for DemGeneralCallbackMonitorStatusChangedFnc ⌈The DemGeneralCallbackMonitorStatusChangedFnc shall only be present if DemGeneralInterfaceSupport is set to TRUE."⌋

Limitation: The usage of monitorStatusChange via RTE is limited to SWC-monitorings. All basic software monitorings cannot be covered with this callback via RTE. To get complete statusChange information of all monitorings in a SW-C, the eventStatusChange callbacks should be used.

The CallbackMonitorStatusChange and GeneralCallbackMonitorStatusChanged via RTE is furthermore limited to monitorings, which are exclusively changing its status in known contexts. E.g. if a monitor status is changed from CDD (e.g. Dem_SetEventAvailable via C-function call), the monitor status change callback via RTE must not be used.

Rational: AUTOSAR RTE requires that calls from a BSW to a SW-C via RTE are done in a known call context. For the synchronous monitor status changed callbacks, this is only possible for events reported from a SW-C. For events reported from a BSW module it is not possible to provide the calling context to the RTE. As a result of this, the Dem provides the C/S interface CallbackMonitorStatusChange only for events that are reported from a SW-C. For events reported from BSW modules, no such C/S interface is provided. Furthermore, the interface GeneralCallbackMonitorStatusChanged is not provided as C/S via RTE, as this callback could only be called for events reported from a SW-C and not for events reported from BSW modules. This would obviously lead to confusion.

If it is required to have a general synchronous reaction in a SW-C on monitor status changed callbacks, the synchronous reaction could be handled within the scope of a CDD using <Module>_DemGeneralTriggerOnMonitorStatus or <Module>_DemTriggerOnMonitorStatus.

[SWS_Dem_CONSTR_06116] Limited use of monitor status change callbacks to events reported from SW-Cs only ⌈In case of a BSW event (DemEventKind = DEM_EVENT_KIND_BSW), the corresponding monitor status changed callback can only be used as C-function, but not via RTE interface, thus the DemCallbackMonitorStatusChangedFnc parameter shall be filled.⌋

[SWS_Dem_01284]

Upstream requirements: RS_Diag_04067

⌈When processing a clear DTC command (e.g. via Dem_ClearDTC), the Dem shall set the monitor status assigned to the cleared DTCs to 0x02.⌋

[SWS_Dem_01285]

Upstream requirements: RS_Diag_04067

⌈The Dem shall process the UDS status asynchronously. This means it is calculated outside the context of the calling Dem_SetEventStatus function.⌋

[SWS_Dem_01286]

Upstream requirements: RS_Diag_04067

⌈If the API `Dem_SetEventStatus` is called and the `Dem` has synchronously processed the reported monitor status according to [SWS_Dem_01280] and the corresponding UDS status (see [SWS_Dem_01285]) cannot be processed, the `Dem` shall report the runtime error DEM_E_UDS_STATUS_PROCESSING_FAILED.⌋

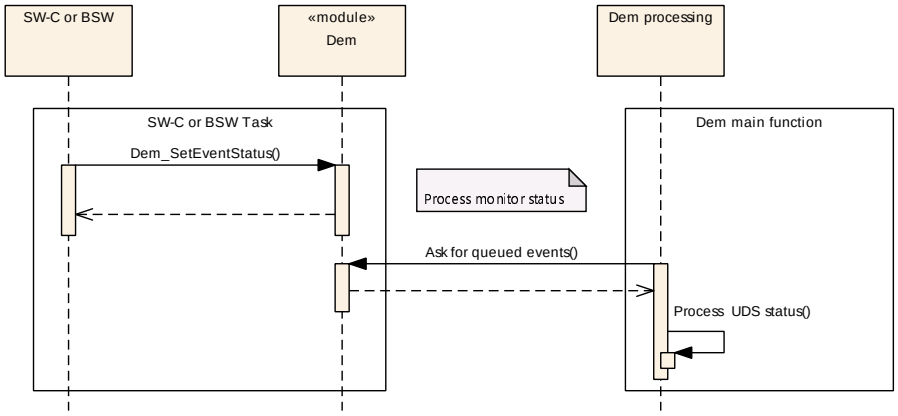


Figure 7.15: Synchronous and asynchronous event processing

Note: Refer to chapter 7.7.2 for a detailed description about the `event memory` management.

[SWS_Dem_01287]

Upstream requirements: RS_Diag_04067

⌈If the `API Dem_GetMonitorStatus` is called, the `Dem` shall copy the current monitor status into the out parameter MonitorStatus and return E_OK.⌋

[SWS_Dem_01326] Monitor status initialization

Upstream requirements: RS_Diag_04031

⌈The `Dem` shall initialize the monitor status in the following way:

- If `Dem_SetEventStatus` is called between `Dem_PreInit` and `Dem_Init`, the reported monitor status shall be used.
- If no monitor status was reported before `Dem_Init`, the Dem shall use the event status recovered from NvM as the monitor status.

⌋

[SWS_Dem_01419] Monitor status availability

Upstream requirements: RS_Diag_04031

⌈If `Dem_GetMonitorStatus` is called before `Dem_Init` has initialized the monitor status, the `Dem` shall return E_NOT_OK.⌋

7.7.1.3 Status bit transitions

This section describes the behavior of the individual status bits of the UDS status byte (refer to ISO-14229-1[2], UDS status byte - bit transitions).

[SWS_Dem_00385]

Upstream requirements: RS_Diag_04067

⌈After a clear command has been applied to a specific DTC the Dem module shall set the UDS status byte to 0x50 (readiness bits 4 and 6 set to 1, and all others are set to zero).⌋

For details see chapter 7.7.2.2.

Note: The value of the UDS status byte after a clear command has been applied represents the delivery status (initial state) of this byte as well.

[SWS_Dem_00386] TestFailed Bit behavior

Upstream requirements: RS_Diag_04067

⌈The Dem module shall implement the event and DTC status bit transitions for TestFailed (Bit 0) according to ISO 14229-1:2020.⌋

For details see figure 7.16.

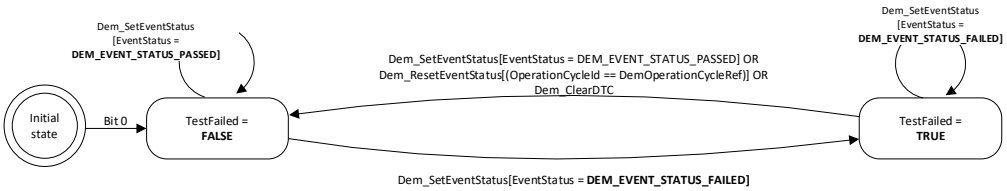


Figure 7.16: UDS status bit 0 TestFailed logic

[SWS_Dem_00389]

Upstream requirements: RS_Diag_04067

⌈The Dem module shall implement the status bit transition for UDS status bit 1.⌋

For details see figure 7.17.

Note: The event status bit 1 (TestFailedThisOperationCycle) is stored in non-volatile memory.

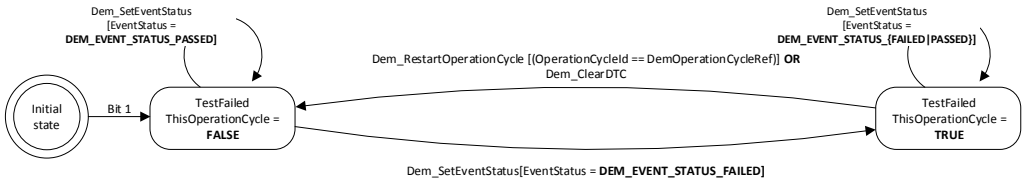


Figure 7.17: UDS status bit 1 TestFailedThisOperationCycle logic

[SWS_Dem_00390]

Upstream requirements: RS_Diag_04067

⌈The Dem module shall implement the status bit transition for `UDS status bit 2`.⌋

For details see figure 7.18.

[SWS_Dem_01183] ⌈The information for `UDS status bit 2` needs to be stored non-volatile⌋

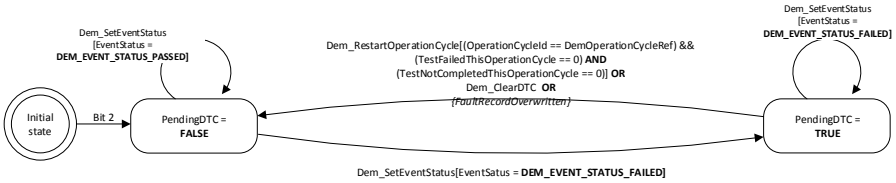


Figure 7.18: UDS status bit 2 PendingDTC logic

[SWS_Dem_00391]

Upstream requirements: RS_Diag_04067

⌈The Dem module shall implement the status bit transition for `UDS status bit 3`.⌋

For details see figure 7.19.

Note: The information for event status bit 3 (ConfirmedDTC) is non-volatile (but it is also calculable based on the respective `event memory` entry).

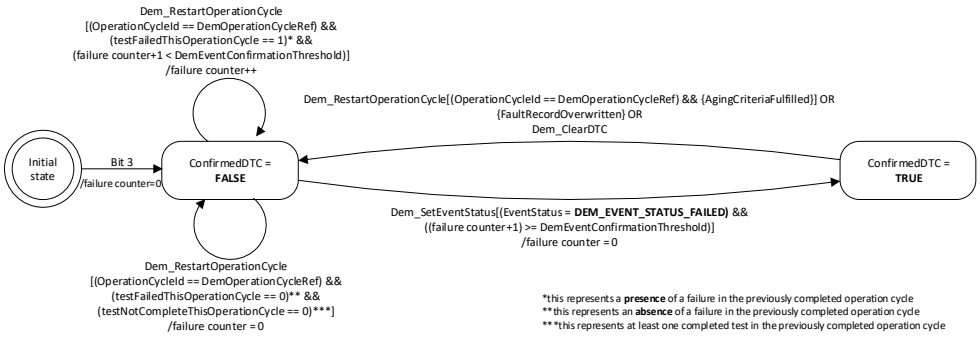


Figure 7.19: UDS status bit 3 ConfirmedDTC logic

The figure is applicable If `DemOBDDelayedDCYConfirmedAndMIL` is set to false. If `DemOBDDelayedDCYConfirmedAndMIL` is set to true the confirmed bit transition to 1 dependes further more on the operation cylcle qualification, see [SWS_Dem_01220].

Note: The “AgingCriteriaFulfilled” condition is specified by [SWS_Dem_00498]. The “FaultRecordOverwritten” condition is specified by [SWS_Dem_00409].

The parameter `DemEventConfirmationThreshold` controls the operation cycle in which the confirmedDTC bit is set based on the first occurrence of the failure. The confirmedDTC bit is always set to 1 when the fault occurs, but the operation cycle in which this is done is depending on the use case.

The value is relevant for OEM and legislated requirements on setting the confirmed-DTC bit from 0 to 1. In the Table 7.1 below, the most common use cases and necessary values are given:

DemEventConfirmationThreshold	*Use Case*
1	The confirmedDTC bit is set from 0 to 1 the first time the testFailed bit is set from 0 to 1. This behavior is also called "immediate confirmation" or "zero trip". This setting is mostly used for non-OBD UDS DTCs.
2	The confirmedDTC bit is set from 0 to 1 in the operation cycle after the failure was reported the first time. This setting is used to fulfill legislated OBD requirements in the US.
3	The confirmedDTC bit is set from 0 to 1 in the second operation cycle after the failure was reported the first time. This setting is used to fulfill legislated OBD requirements in Europe.

Table 7.1: Common values for DemEventConfirmationThreshold to solve typical requirements

Other OEM specific values for `DemEventConfirmationThreshold` beside the mentioned values from Table 7.1 are also possible.

The parameter `DemEventConfirmationThreshold` is an AUTOSAR Dem specific threshold value that is by "1" greater than the "trip counter" that is used in ISO 14229-1:2020[2] figure D.4.

[SWS_Dem_00392]

Upstream requirements: RS_Diag_04067

⌈The Dem module shall implement the status bit transition for `UDS status bit 4`.⌋

For details see figure 7.20.

Note: The information for event status bit 4 (TestNotCompletedSinceLastClear) is non-volatile.

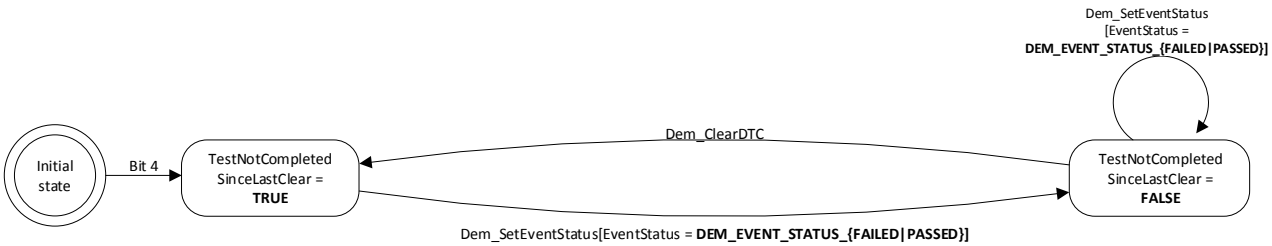


Figure 7.20: UDS status bit 4 TestNotCompletedSinceLastClear logic

[SWS_Dem_00393]

Upstream requirements: RS_Diag_04067

⌈The Dem module shall implement the status bit transition for `UDS status bit 5`.⌋

For details see figure 7.21.

Note: The information for UDS status bit 5 (TestFailedSinceLastClear) is non-volatile.

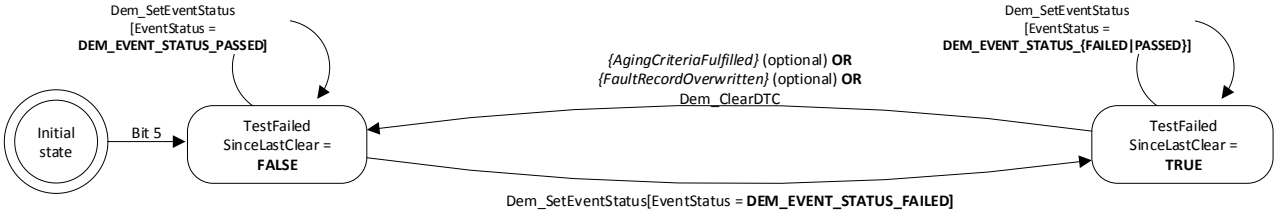


Figure 7.21: UDS status bit 5 TestFailedSinceLastClear logic

Note: The relevance of “AgingCriteriaFulfilled” condition (specified by [SWS_Dem_00498]) and “FaultRecordOverwritten” condition (specified by [SWS_Dem_00409]) depends on the configuration parameter `DemStatusBitHandlingTestFailedSinceLastClear` (refer to `DemGeneral`).

[SWS_Dem_00394]

Upstream requirements: RS_Diag_04067

⌈The Dem module shall implement the status bit transition for `UDS status bit 6`.⌋

For details see figure 7.22.

Note: The event status bit 6 (TestNotCompleteThisOperationCycle) is stored in non-volatile memory.

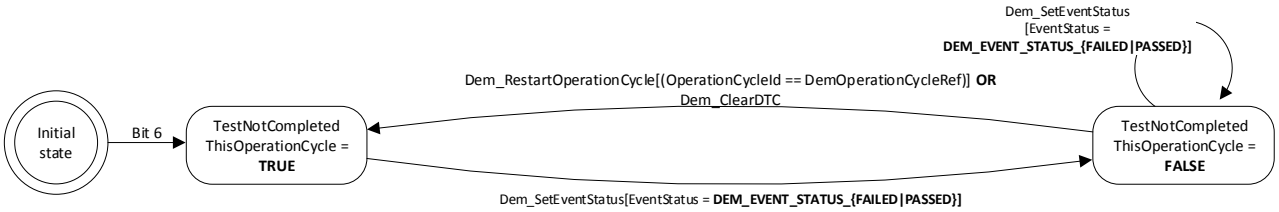


Figure 7.22: UDS status bit 6 TestNotCompleteThisOperationCycle logic

[SWS_Dem_00395]

Upstream requirements: RS_Diag_04067

⌈The Dem module shall implement the status bit transition for UDS status bit 7 (WarningIndicatorRequested).⌋

For details see figure 7.23.

Note: The information for UDS status bit 7 (WarningIndicatorRequested) may be volatile (because it is calculated based on the assigned warning indicator states).

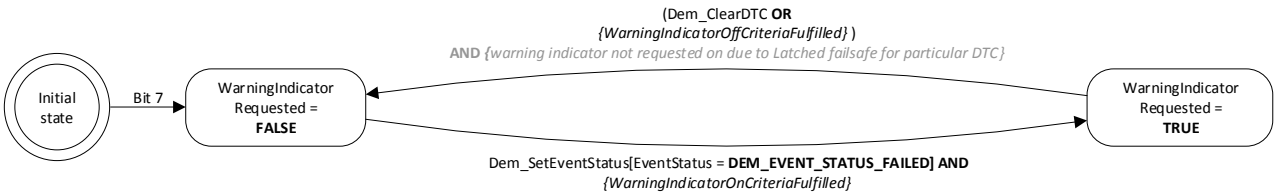


Figure 7.23: UDS status bit 7 WarningIndicatorRequested logic

Note: The "WarningIndicatorOffCriteriaFulfilled" and the "WarningIndicatorOnCriteriaFulfilled" condition are specified in chapter 7.7.9.

Note: ISO-14229-1[2] additionally specifies "warning indicator not requested on due to latched failsafe for particular DTC" as condition. This has to be ensured by the monitor.

7.7.1.4 Active/Passive status

If an event gets qualified as failed, it becomes active. If the event gets qualified as passed, it becomes passive. This status can be derived from the event status byte.

The definition of "event active" equals to TestFailed = 1 and "event passive" equals to TestFailed = 0.

7.7.1.5 Notification of status bit changes

The Dem module can inform the monitor and/or other components about the status change of the event or DTC.

Note: For asynchronous event processing, the Dem module will trigger the respective notification functions twice per event/DTC qualification.

[SWS_Dem_00016]

Upstream requirements: RS_Diag_04067

⌈The `Dem` module shall trigger the event-specific callback-function EventUdsStatusChanged on each UDS status change.⌋

Note: The Dem module does not evaluate the return value (e.g. if other than E_OK) of this callback function.

Note: The configuration container `DemCallbackEventUdsStatusChanged` (in `DemEventParameter`) is used to specify one or more ports/c-callbacks per event.

[SWS_Dem_01426] General Event UDS Status Changed notification

Upstream requirements: RS_Diag_04067, RS_Diag_04148, RS_Diag_04183

⌈If the parameter `DemGeneralCallbackEventUdsStatusChangedFnc` is configured, the `Dem` shall call the General Event UDS Status Changed callback function

<DemGeneralTriggerOnEventUdsStatus> whenever the UDS status byte of an event has changed.⌋

Note: The respective callback-functions for FiM are specified in chapter 7.10.

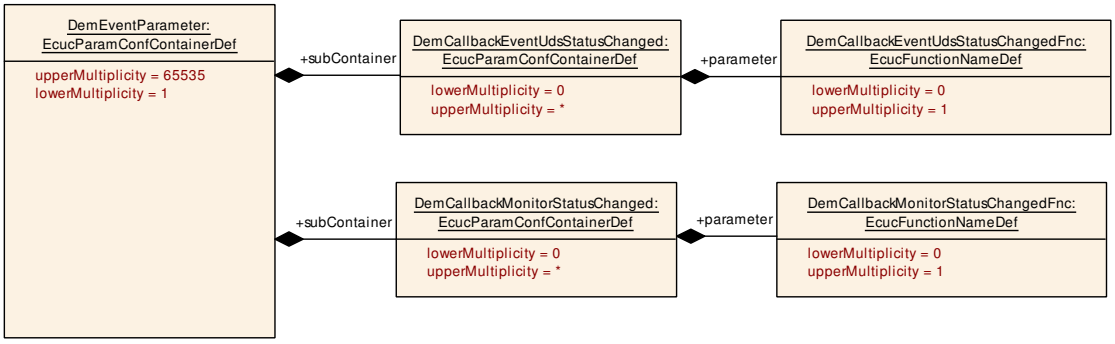


Figure 7.24: EventStatusChanged callback configuration

[SWS_Dem_00284]

Upstream requirements: SRS_BSW_00457

⌈The Dem module shall trigger the callback function configured in DemCallbackDTCStatusChanged on every DemDTC status change if primary memory is the configured DTC origin.⌋

[SWS_Dem_00986]

Upstream requirements: SRS_BSW_00457

⌈The Dem module shall trigger the callback function configured in DemCallbackOBDDTCStatusChanged either the 2-byte or the 3-byte OBD DTC on every DemObdDTC status change.⌋

Note: OBD DTCs are limited to primary memory, see [SWS_Dem_CONSTR_06172].

[SWS_Dem_00987]

Upstream requirements: SRS_BSW_00457

⌈The Dem module shall trigger the callback function configured in DemCallbackJ1939DTCStatusChanged on every J1939 DTC status change if primary memory is the configured DTC origin.⌋

Note: The Dem module does not evaluate the return value (e.g. if other than E_OK) of this callback function.

Note: The result of EventUdsStatusChanged may only be different to the result of DTCStatusChanged in case of event combination (refer to chapter 7.7.5).

Note: The configuration containers DemCallbackDTCStatusChanged,DemCallbackOBDDTCStatusChanged and DemCallbackJ1939DTCStatusChanged are used to specify one or more ports/c-callbacks for the Dem module globally.

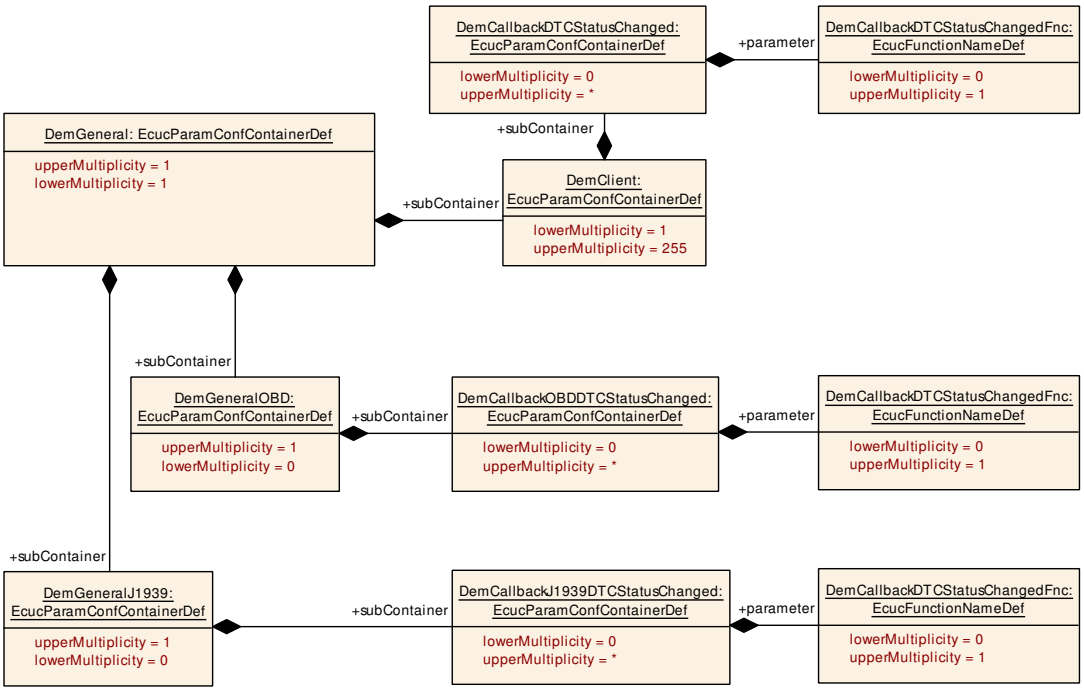


Figure 7.25: DTCStatusChanged callback configuration

7.7.2 Event memory management

The event memory management defines as the process of adding, updating and removing event memory entries in and out of the Dem module. The Dem module determines if the event memory entry is new or currently exists in the event memory.

Note: The requirements in this section do not determine the software implementation but describe the behavior of the ECU towards requests from the test tool.

[SWS_Dem_01199] ⌈The Dem module shall support the configuration parameter DemMemoryDestinationRef to define the destination memory/memories for individual DTCs.⌋

When using the Dem interfaces, Dem_DTCOriginType selects the destination memories by evaluating the possible DemMemoryDestinationRef.

[SWS_Dem_01063]

Upstream requirements: RS_Diag_04131

⌈The user defined memory shall have the same behavior as the primary memory(event retention, event prioritization, aging, displacement).⌋

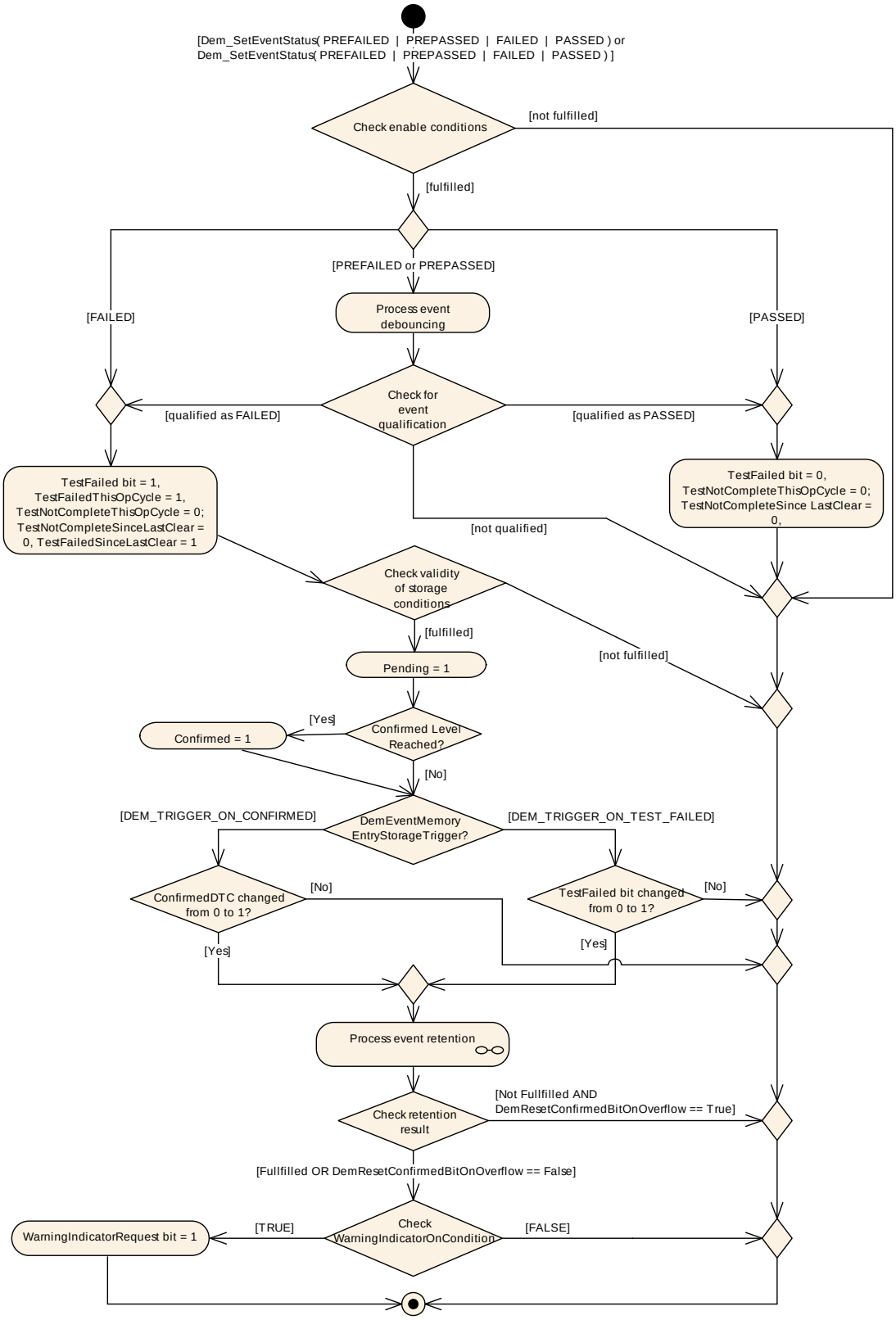


Figure 7.26: General diagnostic event storage processing

7.7.2.1 Event retention

Event retention defines the ability of the Dem module to record and handle events (DTCs), UDS status information and event related data (e.g. freeze frames, extended data).

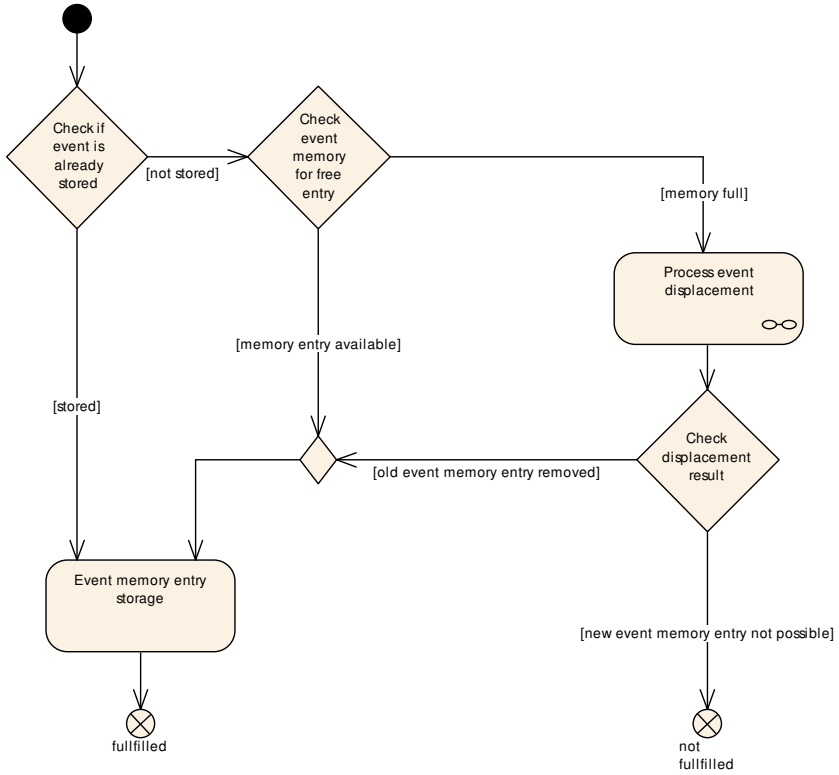


Figure 7.27: General diagnostic event retention processing

There are several different strategies in the market when to allocate an `event memory entry`. Therefore, the Dem module has the capability to configure the primary trigger to allocate an `event memory entry`. Beside the primary trigger there might be secondary triggers (refer `DemFreezeFrameRecordTrigger`) mainly to update content of event `event memory` (e.g. to have most recent update FreezeFrames).

[SWS_Dem_00783]

Upstream requirements: RS_Diag_04105

⌈If an event

1. gets qualified as failed (`UDS status bit 0` changes from 0 to 1) and
2. the configuration parameter `DemEventMemoryEntryStorageTrigger` is set to DEM_TRIGGER_ON_TEST_FAILED or DEM_TRIGGER_ON_FDC_THRESHOLD (refer to `DemPrimaryMemory` or `DemUserDefinedMemory`) and
3. no `event memory entry` exist,

the Dem module shall try to allocate an event memory entry in its configured event memory (refer to DemMemoryDestinationRef).⌋

For details see figure 7.27.

[SWS_Dem_00784]

Upstream requirements: RS_Diag_04105

⌈If an event

1. gets pending (UDS status bit 2 changes from 0 to 1) and
2. the configuration parameter DemEventMemoryEntryStorage-Trigger is set to DEM_TRIGGER_ON_TEST_FAILED or DEM_TRIGGER_ON_FDC_THRESHOLD (refer to DemPrimaryMemory or DemUserDefinedMemory) and
3. no event event memory is existing,

the Dem module shall try to allocate an event memory entry in its configured event memory (refer to DemMemoryDestinationRef).⌋

For details see figure 7.27.

[SWS_Dem_00922]

Upstream requirements: RS_Diag_04105

⌈If an event

1. is pending (UDS status bit 2 is set to 1) and gets re-qualified as failed (UDS status bit 0 changes from 0 to 1) and
2. the configuration parameter DemEventMemoryEntryStorage-Trigger is set to DEM_TRIGGER_ON_TEST_FAILED or DEM_TRIGGER_ON_FDC_THRESHOLD (refer to DemPrimaryMemory or DemUserDefinedMemory) and
3. no event memory entry is existing,

the Dem module shall try to allocate an event memory entry in its configured event memory (refer to DemMemoryDestinationRef).⌋

For details see figure 7.27.

[SWS_Dem_00785]

Upstream requirements: RS_Diag_04105

⌈If an event

1. gets confirmed (UDS status bit 3 changes from 0 to 1) and
2. the configuration parameter DemEventMemoryEntryStorageTrigger is set to DEM_TRIGGER_ON_CONFIRMED or DEM_TRIGGER_ON_TEST_FAILED

or DEM_TRIGGER_ON_FDC_THRESHOLD (refer to DemPrimaryMemory or DemUserDefinedMemory) and

3. no event memory entry is existing,

the Dem module shall try to allocate an event memory entry in its configured event memory (refer to DemMemoryDestinationRef).⌋

For details see figure 7.27.

[SWS_Dem_00923]

Upstream requirements: RS_Diag_04105

⌈If an event

1. is confirmed (UDS status bit 3 is set to 1) and gets re-qualified as failed (UDS status bit 0 changes from 0 to 1) and
2. the configuration parameter DemEventMemoryEntryStorageTrigger is set to DEM_TRIGGER_ON_CONFIRMED or DEM_TRIGGER_ON_TEST_FAILED or DEM_TRIGGER_ON_FDC_THRESHOLD (refer to DemPrimaryMemory or DemUserDefinedMemory) and
3. no event memory entry is existing,
4. if DemResetConfirmedBitOnOverflow is set to false,

the Dem module shall try to allocate an event memory entry in its configured event memory (refer to DemMemoryDestinationRef).⌋

For details see figure 7.27.

[SWS_Dem_00786]

Upstream requirements: RS_Diag_04105

⌈If an event

1. uses Dem-internal debouncing (either counterbased or timebased) and
2. the Dem-internal debounce counter is incremented and reaches DemCounterBasedFdcThresholdStorageValue or DemTimeBasedFdcThresholdStorageValue and
3. the configuration parameter DemEventMemoryEntryStorageTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD (refer to DemPrimaryMemory or DemUserDefinedMemory) and
4. no event event memory exists,

the Dem module shall try to allocate an event memory entry in its configured event memory (refer to DemMemoryDestinationRef).⌋

For details see figure 7.27.

Note: In case the event memory entry already exists, there might be other triggers to update content of event memory entry(refer chapter 7.7.7 “Event related data”).

[SWS_Dem_CONSTR_06155] Dependency on DemTimeBasedFdcThresholdStorageValue ⌈DemTimeBasedFdcThresholdStorageValue shall only be present if DemFreezeFrameRecordTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD or DemExtendedDataRecordTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD or DemEventMemoryEntryStorageTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD (refer to DemPrimaryMemory or DemUserDefinedMemory).⌋

7.7.2.2 Clearing event memory entries

The event memory entries of the Dem module can be cleared DTC-based via various diagnostic services (e.g. service 0x14 ClearDiagnosticInformation or service $04 Clear/Reset emission-related diagnostic information). Therefore, the Dem offers separate APIs for different users:

- Dem_ClearDTC
- Dem_J1939DcmClearDTC (for J1939 to the J1939Dcm [7], refer to chapter 7.10.2).
- Dem_J1939DcmClearSingleDTC (for individual DM22 DTC clearing; refer to chapter 7.10.2).

[SWS_Dem_01203]

Upstream requirements: RS_Diag_04150, RS_Diag_04214

⌈If the Dem module is requested to clear diagnostic information and the selected DTC is set to DTC group ‘all DTCs’ and the selected DTCOrigin is set to DEM_DTC_ORIGIN_PRIMARY_MEMORY or DEM_DTC_ORIGIN_USERDEFINED_MEMORY_<NAME>, the Dem shall reset all event and DTC status bytes and clear event related data assigned to the requested DTCOrigin.⌋

[SWS_Dem_01206]

Upstream requirements: RS_Diag_04150

⌈The Dem shall call the callback DemClearEventAllowed only if the selected DTCOrigin for the clear operation is the primary or user defined event memory.⌋

The clearing process can be started by different clients. Whenever one of these interfaces starts the clearing process, the Dem-internal clearing mechanism is locked until the clear request is finished and the result is returned to the requesting user.

[SWS_Dem_01240]

Upstream requirements: RS_Diag_04185

⌈If DemClearDtcNotificationTime is configured to START, the Dem shall inform the application about the start of a clear DTC operation. If the DemClearDtc-NotificationFnc is configured, the Dem shall shall call this configured function otherwise the Dem shall call the R-Port ClearDtcNotification.⌋

[SWS_Dem_01241]

Upstream requirements: RS_Diag_04185

⌈If DemClearDtcNotificationTime is configured to FINISH, the Dem shall inform the application about the end of a clear DTC operation. The Dem shall consider the configuration parameter DemClearDTCBehavior which defines the clearing process of diagnostic information including clearing completion. If the DemClearDtc-NotificationFnc is configured, the Dem shall call this configured function otherwise the Dem shall call the R-Port ClearDtcNotification.⌋

[SWS_Dem_00661]

Upstream requirements: RS_Diag_04194

⌈In case of multiple clients are requesting Dem_ClearDTC in parallel, the Dem shall internally process the different requests.⌋

[SWS_Dem_00664] ⌈If the clear DTC process is started by one client and still ongoing, then in case another client calls Dem_ClearDTC, the Dem shall return DEM_PENDING⌋

[SWS_Dem_00670]

Upstream requirements: RS_Diag_04117

⌈If the configuration parameter DemClearDTCLimitation is set to DEM_ONLY_CLEAR_ALL_DTCS, the APIs Dem_ClearDTC are called with a selected DTC group different to DEM_DTC_GROUP_ALL_DTCS the Dem shall return DEM_WRONG_DTC.⌋

[SWS_Dem_00570]

Upstream requirements: RS_Diag_04194

⌈If the Dem module is requested to clear diagnostic information and the configuration parameter DemClearDTCBehavior is set to DEM_CLRRESP_VOLATILE, the Dem module shall return E_OK after the volatile memory is cleared.⌋

[SWS_Dem_00571]

Upstream requirements: RS_Diag_04194

⌈If the Dem module is requested to clear diagnostic information and the configuration parameter DemClearDTCBehavior is set to DEM_CLRRESP_NONVOLATILE_TRIGGER, the Dem module shall return E_OK after the volatile memory is cleared and clearing of the non-volatile memory is triggered.⌋

[SWS_Dem_00572]

Upstream requirements: RS_Diag_04194

⌈If the Dem module is requested to clear diagnostic information and the configuration parameter DemClearDTCBehavior is set to DEM_CLRRESP_NONVOLATILE_FINISH, the Dem module shall return E_OK after the volatile memory and the non-volatile memory is cleared.⌋

Note: The Dem implementation is responsible for consistency of the persistant data. (See SWS_NvM_00698: Contents of RAM buffer passed to NvM_WriteBlock may not change while the NvM processes the job). How to ensure data consistency (UDS status byte, freeze frame data and all other related information) is implementation specific (depends on the implemented NvM strategy). Options include, but are not restricted to, additional RAM mirrors and queuing of changes until the RAM area can be modified again.

Note: If the Dcm module receives the return type DEM_CLEAR_OK of the API Dem_ClearDTC, the Dcm module sends the positive response (refer to [9]).

[SWS_Dem_01057] ⌈If the Dem module is requested to clear diagnostic information and the configuration parameter DemClearDTCBehavior is set to DEM_CLRRESP_NONVOLATILE_FINISH, the Dem module shall return DEM_CLEAR_MEMORY_ERROR if the clearing of the non-volatile memory fails.⌋

[SWS_Dem_01202] Defintion of a failed clear DTC operation

Upstream requirements: RS_Diag_04150, RS_Diag_04214

⌈If the Dem module is requested to clear diagnostic information, the Dem shall return DEM_CLEAR_FAILED:

- in case of clearing a single DTC at least one DemClearEventAllowed callback assigned to the event of the DTC returns FALSE.
- in case of clearing a DTC group or all DTCs, the individual DTCs shall be cleared like described in clearing single DTC. The resulting return value shall be DEM_CLEAR_FAILED if the clearing of all individual DTCs returned DEM_CLEAR_FAILED.

⌋

[SWS_Dem_00573]

Upstream requirements: RS_Diag_04194

⌈If DemTriggerMonitorInitBeforeClearOk is set to TRUE, the Dem_ClearDTC shall not return E_OK until the events intialization is triggered (callback function InitMonitorForEvent is called). If DemTriggerMonitorInitBeforeClearOk is set to FALSE, DEM_CLEAR_OK is not considering Event initialization (InitMonitorForEvent).⌋

This functionality is intended for some special events, which must never be cleared from event memory within specific states, like for example special operation modes of the ECU (e.g. assembly-, transport-, or flash-mode) or Dcm sessions (which can be determined, using API Dcm_GetSesCtrType).

It is configurable per event (refer to DemCallbackClearEventAllowed), whether a SW-C / BSW module shall be requested about the event-deletion allowance from this callback function. One callback (representing a specific condition) can be assigned per event. If multiple conditions apply, they need to be summarized by one callback individually.

[SWS_Dem_00515]

Upstream requirements: RS_Diag_04194

⌈The Dem module shall call the callback DemClearEventAllowed for each event it is configured for, before clearing this event.⌋

[SWS_Dem_00667]

Upstream requirements: RS_Diag_04194

⌈If the DTC is not configured to use event combination, the out-parameter Allowed of the callback DemClearEventAllowed returns “false” and the return value is equal to E_OK, the Dem shall not clear the respective event memory entry and Dem-internal data values.⌋

[SWS_Dem_01295]

Upstream requirements: RS_Diag_04194

⌈If the DTC is configured to use event combination, if only one DemClearEventAllowed callback has the out-parameter Allowed set to FALSE and returns E_OK, the Dem shall not clear the event status, respective event memory entry and Dem-internal data values.⌋

[SWS_Dem_00668]

Upstream requirements: RS_Diag_04194

⌈If the out-parameter ‘Allowed’ of the callback DemClearEventAllowed returns “false” and the configuration parameter DemClearEventAllowedBehavior is set to DEM_NO_STATUS_BYTE_CHANGE, the related UDS status byte shall not be modified.⌋

[SWS_Dem_00669]

Upstream requirements: RS_Diag_04194

⌈If the out-parameter Allowed of the callback DemClearEventAllowed returns “false” and the configuration parameter DemClearEventAllowedBehavior is set to DEM_ONLY_THIS_CYCLE_AND_READINESS, the related UDS status bits 1 (TestFailedThisOperationCycle), 4 (TestNotCompletedSinceLastClear), 5 (TestFailedSinceLastClear), and 6 (TestNotCompletedThisOperationCycle) shall be reset.⌋

[SWS_Dem_01332]

Upstream requirements: RS_Diag_04194

⌈In case an event is partially cleared according to [SWS_Dem_00669], the dependent functionality shall be handled as if the event is not cleared at all. This means the callback function `InitMonitorForEvent` will not be called and the FDC will remain unchanged in this situation.⌋

[SWS_Dem_00516]

Upstream requirements: RS_Diag_04194

⌈If the return value of the callback `DemClearEventAllowed` is equal to E_NOT_OK, the return value shall not be changed and event-deletion is allowed.⌋

The `Dem` supports events that are not assigned to any `event memory`. This is the case, if the optional parameter `DemDTCRef` is not configured in `DemEventParameter`. Events are cleared, when the `event memory` assigned to them is cleared. In case there is no assigned `event memory`, these events would never be cleared. For this case the `Dem` provides the configuration parameter `DemClearEventsWithout-DTCEventMemoryRef`.

[SWS_Dem_01249] ⌈If the `event memory` referenced by `DemClearEvents-WithoutDTCEventMemoryRef` is cleared via any of the `Dem_ClearDTC` APIs, the `Dem` shall clear also all the events that are not assigned to any `event memory`.⌋

Note: Event displacement (mainly controlled by the event priority) and `aging` are not influenced by this callback.

7.7.2.3 Event memory overflow indication

[SWS_Dem_00397]

Upstream requirements: RS_Diag_04093

⌈The Dem module shall indicate for each `event memory` if the `event memory` is full and there was an attempt to store an additional event in this `event memory`.⌋

This overflow indication can be used to trigger further internal behavior of the Dem module (e.g. displacement strategy). Furthermore, it can be used for additional fault analysis in workshops.

[SWS_Dem_00398]

Upstream requirements: RS_Diag_04093

⌈The Dem module shall provide the API `Dem_GetEventMemoryOverflow` to provide access to the `event memory` overflow indication status of the respective `event memory`.⌋

Note: This API can be used for vendor-specific Diagnostic Services or other vendor-specific handling, and is provided to SW-Cs, as well as complex device drivers.

[SWS_Dem_00651]

Upstream requirements: RS_Diag_04109

⌈The Dem module shall provide the API `Dem_GetNumberOfEventMemoryEntries` to return the number of `event memory entries` currently stored in the `event memory`.⌋

Note: For the reported number all events are considered independently of the fault confirmation.

[SWS_Dem_00399]

Upstream requirements: RS_Diag_04093

⌈The `event memory` overflow indication of the respective `event memory` shall be reset, if all DTCs of this memory are deleted by `Dem_ClearDTC`.⌋

Note: In case of `aging` and deleting single DTCs, the overflow indication of the `event memory` is not reset to keep this status information until the ECU receives an explicit clear command of this specific `event memory`. The ECU itself should not change this status information during normal operation (because it is used in workshops).

7.7.2.4 Event displacement

Event displacement means, that the most insignificant, already existing `event memory entry` is replaced by a new `event memory entry`, which needs to be stored. During displacement, the most insignificant entry gets lost.

[SWS_Dem_00400]

Upstream requirements: RS_Diag_04118

⌈If the event retention want to allocate a new `event memory entry` and there is no free `event memory entry` available, the Dem module shall check according [SWS_Dem_00406] for allocated `event memory entries` to be displaced by the new `event memory entry`.⌋

Note: If the `event memory` size is configured to cover all possible events, no displacement will occur.

[SWS_Dem_00401]

Upstream requirements: RS_Diag_04118

⌈The Dem module provides the configuration parameter DemEventDisplacementStrategy (refer to `DemPrimaryMemory` or `DemUserDefinedMemory`) defining whether the existing `event memory entry` can be displaced or not.⌋

[SWS_Dem_00402]

Upstream requirements: RS_Diag_04118

⌈If event displacement is disabled (DemEventDisplacementStrategy in DemPrimaryMemory or DemUserDefinedMemory selects DEM_DISPLACEMENT_NONE), the Dem module shall not displace existing event memory entries if the event memory is full.⌋

[SWS_Dem_00406]

Upstream requirements: RS_Diag_04118

⌈If event displacement is enabled (DemEventDisplacementStrategy selects DEM_DISPLACEMENT_FULL in DemPrimaryMemory or DemUserDefinedMemory or DEM_DISPLACEMENT_PRIO_OCC), the Dem module shall perform the following sequence by combination of the different displacement criteria:

1. Priority (refer [SWS_Dem_00403])
2. Active/Passive status (configurable, only for DEM_DISPLACEMENT_FULL in DemPrimaryMemory or DemUserDefinedMemory) (refer [SWS_Dem_00404])
3. Occurrence (refer [SWS_Dem_00405])

⌋

Note: Priority needs to have Fuel system/Misfire over the rest (US/EU5) and Class A before B1,B2, C (Euro VI).

[SWS_Dem_00403]

Upstream requirements: RS_Diag_04118

⌈If displacement strategy is DEM_DISPLACEMENT_FULL or DEM_DISPLACEMENT_PRIO_OCC (refer to DemPrimaryMemory or DemUserDefinedMemory), the Dem module shall displace lower prioritized events by higher prioritized events.⌋

[SWS_Dem_00404]

Upstream requirements: RS_Diag_04118

⌈If displacement strategy is DEM_DISPLACEMENT_FULL(refer to DemPrimaryMemory or DemUserDefinedMemory), the Dem module shall displace passive events preferably.⌋

For details see chapter 7.7.1.4.

[SWS_Dem_00405]

Upstream requirements: RS_Diag_04118

⌈If displacement strategy is DEM_DISPLACEMENT_FULL or DEM_DISPLACEMENT_PRIO_OCC (refer to DemPrimaryMemory or DemUserDefinedMemory), the Dem module shall displace older stored events in the chronological order prior to newer stored events (refer [SWS_Dem_00412] and [SWS_Dem_00787]).⌋

[SWS_Dem_00407]

Upstream requirements: RS_Diag_04118

⌈If no event memory entry for displacement was identified, the Dem module shall discard the storage request.⌋

[SWS_Dem_00692]

Upstream requirements: RS_Diag_04118, RS_Diag_04071

⌈If an event is reported and an event memory entry with the same priority exists and the existing event is not tested in this operation cycle (TestNotCompletedThisOperationCycle bit == 1), the Dem shall displace the existing event with the reported event.⌋

[SWS_Dem_00693]

Upstream requirements: RS_Diag_04178

⌈During displacement, if an event is reported and an event memory entry with the same priority exists and the existing event is tested in this operation cycle (TestNotCompletedThisOperationCycle bit == 0), the Dem shall not displace the existing event with the reported event.⌋

[SWS_Dem_00695]

Upstream requirements: RS_Diag_04195

⌈If the configuration parameter DemOBDEventDisplacement is set to TRUE all emission related events with the following conditions shall not be considered during the displacement mechanism and shall not be displaced from error memory:
Event*old* is emission related
AND
(Event*old* triggers MIL indicator
OR (Event*old* holds legislative Freeze Frame **AND** has equal or higher priority)
OR Pending-DTC of Event*old*==1)⌋

[SWS_Dem_00696]

Upstream requirements: RS_Diag_04195

⌈If the configuration parameter DemOBDEventDisplacement is set to FALSE, no special rules of displacement are applied to emission related events. In this case emission related events shall be displaced like every other event.⌋

[SWS_Dem_00408]

Upstream requirements: RS_Diag_04118

⌈If the event memory entry for displacement was identified, the Dem module shall remove this event memory entry from the event memory.⌋

Note: This removed event memory entry should now used for the pending storage trigger.

[SWS_Dem_00409] Configurable reset of the confirmedDTC bit in case of displacement

Upstream requirements: RS_Diag_04067, RS_Diag_04118

⌈If an event memory entry was removed during displacement and the configuration parameter DemResetConfirmedBitOnOverflow is set to TRUE, the Dem module shall reset the UDS DTC status bit 3 to 0.⌋

[SWS_Dem_01395] Configurable reset of the pendingDTC bit in case of displacement

Upstream requirements: RS_Diag_04067, RS_Diag_04118

⌈If an event memory entry was removed during displacement and the configuration parameter DemResetPendingBitOnOverflow is set to TRUE, the Dem module shall reset the UDS DTC status bit 2 to 0.⌋

[SWS_Dem_01186]

Upstream requirements: RS_Diag_04067, RS_Diag_04118

⌈If an event memory entry was removed during displacement, the UDS status bit 5 shall be reset to 0 in case of DemStatusBitHandlingTestFailedSinceLastClear is set to DEM_STATUS_BIT_AGING_AND_DISPLACEMENT and DemResetConfirmedBitOnOverflow is set to true.⌋

7.7.2.5 Reporting order of event memory entries

[SWS_Dem_00410]

Upstream requirements: RS_Diag_04195

⌈The Dem module shall report DTCs in the chronological order of the event storage (refer to API Dem_GetNextFilteredDTC), if:

- the DTCStatus parameter has the“pending DTC” or “confirmed DTC” bit or both bits set
- all other bits of the DTCStatus parameter are set to false
- DemResetConfirmedBitOnOverflow is set to true

⌋

Note: The chronological order is for reporting purposes only and does not imply explicit memory structure, which is implementation specific.

Note: The reporting order may vary with the customer specific attributes used by the algorithm for sorting the DTC records in case of not confirmed or pending. If the chronological order is required for further UDS status mask parameters, additional resources may be necessary.

[SWS_Dem_00787]

Upstream requirements: RS_Diag_04195

⌈If an 'stored event' gets re-qualified as failed (UDS status bit 0 changes from 0 to 1) and a respective event memory entry exists, the Dem module shall update the chronological order of the event storage by setting the particular event as most recent event memory entry.⌋

[SWS_Dem_00411]

Upstream requirements: RS_Diag_04195

⌈If the Dem module is requested to report in chronological order, the most recent event memory entry shall be reported at first.⌋

[SWS_Dem_00412]

Upstream requirements: RS_Diag_04195

⌈If a new event get stored in the event memory, the Dem shall update the chronological order by setting the new event as most recent event memory entry.⌋

Note: If there are multiple event memory entries stored at the same point in time the chronological reporting order of these elements is undefined. It depends on the implementation which of these entries is reported or displaced first..

7.7.3 Debouncing of diagnostic events

In general, an ECU may implement several types of debouncing improving signal quality. This section describes methods, how the Dem module shall implement basic algorithms used for fault maturation (diagnostic event debouncing).

If the Dem module is configured to implement the debounce algorithm for a specific event, one of the following debounce algorithms are to be performed Dem-internally. Otherwise, the respective monitor implemented in a SW-C or BSW will report the status of a certain event, after diagnostic event debouncing has been completed. For interaction between the Dem and SW-C or BSW please refer to the API InitMonitorForEvent (refer to chapter 7.2).

If there are any requirements to get the passed or failed status within a certain amount of time, the diagnostic monitor is responsible.

[SWS_Dem_00413]

Upstream requirements: RS_Diag_04068

⌈The Dem module shall support the event-specific configuration of debounce algorithms by using the configuration container DemDebounceAlgorithmClass.⌋

Note: That means for each individual event a specific algorithm can be specified.

7.7.3.1 Counter based debounce algorithm

[SWS_Dem_00526]

Upstream requirements: RS_Diag_04068

⌈The Dem module shall provide a configuration parameter DemDebounceCounterBasedSupport (refer to DemGeneral) to enable or disable the Dem-internal counter based debouncing.⌋

[SWS_Dem_00414]

Upstream requirements: RS_Diag_04068

⌈If the configuration container DemDebounceAlgorithmClass is set to DemDebounceCounterBased, the Dem module shall provide an internal debounce counter for each individual event, to qualify the reported event.⌋

For huge debounce ranges, the maximal range of the internal debounce counter is defined as sint16. This does not imply any explicit implementation.

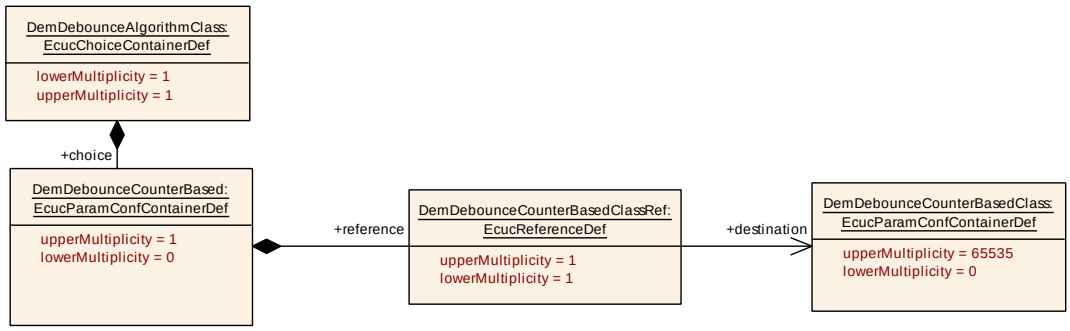


Figure 7.28: Counter based debounce algorithm

[SWS_Dem_00415] ⌈The Dem module shall calculate the fault detection counter (-128 ... +127 according to UDS) based on the value and range of the internal debounce counter to map the internal counter values linearly to the external values.⌋

[SWS_Dem_00416]

Upstream requirements: RS_Diag_04068

⌈The Dem module shall provide the configuration parameter DemDebounceCounterFailedThreshold used to define the event-specific limit indicating the failed status (active).⌋

[SWS_Dem_00417]

Upstream requirements: RS_Diag_04068

⌈The Dem module shall provide the configuration parameter DemDebounceCounterPassedThreshold used to define the event-specific limit indicating the passed status (passive).⌋

[SWS_Dem_00418]

Upstream requirements: RS_Diag_04068

⌈The Dem module shall increment the internal debounce counter with its configured step-size (refer to DemDebounceCounterIncrementStepSize), when the monitor reports DEM_EVENT_STATUS_PREFAILED. If after the incrementation the debounce counter value is greater than the threshold DemDebounceCounterFailedThreshold, the Dem shall set the debounce counter value equal to the threshold.⌋

[SWS_Dem_00419]

Upstream requirements: RS_Diag_04068

⌈The Dem module shall decrement the internal debounce counter with its configured step-size (refer to DemDebounceCounterDecrementStepSize), when the monitor reports DEM_EVENT_STATUS_PREPASSED. If after the decrementation the debounce counter value is lower than the threshold DemDebounceCounterPassedThreshold, the Dem shall set the debounce counter value equal to the threshold.⌋

[SWS_Dem_00420]

Upstream requirements: RS_Diag_04068

⌈If the monitor reports DEM_EVENT_STATUS_FAILED, the Dem module shall set the internal debounce counter value to its configured threshold being the failed criteria.⌋

[SWS_Dem_00421]

Upstream requirements: RS_Diag_04067, RS_Diag_04068

⌈If the monitor reports DEM_EVENT_STATUS_PASSED, the Dem module shall set the internal debounce counter value to its configured threshold being the passed criteria.⌋

[SWS_Dem_00422]

Upstream requirements: RS_Diag_04068

⌈The Dem module shall provide the configuration parameter DemDebounceCounterJumpDown for enabling the jump-down behavior.⌋

[SWS_Dem_00423]

Upstream requirements: RS_Diag_04068

⌈If the jump-down behavior is enabled, the Dem module shall provide the configuration parameter DemDebounceCounterJumpDownValue defining the new internal debounce counter init value. Each reporting of a pre-passed value while the current debounce counter value is greater than the DemDebounceCounterJumpDownValue shall first reset the debounce counter to DemDebounceCounterJumpDownValue before performing the pre-passed debounce event ([SWS_Dem_00419]).⌋

Note: This will only happen if the counting direction changes from incrementing to decrementing.

[SWS_Dem_00424]

Upstream requirements: RS_Diag_04068

⌈The Dem module shall provide the configuration parameter DemDebounceCounterJumpUp for enabling the jump-up behavior.⌋

[SWS_Dem_00425]

Upstream requirements: RS_Diag_04068

⌈If the jump-up behavior is enabled, the Dem module shall provide the configuration parameter DemDebounceCounterJumpUpValue defining the new internal debounce counter init value. Each reporting of a pre-failed value while the current debounce counter value is smaller than the DemDebounceCounterJumpUpValue shall first reset the debounce counter to DemDebounceCounterJumpUpValue before performing the pre-failed debounce event ([SWS_Dem_00418]).⌋

Note: This will only happen if the counting direction changes from decrementing to incrementing.

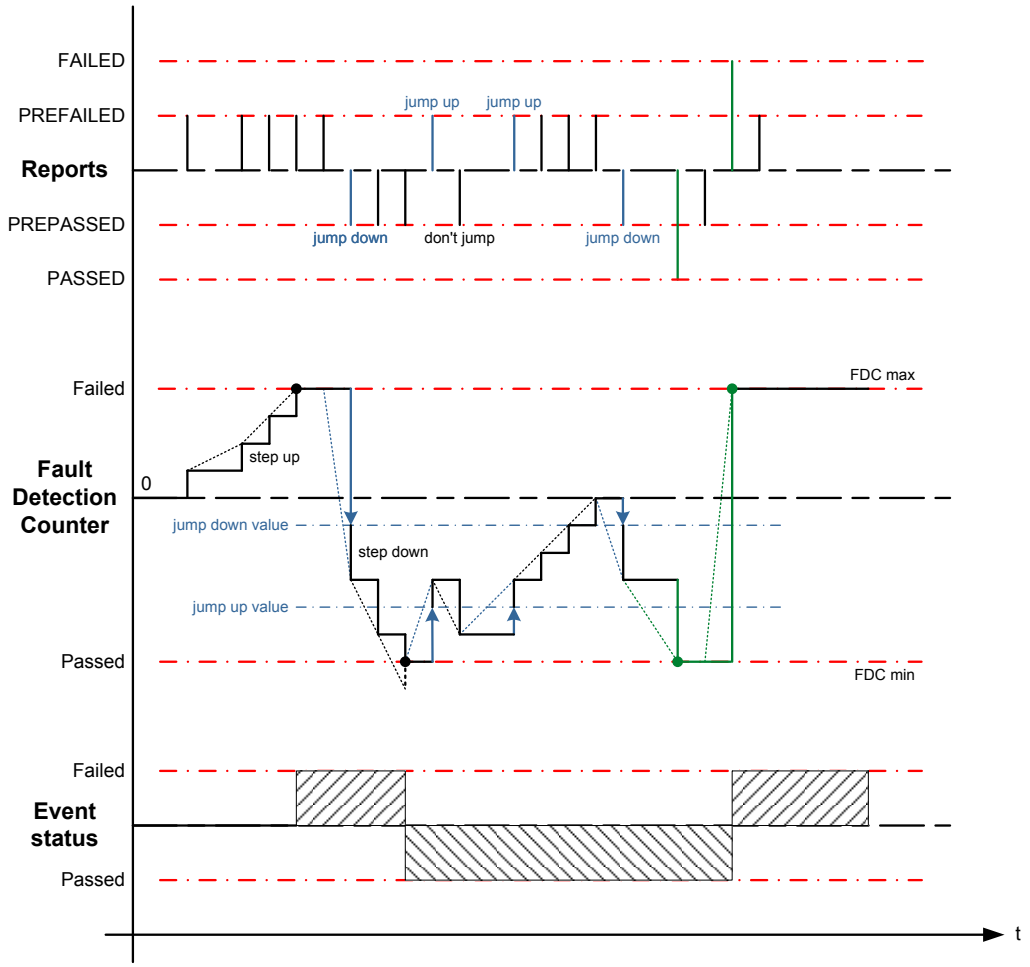


Figure 7.29: Example of counter based debouncing (including jump behaviour)

Note: In figure 7.29, the value of Fault Detection Counter maps linearly to the internal debounce counter value.

The action DEM_DEBOUNCE_STATUS_FREEZE of the API Dem_ResetEvent-DebounceStatus is not relevant for counter-base debouncing. The action DEM_DEBOUNCE_STATUS_RESET of the Dem_ResetEventDebounceStatus is covered by [SWS_Dem_00684].

The DemDebounceBehavior DEM_DEBOUNCE_STATUS_FREEZE is covered in chapter 7.7.6. The DemDebounceBehavior DEM_DEBOUNCE_RESET is covered by [SWS_Dem_00654] and [SWS_Dem_00677]. This is shown in figure 7.30.

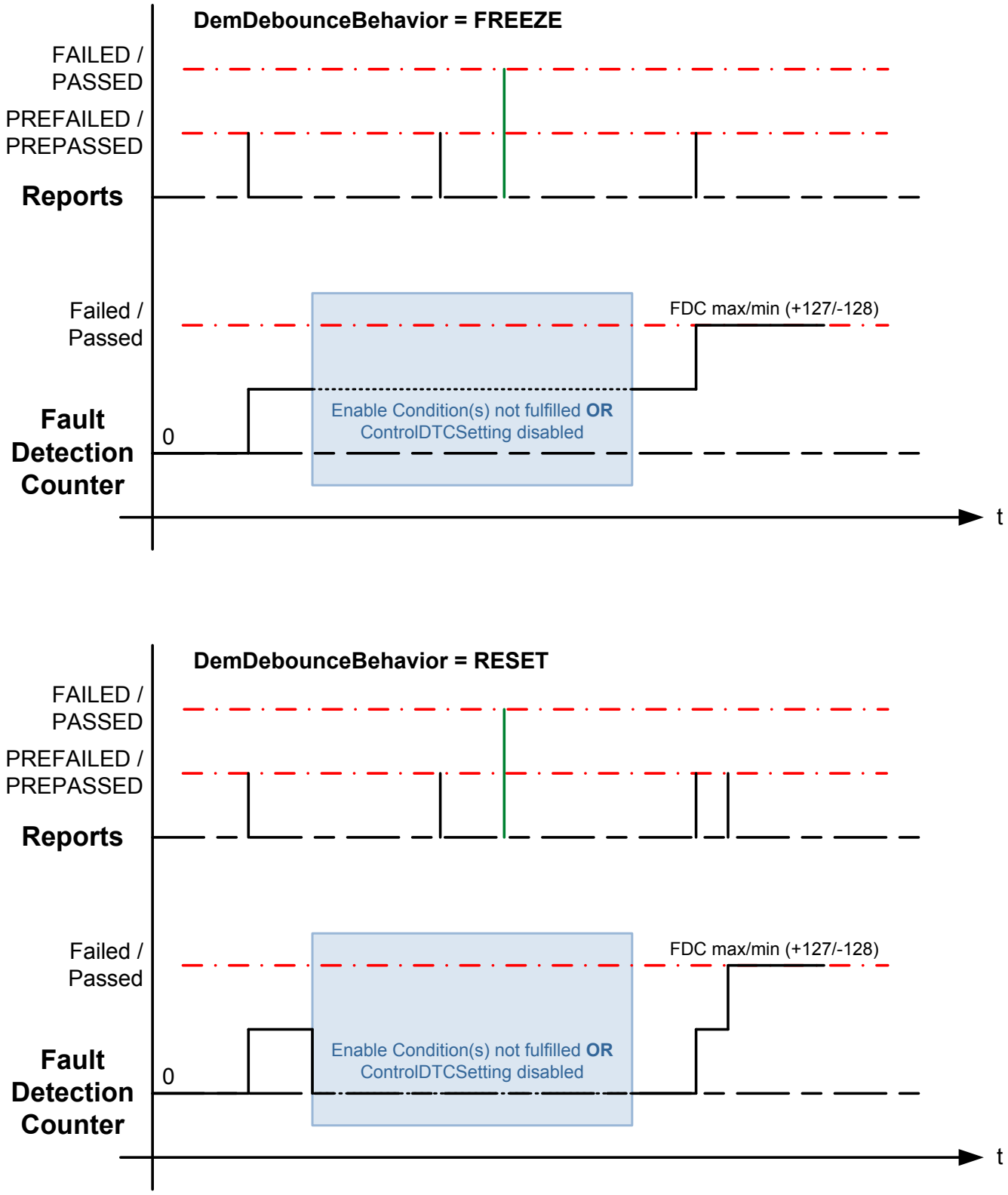


Figure 7.30: Counter based debouncing behavior on enable conditions

[SWS_Dem_00674]

Upstream requirements: RS_Diag_04124

⌈If the configuration parameter DemDebounceCounterStorage is set to True, the Dem module shall store the current value of the internal debounce counter non-volatile (refer to [SWS_Dem_00341]).⌋

Note: DemDebounceCounterStorage is not supported for time based debouncing.

[SWS_Dem_00675]

Upstream requirements: RS_Diag_04124

⌈If the configuration parameter DemDebounceCounterStorage is set to True, the Dem module shall re-store the current value of the internal debounce counter during each startup (latest in Dem_Init).⌋

Note: BSW events using non-volatile storage of the internal debounce counter cannot be reported before Dem_Init, because the value needs first to be restored.

[SWS_Dem_00676]

Upstream requirements: RS_Diag_04124

⌈If development error detection is enabled and Dem_SetEventStatus or Dem_ResetEventDebounceStatus is called before Dem_Init or after Dem_Shutdown for events which have DemDebounceCounterStorage set to True, the Dem module shall set the error code DEM_E_WRONG_CONDITION.⌋

[SWS_Dem_CONSTR_06151] Dependency on DemCounterBasedFdcThresholdStorageValue ⌈The configuration parameter DemCounterBasedFdcThresholdStorageValue shall only be present if DemFreezeFrameRecordTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD or DemExtendedDataRecordTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD or DemEventMemoryEntryStorageTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD.⌋

[SWS_Dem_CONSTR_06152] Dependency on DemDebounceCounterJumpDownValue ⌈DemDebounceCounterJumpDownValue shall only be present if DemDebounceCounterJumpDown is set to TRUE.⌋

[SWS_Dem_CONSTR_06153] Dependency on DemDebounceCounterJumpUpValue ⌈DemDebounceCounterJumpUpValue shall only be present if DemDebounceCounterJumpUp is set to TRUE.⌋

7.7.3.2 Time based debounce algorithm

[SWS_Dem_00527]

Upstream requirements: RS_Diag_04068

⌈The Dem module shall provide a configuration parameter `DemDebounceTimeBasedSupport` (refer to `DemGeneral`) to enable or disable the Dem-internal time based debouncing.⌋

[SWS_Dem_00426]

Upstream requirements: RS_Diag_04068

⌈If the configuration container `DemDebounceAlgorithmClass` is set to DemDebounceTimeBased, the Dem module shall provide an internal debounce timer for each individual event, to qualify the reported event.⌋

For long debounce periods, the maximal range of the internal debounce timer is defined as sint16. This does not imply any explicit implementation.

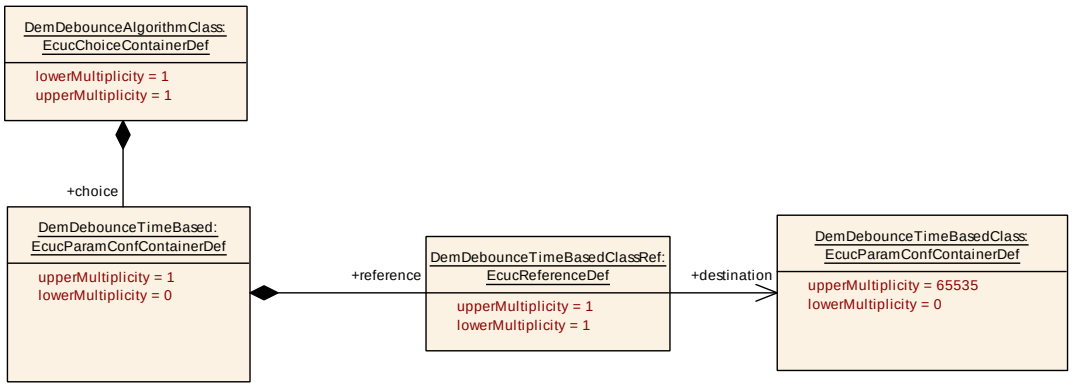


Figure 7.31: Time based debounce algorithm

[SWS_Dem_00427]

Upstream requirements: RS_Diag_04068

⌈The Dem module shall calculate the fault detection counter (-128 ... +127 according to UDS) based on the value and range of the internal debounce timer, to map the internal timer values linearly to the external values.⌋

For details see figure 7.32.

[SWS_Dem_00428]

Upstream requirements: RS_Diag_04068

⌈The `Dem` module shall start the internal debounce timer to qualify the reported event as failed when the monitor reports `DEM_EVENT_STATUS_PREFAILED`.⌋

[SWS_Dem_00429]

Upstream requirements: RS_Diag_04068

⌈If the internal debounce timer of a specific event was already triggered as pre-failed or the event is qualified as failed, and the monitor reports consecutively DEM_EVENT_STATUS_PREFAILED again, the Dem module shall not restart the internal debounce timer.⌋

Note: The Dem module provides the configuration parameter DemDebounceTimeFailedThreshold used to define the event-specific delay indicating the failed status (active).

[SWS_Dem_00431]

Upstream requirements: RS_Diag_04067, RS_Diag_04068

⌈If the monitor reports DEM_EVENT_STATUS_FAILED, the Dem module shall set the internal debounce timer value to its configured threshold being the failed criteria (refer to DemDebounceTimeFailedThreshold).⌋

[SWS_Dem_00432]

Upstream requirements: RS_Diag_04068

⌈The Dem module shall start the internal debounce timer to qualify the reported event as passed when the monitor reports DEM_EVENT_STATUS_PREPASSED.⌋

[SWS_Dem_00433]

Upstream requirements: RS_Diag_04068

⌈If the internal debounce timer of a specific event was already triggered as pre-passed or the event is qualified as passed, and the monitor reports consecutively DEM_EVENT_STATUS_PREPASSED again, the Dem module shall not restart the internal debounce timer.⌋

[SWS_Dem_00434]

Upstream requirements: RS_Diag_04068

⌈The Dem module shall provide the configuration parameter DemDebounceTimePassedThreshold used to define the event-specific delay indicating the passed status (not active).⌋

[SWS_Dem_00435]

Upstream requirements: RS_Diag_04068

⌈If the monitor reports DEM_EVENT_STATUS_PASSED the Dem module shall set the internal debounce timer value to its configured threshold being the passed criteria (refer to DemDebounceTimePassedThreshold).⌋

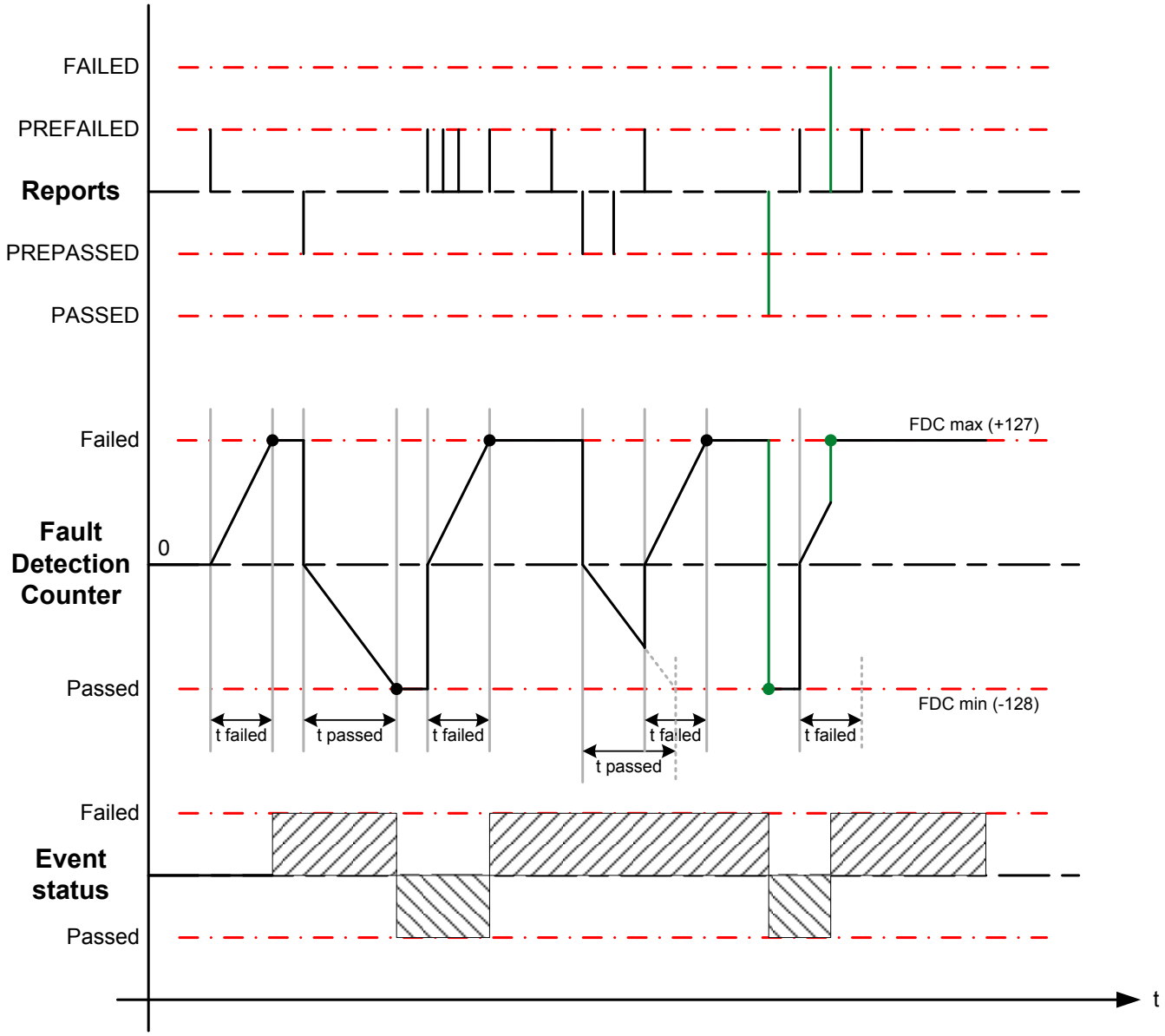


Figure 7.32: Example of time based debouncing

[SWS_Dem_00685]

Upstream requirements: RS_Diag_04068

⌈If the API `Dem_ResetEventDebounceStatus` is called with `DEM_DEBOUNCE_STATUS_FREEZE`, it shall freeze the related internal debounce timer.⌋

The action `DEM_DEBOUNCE_STATUS_RESET` of the API `Dem_ResetEventDebounceStatus` is covered by [SWS_Dem_00684].

[SWS_Dem_00655]

Upstream requirements: RS_Diag_04125

⌈If the configuration parameter `DemDebounceBehavior` is set to `DEM_DEBOUNCE_STATUS_FREEZE`, the Dem module shall freeze the internal debounce timer when at least one enable condition for the related event is set to not fulfilled.⌋

[SWS_Dem_00678]

Upstream requirements: RS_Diag_04125

⌈If the configuration parameter `DemDebounceBehavior` is set to `DEM_DEBOUNCE_STATUS_FREEZE`, the Dem module shall freeze the internal debounce timer when ControlDTCSetting is set to disabled for the related event.⌋

[SWS_Dem_00656]

Upstream requirements: RS_Diag_04125

⌈If an internal debounce timer is frozen and a new (valid) (debouncing-)result is reported for this event (as per [SWS_Dem_00428] / [SWS_Dem_00432]), the Dem module shall continue running the internal debounce timer (from current value).⌋

The `DemDebounceBehavior DEM_DEBOUNCE_RESET` is covered by [SWS_Dem_00654] and [SWS_Dem_00677].

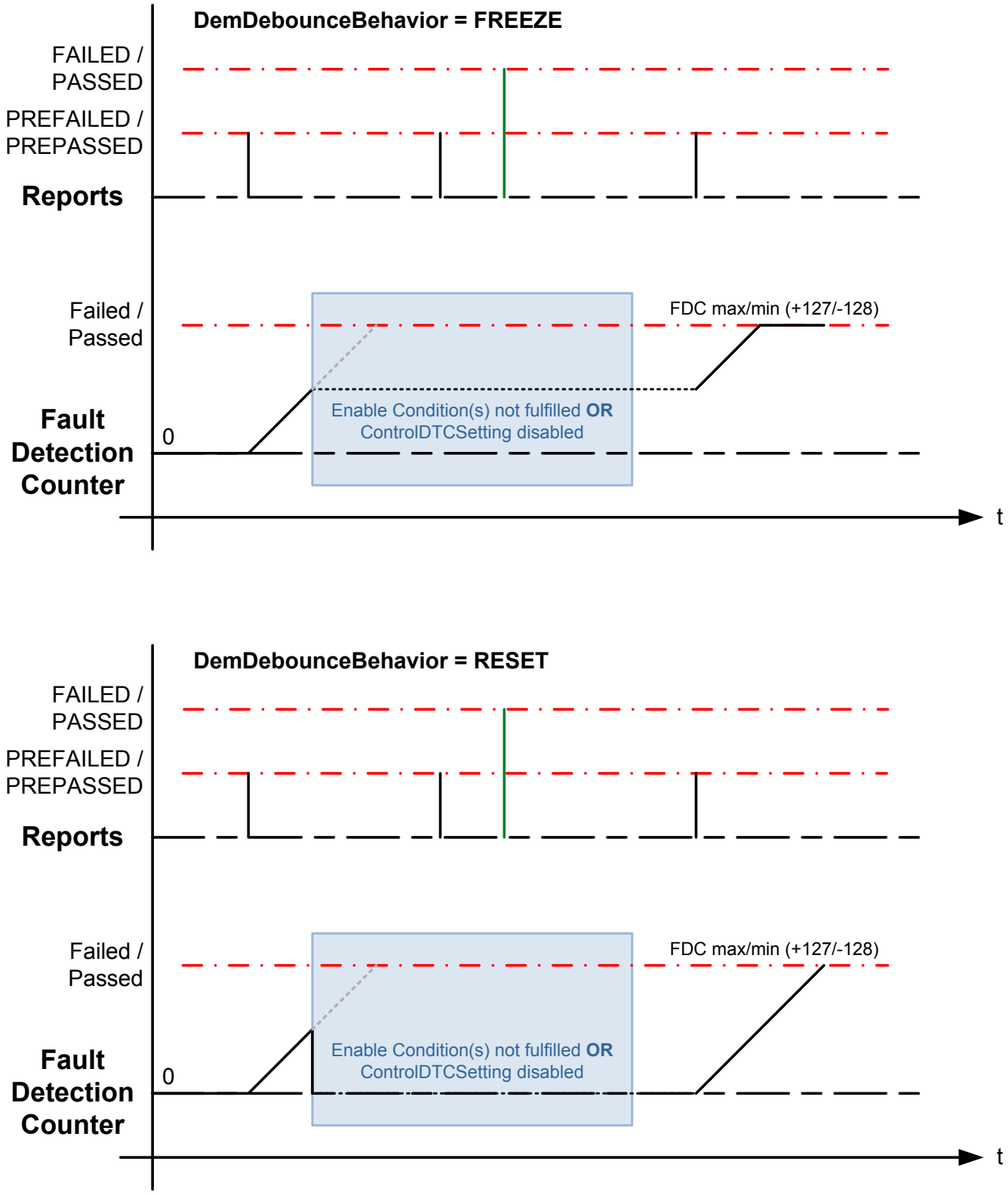


Figure 7.33: Time based debouncing behavior on enable conditions

7.7.3.3 Monitor internal debounce algorithm

Monitor internal debouncing is used, if the event does not need debouncing or requires a more complex debouncing that is not covered by counter or time based debouncing. In that case the monitor reports only qualified results to the Dem and DEM_EVENT_STATUS_PREFAILED and DEM_EVENT_STATUS_PREPASSED cannot be used (eg. Dem_SetEventStatusWithMonitorData, Dem_SetEventStatus...) in a call of Dem_SetEventStatus.

[SWS_Dem_01408] Behavior on invalid use of pre qualified results

Upstream requirements: RS_Diag_04068

⌈If the configuration container DemDebounceAlgorithmClass is set to DemDebounceMonitorInternal for an event and any API reporting an event status by Dem_EventStatusType is called with DEM_EVENT_STATUS_PREFAILED or DEM_EVENT_STATUS_PREPASSED for that event, the Dem shall report the DET error DEM_E_WRONG_CONFIGURATION.⌋

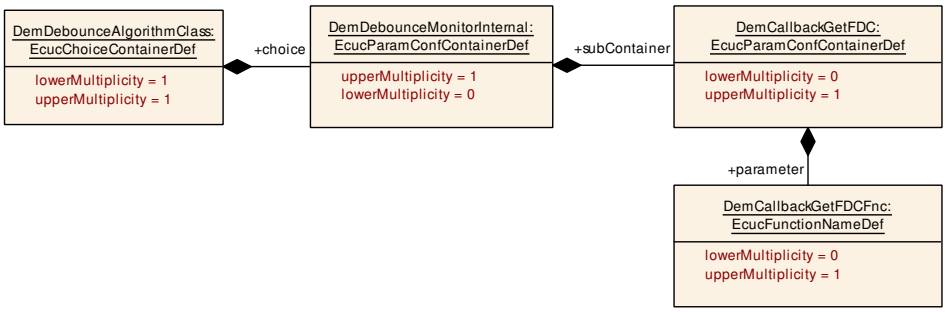


Figure 7.34: Monitor internal debounce algorithm

7.7.3.4 Debounce algorithm initialization and reset conditions

[SWS_Dem_00438]

Upstream requirements: RS_Diag_04068

⌈If Dem-internal debouncing is configured, the Dem module shall reset the Dem-internal debounce algorithm when Dem_PreInit has been called.⌋

[SWS_Dem_00343]

Upstream requirements: RS_Diag_04117, RS_Diag_04068

⌈After receiving a command for clearing the event memory, the Dem-internal debounce algorithm shall be reset, presuming event debouncing is handled Dem-internally.⌋

For details see chapter 7.7.2.2.

[SWS_Dem_00684]

Upstream requirements: RS_Diag_04068

⌈If the API `Dem_ResetEventDebounceStatus` is called with `DEM_DEBOUNCE_STATUS_RESET`, it shall reset and freeze the related fault detection counter until the status of the event is reported again.⌋

[SWS_Dem_00344]

Upstream requirements: RS_Diag_04068

⌈If Dem-internal debouncing is configured, the Dem module shall reset the Dem-internal debounce algorithm upon starting a new operation cycle.⌋

For details see chapter 7.6.

Note: Resetting the debounce algorithm will also lead to a fault detection counter value of 0 (zero).

[SWS_Dem_00654]

Upstream requirements: RS_Diag_04125

⌈If the configuration parameter `DemDebounceBehavior` is set to `DEM_DEBOUNCE_RESET` and the function `Dem_SetEnableCondition` sets one configured enable condition for the event to not fulfilled, it shall reset the related fault detection counter(s).⌋

For details see chapter 7.7.6.

[SWS_Dem_00677]

Upstream requirements: RS_Diag_04125

⌈If the configuration parameter `DemDebounceBehavior` is set to `DEM_DEBOUNCE_RESET`, the function `Dem_DisableDTCSetting` shall reset the related fault detection counter(s).⌋

For details see chapter 7.11.2.6.

7.7.3.5 Fault detection counter retrieval

[SWS_Dem_00204]

Upstream requirements: RS_Diag_04068

⌈The event-specific fault detection counter shall be accessible by using the API `Dem_GetFaultDetectionCounter`.⌋

The fault detection counter can be:

1. located inside the Dem, if Dem-internal debouncing is configured (refer to chapter 7.7.3.1 and chapter 7.7.3.2).

2. located inside the monitor, if debouncing is performed by the monitor (refer to [SWS_Dem_00264], [SWS_Dem_00439], [SWS_Dem_00513]).

[SWS_Dem_00264]

Upstream requirements: RS_Diag_04068

⌈If debouncing is performed by a SW-C (not handled Dem internally), the Dem module shall retrieve the current value of the fault detection counter for the requested event by using the method defined in DemCallbackGetFDC container.⌋

Note: The configuration container DemCallbackGetFDC (in DemDebounceMonitorInternal) is used to specify the related port or c-callback per event.

[SWS_Dem_00439]

Upstream requirements: RS_Diag_04068

⌈If monitor internal debouncing is used and the container DemCallbackGetFDC is configured and the configured callback-function returns other than E_OK, this return value shall also be returned by the API Dem_GetFaultDetectionCounter.⌋

Note: For resetting the fault detection counter implemented in a monitor, the Dem module uses the callback-function InitMonitorForEvent (refer to chapter 7.2).

[SWS_Dem_00671] ⌈If monitor internal debouncing is used and the container DemCallbackGetFDC is not configured for a given event, the API Dem_GetFaultDetectionCounter shall return DEM_E_NO_FDC_AVAILABLE.⌋

7.7.3.6 Fault detection counter reporting

The maximum value the DTC Fault Detection Counter has reached during current operation cycle.

[SWS_Dem_00788]

Upstream requirements: RS_Diag_04125

⌈If the maximum FDC during current operation cycle is mapped to an extended data record (DemInternalDataElement set to DEM_MAX_FDC_DURING_CURRENT_CYCLE), it shall be available per 'event related data' record.⌋

[SWS_Dem_00790]

Upstream requirements: RS_Diag_04125

⌈The maximum FDC during current operation cycle shall be reset to zero with each restart operation cycle (refer to DemOperationCycleRef).⌋

[SWS_Dem_00791]

Upstream requirements: RS_Diag_04125

⌈The maximum FDC during current operation cycle shall be update if the current fault detection counter value is greater than the current value of the maximum FDC during current operation cycle.⌋

The maximum value the DTC Fault Detection Counter has reached since the last time DTC information was cleared.

[SWS_Dem_00792]

Upstream requirements: RS_Diag_04125

⌈If the maximum FDC since last clear is mapped to an extended data record (DemInternalDataElement set to DEM_MAX_FDC_SINCE_LAST_CLEAR), it shall be available per 'event related data' record.⌋

[SWS_Dem_00794]

Upstream requirements: RS_Diag_04125

⌈The maximum FDC since last clear shall be reset to zero with each clear DTC command affecting this particular event.⌋

[SWS_Dem_00795]

Upstream requirements: RS_Diag_04125

⌈The maximum FDC since last clear shall be update if the current fault detection counter value is greater than the current value of the maximum FDC since last clear.⌋

The Dem module provides various counters with fault detection counter (FDC) information as internal data elements. If requested per DTC, the question arises for which of the combined DTCs the FDC is requested. For a requester of the FDC, the highest value is most interesting as it indicates how close an event is to fail. The Dem will always choose the highest values in this case.

[SWS_Dem_00672] ⌈If the Dem is requested to report the value of a fault detection counter with Dem_GetNextFilteredDTCAndFDC or any related internal data element (DEM_CURRENT_FDC, DEM_MAX_FDC_DURING_CURRENT_CYCLE, DEM_MAX_FDC_SINCE_LAST_CLEAR) for a combined DTC, the Dem shall report the highest respective value of the sub-events.⌋

Note: The combined fault detection counter value is required for Dem_GetNextFilteredDTCAndFDC (in UDS Service 0x19 14).

7.7.4 Fault confirmation

After reporting, a fault and entering the pending status, the fault confirmation process begins within the Dem (refer to Figure 7.26 General diagnostic event storage processing).

This fault confirmation process results in the confirmed state. For that purpose, respective counter thresholds and counter types are specified.

[SWS_Dem_00528] ⌈The Dem module shall provide the configuration parameter DemEventConfirmationThreshold per event defining the maximum number of tested and failed cycles, before the event becomes “confirmed”, i.e. enters the confirmed state.⌋

[SWS_Dem_00529]

Upstream requirements: RS_Diag_04067

⌈The configuration parameter DemOperationCycleRef (refer to DemEventParameter) shall be used for calculating the trip counter according ISO14229-1[2].⌋

For details see figure 7.19.

The healing confirmation process (described in chapter 7.7.8) handles reported OK / passed results, which yields in the deactivation of a specific warning indicator. For that purpose, respective counter thresholds and counter types are specified.

Note: If the fault confirmation of an event is disabled, its UDS status bit 3 is set during the qualification to failed (according to [SWS_Dem_00391]).

[SWS_Dem_01242]

Upstream requirements: RS_Diag_04161

⌈The Dem shall support calls of Dem_SetEventConfirmationThresholdCounter after Dem_PreInit.⌋

[SWS_Dem_01243]

Upstream requirements: RS_Diag_04161

⌈A call of the API Dem_SetEventConfirmationThresholdCounter shall set a new failure confirmation threshold.⌋

Note: A new failure confirmation threshold value will overwrite the default value given by DemEventConfirmationThreshold).

[SWS_Dem_01244]

Upstream requirements: RS_Diag_04161

⌈If Dem_SetEventConfirmationThresholdCounter is called and the corresponding event is already stored in event memory and the new failure threshold is less or equal to the current failure counter, then the event will get confirmed. This

implies, that the event related data will also be updated according their defined triggers.⌋

[SWS_Dem_01245]

Upstream requirements: RS_Diag_04161

⌈If Dem_SetEventConfirmationThresholdCounter is called and the corresponding event is already stored in event memory and already confirmed, the event memory entry and confirmation status will remain, even if the new threshold is greater than the failure counter. Also after new error status reports, the confirmation status will remain.⌋

[SWS_Dem_01246]

Upstream requirements: RS_Diag_04161

⌈A DemIndicatorAttribute with a configured DemMILIndicatorRef, shall use the DemEventConfirmationThreshold instead of the DemIndicatorFailureCycleCounterThreshold to calculate the indicator On Criteria (WarningIndicatorOnCriteriaFulfilled).⌋

7.7.4.1 Method for grouping of association of events for OBD purpose

[SWS_Dem_00965] ⌈The Dem module shall provide a configuration parameter DemOBDGroupingAssociativeEventsRef for each event to reference a representive event to one group of associate events.⌋

Note: One event is only allowed to be referenced to only one group of associate events. A referenced event must refer to it self.

[SWS_Dem_00967] ⌈For each report of a Malfunction within in a group of events Dem shall proceed cycle counter towards MIL activation, either by processing a common cycle counter or by counting the respective counter of the reported event referenceing to the event with the counter value “closest to MIL ON”.⌋

[SWS_Dem_00968]

Upstream requirements: RS_Diag_04178

⌈The healing / aging for each event shall to be performed individually.⌋

7.7.5 Event Combination

Event combination defines the ability of the Dem module to merge several events to one DTC. It is used to adapt different monitor results to one significant fault, which is clearly evaluable in a service station. The essential part of implementing a combined DTC is the calculation of its status information. The combined DTC status byte results from a bitwise logical operation of all associated events.

[SWS_Dem_00536] Support event combination

Upstream requirements: RS_Diag_04200

⌈The following event combination types are supported:

1. Combination on storage: configuration parameter DemEventCombination-Support is set to DEM_EVCOMB_ONSTORAGE. The combined DTC is stored and updated in a single event memory entry.
2. Disabled: configuration parameter DemEventCombinationSupport is set to DEM_EVCOMB_DISABLED and event combination is not used

⌋

[SWS_Dem_00024] ⌈In case DemEventCombinationSupport is set to DEM_EVCOMB_ONSTORAGE, the combination of multiple events to a DTC is defined by referencing from each event to the same DTC.⌋

Note: For OBD it is also possible to assign multiple time the same DTC number via Dem-ObdDTCRef. But as a difference to UDS and J1939 DTCs the OBD DTCs are reported multiple times to the Dcm.

Note: The primary use-case of event combination is to map more than one event (represented by an EventId) to only one DTC number. For combination on storage, the Dem module uses one event memory entry and its related data for all events combined to the DTC. The Dem module handles the combined DTC in consideration of its related event memory entry.

Table 7.2 shows an example of a Dem configuration table including a combined DTC. Event[1], Event[2], and Event[3] are mapped to DTC[1] using combination on storage.

Unique EventId	*Monitor status*	*UDS status*	*Assigned DTC*	*Freeze frame*	*....*
Event[1]	S1	S1|S2|S3	DTC[1]	FF[28]	
Event[2]	S2	S1|S2|S3	DTC[1]	FF[28]	
Event[3]	S3	S1|S2|S3	DTC[1]	FF[28]	
Event[4]	S4	S4	DTC[2]	FF[89]	
Event[5]	S5	S5	DTC[3]	FF[67]	
....	....	....	....	....	....

Table 7.2: Example of a Dem configuration table including a combined DTC

[SWS_Dem_00441] ⌈The Dem module shall implement the status bit calculations for the UDS status byte according to Table 7.3.⌋

UDS status bit description		*Combined UDS status information logical equation*
0	TestFailed	$CbDTC_{Bit0}$ = $Event[1]_{Bit0}$ | $Event[2]_{Bit0}$ | ... |$Event[n]_{Bit0}$
1	TestFailedThis OperationCy-cle	$CbDTC_{Bit1}$ = $Event[1]_{Bit1}$ | $Event[2]_{Bit1}$ | ... |$Event[n]_{Bit1}$

2	PendingDTC	CbDTC$_{Bit2}$ = Event[1]$_{Bit2}$ | Event[2]$_{Bit2}$ | ... |Event[n]$_{Bit2}$
3	ConfirmedDTC	CbDTC$_{Bit3}$ = Event[1]$_{Bit3}$ | Event[2]$_{Bit3}$ | ... |Event[n]$_{Bit3}$
4	TestNotCompleted SinceLastClear	CbDTC$_{Bit4}$ = (Event[1]$_{Bit4}$ | Event[2]$_{Bit4}$ | ... |Event[n]$_{Bit4}$) & ! CbDTC$_{Bit5}$
5	TestFailedSince LastClear	CbDTC$_{Bit5}$ = Event[1]$_{Bit5}$ | Event[2]$_{Bit5}$ | ... |Event[n]$_{Bit5}$
6	TestNotCompleted ThisOpera-tionCycle	CbDTC$_{Bit6}$ = (Event[1]$_{Bit6}$ | Event[2]$_{Bit6}$ | ... |Event[n]$_{Bit6}$) & ! CbDTC$_{Bit1}$CbDTC
7	WarningIndicator Requested	CbDTC$_{Bit7}$ = Event[1]$_{Bit7}$ | Event[2]$_{Bit7}$ | ... |Event[n]$_{Bit7}$

Table 7.3: Calculation of UDS status byte

In the table above the following logical operators are used:

- ! = logical negation (NOT)
- | = logical bitwise OR-operation
- & logical bitwise AND-operation

[SWS_Dem_00440] ⌈If the Dem module is requested to clear a `combined DTC` (refer to `Dem_ClearDTC`), the Dem module shall clear all related events (refer to [SWS_Dem_01049] and [SWS_Dem_01050])⌋

[SWS_Dem_01049] ⌈If the `Dem` module is requested to report the status of a `combined DTC` (refer e.g. `Dem_GetStatusOfDTC` or `Dem_GetNextFilteredDTC`), the calculation of the status byte shall be performed according [SWS_Dem_00441].⌋

[SWS_Dem_01050] ⌈Each time the status of an event is updated, the `combined DTC` status shall be calculated. If the `combined DTC` status has changed the relevant callbacks shall be invoked; refer to [SWS_Dem_00284], [SWS_Dem_00986] and [SWS_Dem_00987].⌋

[SWS_Dem_CONSTR_06103] ⌈In case the `event combination` is disabled, it is not allowed to reference from multiple events to the same dtc.⌋

7.7.5.1 Combination On Storage

The following section describes the behavior of the `Dem` module in case the combination on storage is configured.

[SWS_Dem_00163] ⌈If the Dem module is requested to support combination on storage, the `DTC` status bit transitions of the `combined DTC` (refer [SWS_Dem_00441]) shall be used as trigger source for the allocation of the `event memory entry`, as well as the collection or update of its related data (freeze frames or extended data records).⌋

[SWS_Dem_00442]

Upstream requirements: RS_Diag_04133

⌈If a `combined DTC` (combination on storage) is aged, the Dem module shall remove this `event memory entry` and reset the status bytes of all sub-events according to [SWS_Dem_00823], [SWS_Dem_00824] and [SWS_Dem_00498].⌋

[SWS_Dem_00443]

Upstream requirements: RS_Diag_04118

⌈If a `combined DTC` (combination on storage) is displaced, the Dem module shall remove this `event memory entry` and reset the status bytes of all sub-events according to [SWS_Dem_00409] and [SWS_Dem_01186].⌋

7.7.6 Enable and storage conditions of diagnostic events

In certain cases, the event retention depends on parameters, which are available on operation system level. These parameters are combined to groups and define a certain number of checks (e.g. correct voltage range) before the event report is accepted or the event gets qualified. The checks are done by software components. The Dem module provides the ability to consider the reported result (a specific condition is fulfilled or not) during the event handling. There are two different types of conditions: enable conditions and storage conditions.

The strategy how to use the conditions (e.g. suppression of subsequent faults) and especially the assigning matrix of conditions to the events depends on the OEM.

The enable conditions are defined as a set of parameters, which are assigned to a specific condition. As long as this condition is not fulfilled, the event reports (refer to `Dem_SetEventStatus`) are not valid and therefore will not be accepted. It has no impact on `Dem_ResetEventDebounceStatus`, `Dem_ResetEventStatus` and `Dem_ClearDTC`. A similar functionality is used for the function inhibition. In contrast to the mutual exclusion matrix of the FiM, which is based on events, the enable conditions are based on system parameters (e.g. ignition status, local voltage).

The storage conditions are defined as a set of parameters, which are assigned to a specific condition. As long as this condition is not fulfilled, the Dem module does not store the event to the `event memory`.

The following requirements introduce the handling of the enable conditions.

Note:The `Dem` module supports enable conditions and provides the API `Dem_SetEnableCondition` to receive the current status (condition fulfilled or not)of a specific enable condition.

Note: The `Dem` module supports enable conditions that can be assigned to a specific event.

Note: The enable condition groups are introduced to improve the configuration methodology of the Dem module. If a significant number of events depend always on the same enable conditions, it is less effort to select one of the defined enable condition groups instead of assigning several enable conditions to each of those events.

[SWS_Dem_00447]

Upstream requirements: RS_Diag_04192

⌈If the Dem module is requested to support enable conditions, the Dem module shall check the assigned enable conditions after the diagnostic monitor reports an event (passed/failed or pre-passed/pre-failed, refer to `Dem_SetEventStatus`).⌋

[SWS_Dem_00449]

Upstream requirements: RS_Diag_04192

⌈If one enable condition is not fulfilled, all of `Dem_SetEventStatus` calls for those events being assigned to this condition shall be ignored (no change of `event status byte`) by the Dem.⌋

Note: In case of Dem-internal debouncing the related fault detection counter will be frozen or reset (refer to chapter Figure 29 and Figure 32).

[SWS_Dem_00450]

Upstream requirements: RS_Diag_04192

⌈If all event-specific enable conditions are fulfilled, all status reports from SW-Cs and BSW modules via `Dem_SetEventStatus` for those events being assigned to these conditions shall be accepted by the Dem from this point in time on.⌋

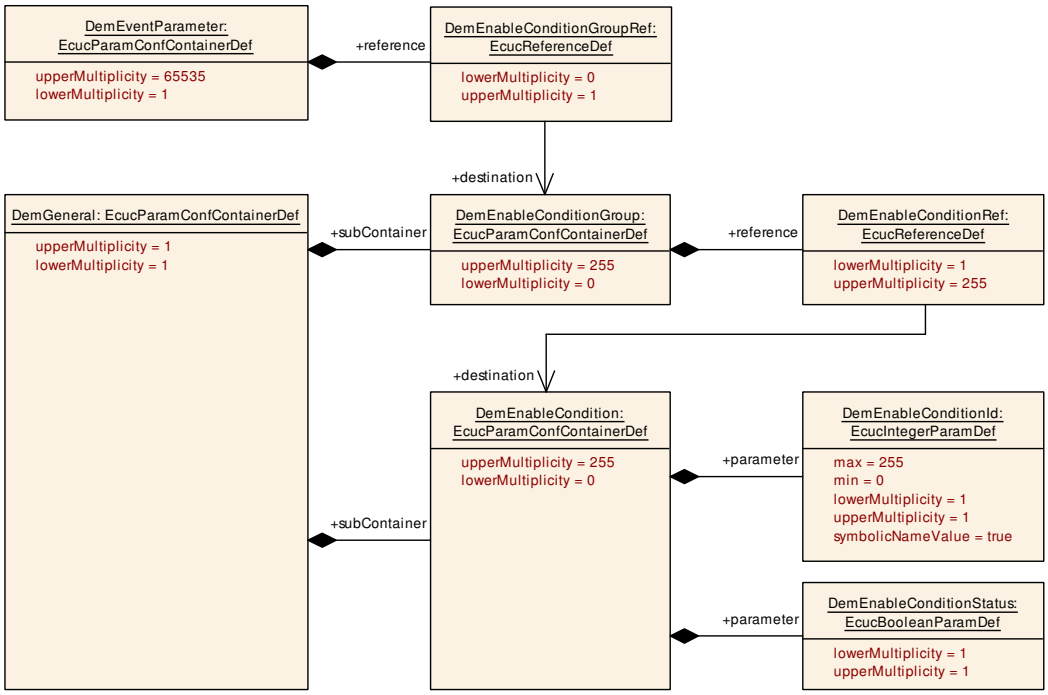


Figure 7.35: Enable condition assignment configuration

The following requirements introduce the handling of the storage conditions.

Note: The Dem module supports storage conditions and provides the API Dem_SetStorageCondition to receive the current status (condition fulfilled or not)of a specific storage condition.

Note: The Dem module supports storage conditions that can be assigned to a specific event.

Note: The storage condition groups are introduced to improve the configuration methodology of the Dem module. If a significant number of events depend always on the same storage conditions, it is less effort to select one of the defined storage condition groups instead of assigning several storage conditions to each of those events.

[SWS_Dem_00455]

Upstream requirements: RS_Diag_04219

⌈If storage conditions are used and assigned to an event, the Dem shall evaluate the assigned storage conditions after the event gets qualified as PASSED or FAILED and process DemFreezeFrameRecordTrigger, DemExtendedDataRecordTrigger and DemEventMemoryEntryStorageTrigger only if the storage condition is fulfilled.⌋

Note: The Dem module supports storage conditions and checks the assigned storage conditions after the event reports DEM_EVENT_STATUS_FDC_THRESHOLD_REACHED (e.g. either via interface call or when reaching the configured threshold in debounce algorithm).

Note: If one storage condition is not fulfilled and no respective event memory entry exists, the Dem module will not enter the reported event into the event memory.

Note: If one storage condition is not fulfilled and a respective event memory entry exists, the Dem module will not update the event memory of the reported event.

[SWS_Dem_01309]

Upstream requirements: RS_Diag_04219

⌈If at least one storage condition assigned to an event is not fulfilled, the Dem shall skip the calculation to set the bits:

- bit 2 (PendingDTC) and
- bit 3 (ConfirmedDTC).

in the corresponding event status byte and UDS status byte from 0 to 1.⌋

[SWS_Dem_00459]

Upstream requirements: RS_Diag_04219

⌈If all event-specific storage conditions are fulfilled, the Dem module shall permit the storage of the reported event.⌋

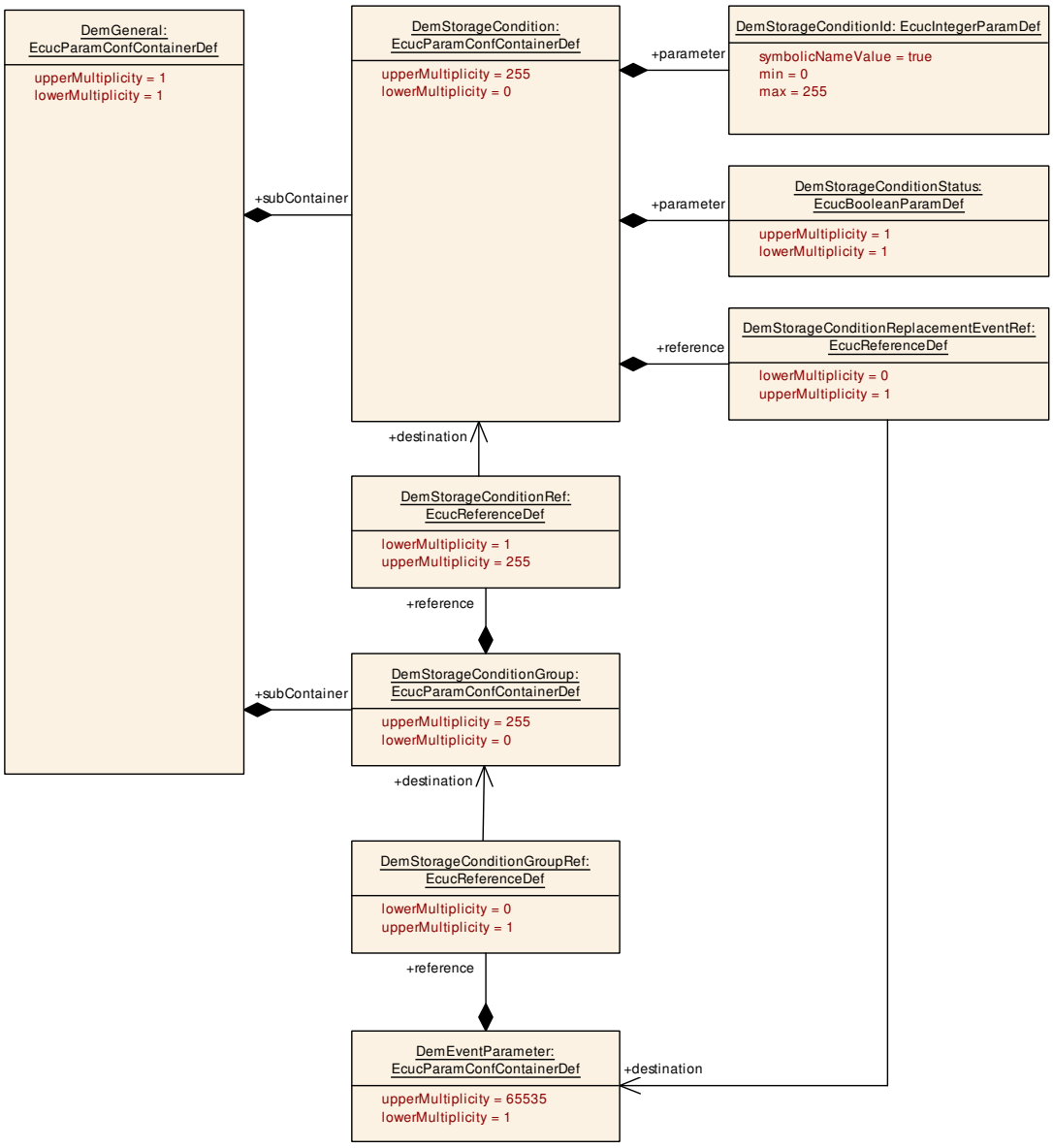


Figure 7.36: Storage condition assignment configuration

For each storage condition a replacement event may be defined (see `DemStorageConditionReplacementEventRef`). This replacement event will be used as a replacement failure info.

[SWS_Dem_01086]

Upstream requirements: RS_Diag_04137

⌈If an event is reported as FAILED and a storage condition referencing this event by `DemStorageConditionReplacementEventRef` exists and the storage condition is FALSE, the replacement event shall be set to FAILED.⌋

Note: As long as the storage condition is not fulfilled and no event assigned to the storage condition reports FAILED, the replacement event shall not be set.

Note: The replacement event has an EventStatus (`Dem_UdsStatusByteType`) as every other event.

Note: The status/behavior of the filtered event (filtered by the storage condition) is not affected by the replacement event. The failure which is filtered by the storage condition behaves identical in case of having a replacement event or in case of NOT having a replacement event configured for the storage condition.

[SWS_Dem_01089]

Upstream requirements: RS_Diag_04137

⌈The replacement event shall be reset (status PASSED), when the storage condition is fulfilled (on enabling the storage condition).⌋

[SWS_Dem_CONSTR_06167]

Upstream requirements: RS_Diag_04137

⌈the replacement event:

- shall not be mapped to a storage condition (DemStorageConditionGroupRef)
- shall not be configured to use debouncing (DemDebounceAlgorithmClass)
- shall not have an InitMForE callback/ clearEventAllowed (DemCallbackInitMForE/DemCallbackClearEventAllowed)
- shall not be configured for enableConditions (DemEnableConditionGroupRef)
- shall be configured as `DEM_EVENT_KIND_BSW`

⌋

7.7.7 Event related data

'Event related data' are additional data, e.g. sensor values or time stamp/mileage, which are stored in case of an event. ISO-14229-1[2] defines two different types of event related data: snapshot data (freeze frames) and extended data. The number or sets of stored event related data are strongly OEM / failure specific and are therefore configurable. This data is provided by SW-C or other BSW modules.

[SWS_Dem_00796] ⌈Each `event memory entry` shall support the capability to store the configured 'Event related data' (freeze frame data (DTCSnapshot) or DTCExtendedData).⌋

For details see chapter 7.7.7.4.

Note: The presence of a Confirmed UDS status does not necessarily mean that 'event related data' is available as well.

The Dem module is not in charge of validity of event related data. Time consistency of event related data is depending on data source and storage time.

The Dem module provides a configuration table to combine event related data (freeze frames & extended data) with a specific DTC number, etc. (refer to chapter 7.7.7.4).

Note: This does not define a specific implementation (e.g. look-up table, matrix, etc.). Furthermore it relates to the link between the configured data. An event is characterized by its event Id, DTC value, configured freeze frames and extended data records, etc.

7.7.7.1 Storage of freeze frame data

[SWS_Dem_00039] ⌈The Dem module shall support event-specific freeze frame storage.⌋

In general, there are two options for freeze frame configuration: (1) Non-emission related freeze frames are configured specific to one particular event. The freeze frame records can be addressed by using relative numbers (calculated or configured), so the record numbers are unique to an event (not globally). (2) Emission related freeze frames are configured globally for a particular ECU (OBD legislation requires one single freeze frame class only). The freeze frame record can be addressed per event by using the value 0x00.

[SWS_Dem_00040] ⌈The Dem module shall support the storage of one or several DIDs per freeze frame record assigned by configuration (refer to `DemFreezeFrameClass`).⌋

Note: A freeze frame is represented as a list of DIDs (refer to `DemDidClass`) or PIDs (refer to `DemPidClass`).

Note: Due to implementation reasons, the Dem usually needs to reserve memory for the maximum freeze frame size multiplied by the number of freeze frames.

[SWS_Dem_00337]

Upstream requirements: RS_Diag_04127

⌈If `DemTypeOfFreezeFrameRecordNumeration` is set to DEM_FF_RECNUM_CALCULATED (refer to `DemPrimaryMemory` or `DemUserDefinedMemory`), the `Dem` shall store `DemMaxNumberFreezeFrameRecords` number of freeze frames for this DTC.⌋

[SWS_Dem_CONSTR_06166]

Upstream requirements: RS_Diag_04127

⌈If `DemTypeOfFreezeFrameRecordNumeration` is set to DEM_FF_RECNUM_CALCULATED, parameter `DemFreezeFrameRecordTrigger` shall be set to `DEM_TRIGGER_ON_TEST_FAILED`.⌋

[SWS_Dem_00581]

Upstream requirements: RS_Diag_04127

⌈If the Dem module uses calculated record numbers, the Dem module shall numerate the event-specific freeze frame records consecutively starting by 1, based on their chronological order.⌋

[SWS_Dem_00582] ⌈If DemTypeOfFreezeFrameRecordNumeration is set to DEM_FF_RECNUM_CONFIGURED (refer to DemPrimaryMemory or DemUserDefinedMemory), the Dem shall store per event memory entry all configured freeze frame records for this DTC.⌋

[SWS_Dem_00797]

Upstream requirements: RS_Diag_04127

⌈If the FreezeFrame uses dedicated, configured record numbers (refer to DemFreezeFrameRecNumClassRef) and an event memory entry exists, the Dem module shall capture the FreezeFrame on the configured trigger (refer to DemFreezeFrameRecordTrigger) and store it to the event memory entry.⌋

In case, the storage trigger was not able to allocate an event memory entry (due to event retention) there might meanwhile the possibility (due to aging) to have suitable memory entries Therefore the FreezeFrame trigger should try again to allocate an event memory entry.

[SWS_Dem_00798]

Upstream requirements: RS_Diag_04127

⌈If the FreezeFrame uses dedicated, configured record numbers (refer to DemFreezeFrameRecNumClassRef) and no event memory entry exists, the Dem module shall first try to allocate an event memory entry as described in [SWS_Dem_00783], [SWS_Dem_00784], [SWS_Dem_00785] and [SWS_Dem_00786]. Afterwards requirement [SWS_Dem_00797] applies.⌋

[SWS_Dem_00799]

Upstream requirements: RS_Diag_04127

⌈If the DemFreezeFrameRecordTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD, the FreezeFrame shall be captured (as allowed by [SWS_Dem_00797]) each time the configured FDC threshold (refer to DemCounterBasedFdcThresholdStorageValue or DemTimeBasedFdcThresholdStorageValue) is reached (by a positive increment), but at most once per operation cycle.⌋

[SWS_Dem_01068]

Upstream requirements: RS_Diag_04127

⌈If the DemFreezeFrameRecordTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD and the event reports DEM_EVENT_STATUS_FDC_THRESHOLD_REACHED

(monitor-internal debounced event), the FreezeFrame shall be captured (refer [SWS_Dem_00797]), but at most once per operation cycle.⌋

[SWS_Dem_00800]

Upstream requirements: RS_Diag_04127

⌈If the `DemFreezeFrameRecordTrigger` is set to `DEM_TRIGGER_ON_TEST_FAILED`, the FreezeFrame shall be captured (as allowed by [SWS_Dem_00797]) each time the `UDS status bit 0` is set (changing from 0 to 1).⌋

[SWS_Dem_00801]

Upstream requirements: RS_Diag_04127

⌈If the `DemFreezeFrameRecordTrigger` is set to `DEM_TRIGGER_ON_PENDING`, the FreezeFrame shall be captured (as allowed by [SWS_Dem_00797]) each time the `UDS status bit 2` is set (changing from 0 to 1).⌋

[SWS_Dem_00802]

Upstream requirements: RS_Diag_04127

⌈If the `DemFreezeFrameRecordTrigger` is set to `DEM_TRIGGER_ON_CONFIRMED`, the FreezeFrame shall be captured (as allowed by [SWS_Dem_00797]) each time the Confirmed `UDS status bit 3` is set (changing from 0 to 1).⌋

[SWS_Dem_01351]

Upstream requirements: RS_Diag_04127

⌈If the `DemFreezeFrameRecordTrigger` is set to `DEM_TRIGGER_ON_TEST_FAILED_THIS_OPERATION_CYCLE`, the FreezeFrame shall be captured each time the UDS status bit 1 (TFTOC) has a transition from 0 to 1.⌋

[SWS_Dem_01308] Continous trigger conditions for freeze frames

Upstream requirements: RS_Diag_04127

⌈If the `DemFreezeFrameRecordTrigger` is set to `DEM_TRIGGER_ON_EVERY_TEST_FAILED`, each time an event was reported as FAILED or after optional debouncing a qualified FAILED is detected the `Dem` shall capture the `Freeze frame`.⌋

[SWS_Dem_00803]

Upstream requirements: RS_Diag_04127

⌈If the FreezeFrame uses dedicated, configured record numbers (refer to DemFreezeFrameRecNumClass) and `DemFreezeFrameRecordUpdate` is set to `DEM_UPDATE_RECORD_NO`, the FreezeFrame shall be stored only if the FreezeFrame is currently not stored in this `event memory entry`.⌋

[SWS_Dem_00804]

Upstream requirements: RS_Diag_04127

⌈If the FreezeFrame uses dedicated, configured record numbers (refer to DemFreezeFrameRecNumClass) and DemFreezeFrameRecordUpdate is set to DEM_UPDATE_RECORD_YES, the FreezeFrame shall be updated with each trigger (refer to DemFreezeFrameRecordTrigger).⌋

[SWS_Dem_00461] Asynchronous capturing of freeze frame data

Upstream requirements: RS_Diag_04127

⌈The Dem shall capture the event-specific freeze frame data at latest in the next cycle of the Dem_MainFunction.⌋

[SWS_Dem_01081] Asynchronous capturing of extended data

Upstream requirements: RS_Diag_04127

⌈The Dem shall capture the event-specific extended data at latest in the next cycle of the Dem_MainFunction.⌋

[SWS_Dem_CONSTR_06124] Dependency for DemPTOSupport ⌈DemPTOSupport shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06127] Dependency for DemMaxNumberFreezeFrameRecords ⌈DemMaxNumberFreezeFrameRecords shall only be present if DemTypeOfFreezeFrameRecordNumeration is set to DEM_FF_RECNUM_CALCULATED.⌋

[SWS_Dem_CONSTR_06128] Dependency for DemAgingCycleRef ⌈DemAgingCycleRef shall only be present if DemAgingCycleCounterThreshold is configured.⌋

[SWS_Dem_CONSTR_06129] Dependency for DemFreezeFrameRecNumClassRef ⌈DemFreezeFrameRecNumClassRef shall only be present if that DTC references a fault memory that has DemTypeOfFreezeFrameRecordNumeration is set to DEM_FF_RECNUM_CONFIGURED (refer to DemPrimaryMemory or DemUserDefinedMemory).⌋

[SWS_Dem_CONSTR_06130] Dependency for DemReportBehavior ⌈DemReportBehavior shall only be present if DemEventKind is set to DEM_EVENT_KIND_SWC.⌋

[SWS_Dem_CONSTR_06131] Dependency for DemOBDGroupingAssociativeEventsRef ⌈DemOBDGroupingAssociativeEventsRef shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06132] Dependency for DemOBDCentralizedPID21Handling ⌈DemOBDCentralizedPID21Handling shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06133] Dependency for DemOBDCentralizedPID31Handling ⌈DemOBDCentralizedPID31Handling shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06134] Dependency for DemOBDCompliancy ⌈DemOBDCompliancy shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06135] Dependency for DemOBDEngineType ⌈DemOBDEngineType shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06136] Dependency for DemOBDEventDisplacement ⌈DemOBDEventDisplacement shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06137] Dependency for DemOBDInputAcceleratorPedalInformation ⌈DemOBDInputAcceleratorPedalInformation shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06138] Dependency for DemOBDInputAmbientPressure ⌈DemOBDInputAmbientPressure shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06139] Dependency for DemOBDInputAmbientTemperature ⌈DemOBDInputAmbientTemperature shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06140] Dependency for DemOBDInputDistanceInformation ⌈DemOBDInputDistanceInformation shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06141] Dependency for DemOBDInputEngineSpeed ⌈DemOBDInputEngineSpeed shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06142] Dependency for DemOBDInputEngineTemperature ⌈DemOBDInputEngineTemperature shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06143] Dependency for DemOBDInputProgrammingEvent ⌈DemOBDInputProgrammingEvent shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06144] Dependency for DemOBDInputVehicleSpeed ⌈DemOBDInputVehicleSpeed shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_CONSTR_06145] Dependency for DemConsiderPtoStatus ⌈DemConsiderPtoStatus shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

The snapshot record data is usually captured asynchronously from reporting the event status. In certain situations the captured data needs to be aligned with the reported event status. For this purpose the Dem_SetEventStatusWithMonitorData provides the possibility to add 2 monitor data. The monitorData0 can optionally be passed to the DataServicesCS ReadData operation, so the application can align the data capture call to the initiated Dem_SetEventStatusWithMonitorData function call.

[SWS_Dem_01330] Capturing event related data with monitor data ⌈If the ReadData operation is configured to provide the monitorData0, the Dem shall add the monitorData0 from the Dem_SetEventStatusWithMonitorData to the ReadData operation parameter monitorData0.⌋

[SWS_Dem_01331] Handling of missing monitorData ⌈If the data capturing was triggered from within an API call without monitorData, the Dem shall use a value of 0 as monitorData.⌋

[SWS_Dem_00261]

Upstream requirements: RS_Diag_04127

⌈The Dem module shall use the C-callback DemRead respectively the operation ReadData of the interface DataServices_{Data} to collect all configured data elements of the respective freeze frame.⌋

[SWS_Dem_00463] ⌈If the SW-C or BSW module cannot not provide the requested data (ReadDataElement returns other than E_OK), the Dem shall fill the missing data with the padding value 0xFF, report the runtime error DEM_E_NODATAAVAILABLE to the Det and continue its normal operation.⌋

[SWS_Dem_00585] ⌈If the Dem module uses calculated record numbers, and if more than one freeze frame record is configured for a specific event, and this event is updated in the event memory, and all available freeze frame record slots for this event are occupied, the Dem module shall update the most recent record.⌋

Note: The first freeze frame record slot will always represent the first occurrence.

7.7.7.2 Pre-storage of freeze frame data

The pre-storage of freeze frames can be used for events with highly volatile freeze frame data. With the first indication of the appearance of a specific event, even if the event is not yet de-bounced or qualified, the freeze frame data is captured (e.g. because of rapid changing of event related data during running failure monitoring phase). The pre-stored freeze frame functionality is used by monitors.

[SWS_Dem_00002]

Upstream requirements: RS_Diag_04127

⌈The Dem module shall provide the configuration parameter DemFFPrestorageSupported (refer to DemEventParameter) to enable or disable pre-storage handling of freeze frames per event.⌋

Note: If DemMaxNumberPrestoredFF is set to 0, DemFFPrestorageSupported can not be enabled for any DTC.

[SWS_Dem_00334]

Upstream requirements: RS_Diag_04127

⌈If any event is configured to use pre-storage of freeze frames (refer to DemFFPrestorageSupported), the Dem module shall provide the API Dem_PrestoreFreezeFrame and Dem_ClearPrestoredFreezeFrame. Otherwise they are not provided.⌋

[SWS_Dem_00189] ⌈The Dem module shall provide the API Dem_PrestoreFreezeFrame to capture the pre-storage data of an event-specific freeze frame regardless of the UDS status bit changes.⌋

Note: If a request to prestore a freeze frame (Dem_PrestoreFreezeFrame) is interrupted by reporting an event status (Dem_SetEventStatus) or a call to Dem_ClearPrestoredFreezeFrame of the same EventId the detailed behavior is implementation specific. The freeze frame which is requested to be prestored might be used immediately or might be stored and used with next upcoming event report or vice versa.

[SWS_Dem_00808]

Upstream requirements: RS_Diag_04127

⌈The API Dem_PrestoreFreezeFrame shall return E_NOT_OK if no memory is available (see DemMaxNumberPrestoredFF).⌋

[SWS_Dem_00464] ⌈If a pre-stored freeze frame is available, the Dem module shall use the data of the pre-stored freeze frame instead of the current data at the point in time when the event related date is captured (refer to [SWS_Dem_00461]).⌋

[SWS_Dem_00969] ⌈A pre-stored freeze frame shall be released after it has been stored in the event memory or discarded.⌋

Note : A pre-stored freeze frame is considered to be stored if event retention is successful. A pre-stored freeze frame is discarded if the event retention failed.

[SWS_Dem_00191] ⌈If no pre-stored freeze frame is available, the Dem module shall behave according to Storage of freeze frame data (refer to [SWS_Dem_00039]).⌋

For details see chapter 7.7.7.1.

Note: The captured data while using pre-stored freeze frames can differ from the data, which is collected using the UDS status bit transitions as a trigger.

Note: To ensure absence of reaction to stored freeze frames of qualified events an additional freeze frame buffer should be used. Due to restrictions in hardware usage, the amount of possible entries can be restricted. Therefore, a replacement strategy could be required.

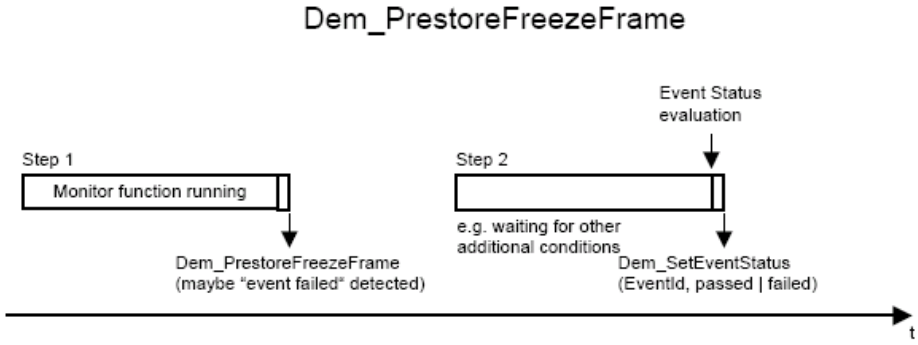


Figure 7.37: Example to use `Dem_PrestoreFreezeFrame` to prestore freeze frame data

Some events detect a failure in one operation cycle of the control unit, but only determine the conditions to allow storage of the failure in the next operation cycle. The freeze-frame of such events shows the values from the old operation cycle. For this purpose, the Dem supports non-volatile storage of pre-stored freeze-frame data. To avoid excessive storage, the storage can be enabled/disabled per event individually through the configuration `DemFFPrestorageInNvm`.

[SWS_Dem_01320] Non volatile storage of pre-stored data ⌈If `DemFFPrestorageInNvm` is set to TRUE, the Dem shall store pre-stored freeze-frame data for this event on in Nvm.⌋

[SWS_Dem_01321] Point in time to store pre-stored data in non-volatile memory ⌈If the Dem is configured to store pre-stored freezeframe data in non-volatile memory according to [SWS_Dem_01320], the Dem shall store the data once and latest during `Dem_Shutdown`.⌋

[SWS_Dem_01322] Point in time to retrieve pre-stored data from non-volatile memory ⌈If the Dem is configured to store pre-stored freezeframe data in non-volatile memory according to [SWS_Dem_01320], the Dem shall restore the pre-stored data before `Dem_Init` has returned.⌋

[SWS_Dem_01323] Consistency of non-volatile stored pre-stored data ⌈If the Dem is restoring pre-stored data according to [SWS_Dem_01322], the Dem shall only restore this data if it is consistent to the current configuration. If after reprogramming or calibration, the data layout has changed, the Dem shall skip the pre-stored data.⌋

[SWS_Dem_00050] ⌈The Dem module shall provide the API Dem_ClearPrestoredFreezeFrame to release the pre-stored freeze frame for the specific event.⌋

[SWS_Dem_00465] ⌈If an event gets qualified as passed (UDS status bit 0 changes from 1 to 0) the Dem module shall release the pre-stored freeze frame for the specific event.⌋

7.7.7.3 Storage of extended data

An extended data record contains additional information associated to a specific event that is not contained in a freeze frame (extended data, e.g. frequency counters, Aging counters, etc.). According to the DID- or PID-based configuration of freeze frame data, extended data are divided in extended data records defined by its record numbers.

[SWS_Dem_00809]

Upstream requirements: RS_Diag_04127

⌈If an event memory entry exists, the Dem module shall capture the ExtendedDataRecord on the configured trigger (refer to DemExtendedDataRecordTrigger) and store it to the event memory entry.⌋

In case, the storage trigger was not able to allocate an event memory entry (due to event retention) there might meanwhile the possibility (due to aging) to have suitable memory entries Therefore the ExtendedDataRecord trigger should try again to allocate an event memory entry.

[SWS_Dem_00810]

Upstream requirements: RS_Diag_04127

⌈If no event memory entry exists, the Dem module shall first try to allocate an event memory entry as described in [SWS_Dem_00783], [SWS_Dem_00784], [SWS_Dem_00785] and [SWS_Dem_00786]. Afterwards requirement [SWS_Dem_00809] applies.⌋

[SWS_Dem_00811]

Upstream requirements: RS_Diag_04127

⌈If the DemExtendedDataRecordTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD, the ExtendedDataRecord shall be captured (as allowed by [SWS_Dem_00810]) each time the configured FDC threshold (refer to DemCounterBasedFdcThresholdStorageValue or DemTimeBasedFdcThresholdStorageValue) is reached (by a positive increment), but at most once per operation cycle.⌋

[SWS_Dem_01069]

Upstream requirements: RS_Diag_04127

⌈If the DemExtendedDataRecordTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD and the event reports DEM_EVENT_STATUS_FDC_THRESHOLD_REACHED (monitor-internal debounced event), the ExtendedDataRecord shall be captured (refer [SWS_Dem_00810]), but at most once per operation cycle.⌋

[SWS_Dem_00812]

Upstream requirements: RS_Diag_04127

⌈If the DemExtendedDataRecordTrigger is set to DEM_TRIGGER_ON_TEST_FAILED, the ExtendedDataRecord shall be captured (as allowed by [SWS_Dem_00810]) each time the UDS status bit 0 is set (changing from 0 to 1).⌋

[SWS_Dem_00813]

Upstream requirements: RS_Diag_04127

⌈If the DemExtendedDataRecordTrigger is set to DEM_TRIGGER_ON_PENDING, the ExtendedDataRecord shall be captured (as allowed by [SWS_Dem_00810]) each time the UDS status bit 2 is set (changing from 0 to 1).⌋

[SWS_Dem_00814]

Upstream requirements: RS_Diag_04127

⌈If the DemExtendedDataRecordTrigger is set to DEM_TRIGGER_ON_CONFIRMED, the ExtendedDataRecord shall be captured (as allowed by [SWS_Dem_00810]) each time the UDS status bit 3 is set (changing from 0 to 1).⌋

[SWS_Dem_01070]

Upstream requirements: RS_Diag_04127

⌈If the DemExtendedDataRecordTrigger is set to DEM_TRIGGER_ON_PASSED, the ExtendedDataRecord shall be captured with the change of Testfailed UDS status bit 0 is reset (changing from 1 to 0). If no entry exists, on passed report no new entry shall be created.⌋

[SWS_Dem_CONSTR_06101] ⌈DemExtendedDataRecordTrigger needs to be configured. DemExtendedDataRecordTrigger shall always be configured, except for internal data elements like occurrence counters.⌋

[SWS_Dem_00815]

Upstream requirements: RS_Diag_04127

⌈If the configuration parameter DemExtendedDataRecordUpdate is set to DEM_UPDATE_RECORD_NO, the ExtendedDataRecord shall be stored only if the ExtendedDataRecord is currently not stored in this event memory entry.⌋

[SWS_Dem_00816]

Upstream requirements: RS_Diag_04127

⌈If the configuration parameter DemExtendedDataRecordUpdate is set to DEM_UPDATE_RECORD_YES, the ExtendedDataRecord shall be updated with each trigger (refer to DemExtendedDataRecordTrigger).⌋

[SWS_Dem_00282] ⌈The Dem module shall use the C-callback ReadDataElement respectively the operation ReadData of the interface DataServicesCS to collect all configured data elements which are not typed as internal data elements (refer to DemInternalDataElementClass) of the respective extended data record.⌋

[SWS_Dem_00468]

Upstream requirements: RS_Diag_04127

⌈If an event is stored or updated in the event memory, the Dem module shall store the collected extended data into the event memory entry.⌋

[SWS_Dem_01391] Configuration of DTC priority as extended data record ⌈If DemInternalDataElement is set to DEM_DTC_PRIORITY, the Dem shall set the value of this internal data element to the DTC priority assigned by DemDTCPriority for this DTC. The length of this internal data element is one byte.⌋

[SWS_Dem_CONSTR_06173] Extended data record values for primary fault memory ⌈A DemExtendedDataRecordClass container referenced from a DTC assigned to the primary memory shall have DemExtendedDataRecordNumber in the range 0x01-0xEF.⌋

[SWS_Dem_CONSTR_06174] Extended data record values for user defined event memory ⌈A DemExtendedDataRecordClass container referenced from a DTC assigned to a user defined memory shall have DemExtendedDataRecordNumber in the range 0x01-0xFE.⌋

The idea behind constraint [SWS_Dem_CONSTR_06174] is to support the definition for user defined memory based on ISO 14229-1:2020[16] DAMD1.

7.7.7.4 Configuration of Event related data

This section describes the configuration of event related data and the access of event related data from the SW-Cs/BSW modules.

Note: The configuration model follows a flexible configuration process, but does not imply any explicit implementation.

The event related data of diagnostic events contain none or one set of extended data records (refer to DemExtendedDataClass), and none or one set of freeze frame records (refer to DemFreezeFrameClass) with its calculated or configured record numbers. Therefore a class-concept is used (refer to Figure 37).

An extended data record, a DID, or a PID can contain one ore more data elements (refer to DemDataElementClass).

Note: Asynchronous DIDs, as well as DIDs with a variable length are not supported by the Dem module, and shall therefore not be connected to the Dem.

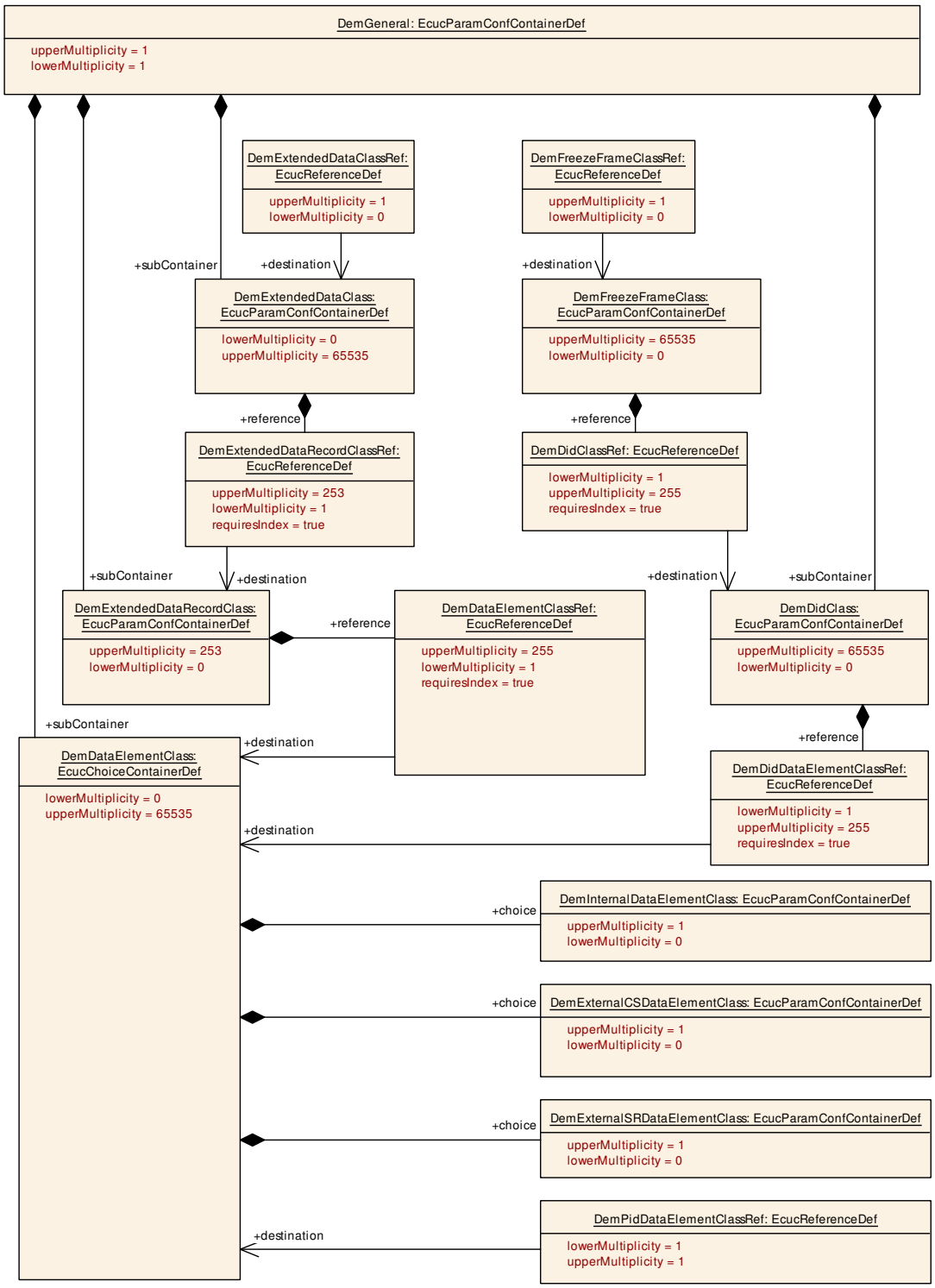


Figure 7.38: Event related data configuration

A data element is provided from a SW-C or BSW module, or is computed Dem-internally.

For each external data element a respective require-port (refer to Service Interface DataServices_<Data>) or C-callback (refer to `ReadDataElement`) is generated based on the configuration `DemExternalCSDataElementClass` and `DemExternalSRDataElementClass`. For each internal data element, the respective Dem-internal value is mapped.

Note: These data elements are typically specified in a Diagnostic Data Template.

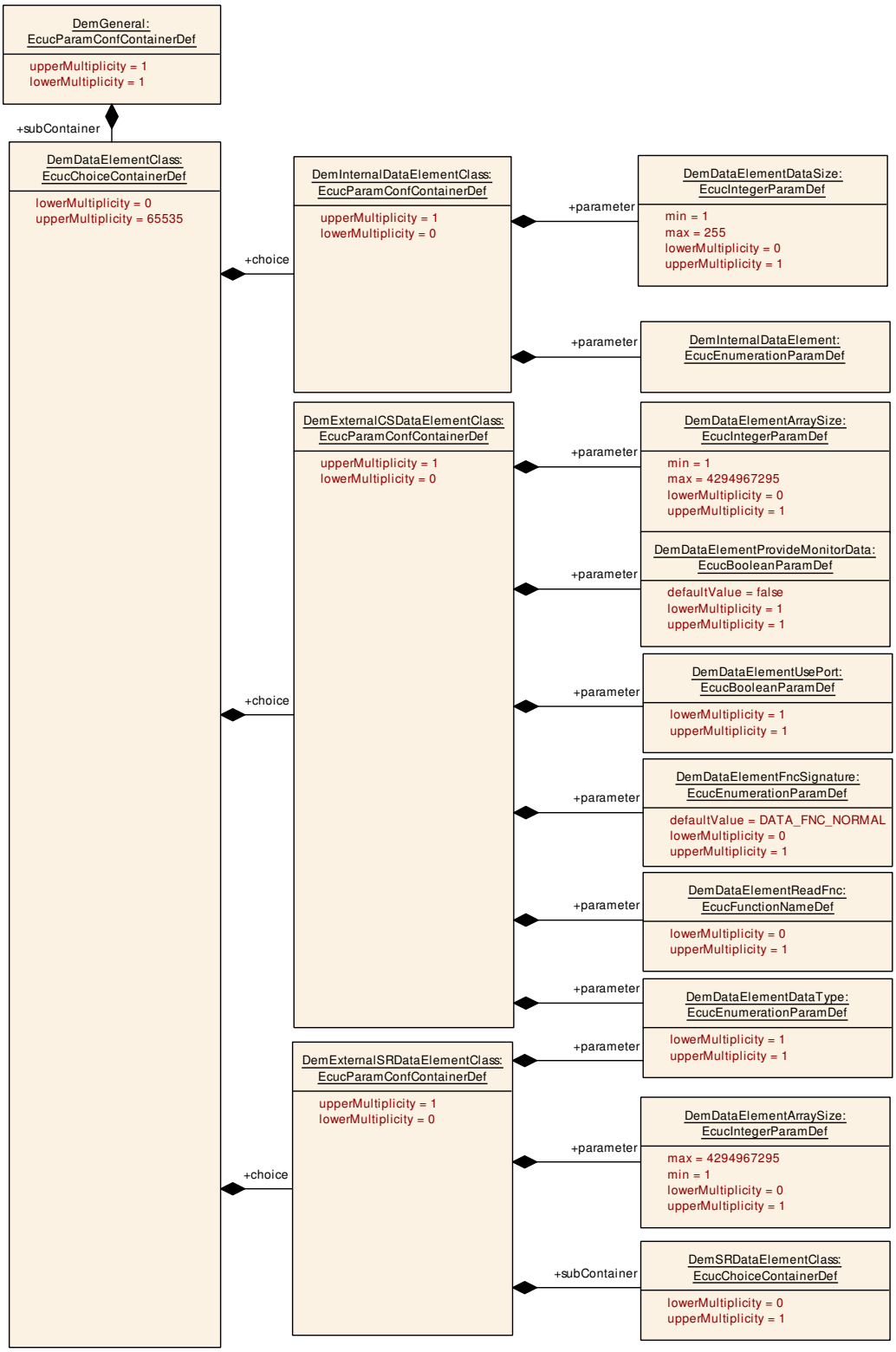


Figure 7.39: Data element configuration

[SWS_Dem_CONSTR_06158] ⌈Existence of size parameter `DemDataElement-ArraySize` [ECUC_Dem_00949] in container `DemExternalCSDataElementClass` shall be present if `DemDataElementDataType` [ECUC_Dem_00950] in the same

container is set to: UINT8_N, SINT8_N, UINT16_N, SINT16_N, UINT32_N, SINT32_N, DOUBLE_N, UINT64_N, SINT64_N, FLOAT_N.⌋

Note: DemDataElementArraySize [ECUC_Dem_00949] in container DemExternalCSDataElementClass is not required for primitive datatypes.

[SWS_Dem_CONSTR_06159] ⌈Restrictions on size parameter for 16 Bit arrays DemDataElementArraySize [ECUC_Dem_00949] shall be a multiple of 2 if the value is greater than 2 and DemDataElementDataType [ECUC_Dem_00950] is UINT16_N or SINT16_N.⌋

[SWS_Dem_CONSTR_06160] ⌈Restrictions on size parameter for 32 Bit arrays DemDataElementArraySize [ECUC_Dem_00949] shall be a multiple of 4 if the value is greater than 4 and DemDataElementDataType [ECUC_Dem_00950] is UINT32_N, SINT32_N or FLOAT_N.⌋

[SWS_Dem_CONSTR_06180] Restriction on 64 bit array parameter size ⌈Restrictions on size parameter for 64 Bit arrays DemDataElementArraySize [ECUC_Dem_00949] shall be a multiple of 8 if the value is greater than 8 and DemDataElementDataType [ECUC_Dem_00950] is DOUBLE_N, UINT64_N, or SINT64_N.⌋

[SWS_Dem_CONSTR_06161] ⌈Existence of size parameter DemDataElementArraySize [ECUC_Dem_00967] in container DemExternalSRDataElementClass shall be present if DemDataElementDataType [ECUC_Dem_00950] in the same container is set to: UINT8_N, SINT8_N, UINT16_N, SINT16_N, UINT32_N, SINT32_N, UINT64_N, SINT64_N, FLOAT_N, DOUBLE_N.⌋

Note: DemDataElementArraySize [ECUC_Dem_00967] in container DemExternalSRDataElementClass is not required for primitive datatypes.

[SWS_Dem_CONSTR_06162] ⌈Restrictions on size parameter for 16 Bit arrays DemDataElementArraySize [ECUC_Dem_00967] shall be a multiple of 2 if the value is greater than 2 and DemDataElementDataType [ECUC_Dem_00950] is UINT16_N or SINT16_N.⌋

[SWS_Dem_CONSTR_06163] ⌈Restrictions on size parameter for 32 Bit arrays DemDataElementArraySize [ECUC_Dem_00967] shall be a multiple of 4 if the value is greater than 4 and DemDataElementDataType [ECUC_Dem_00950] is UINT32_N or SINT32_N.⌋

[SWS_Dem_CONSTR_06181] Restriction on 64 bit array parameter size ⌈Restrictions on size parameter for 64 Bit arrays DemDataElementArraySize [ECUC_Dem_00967] shall be a multiple of 8 if the value is greater than 8 and DemDataElementDataType [ECUC_Dem_00950] is DOUBLE_N, UINT64_N or SINT64_N.⌋

[SWS_Dem_00469]

Upstream requirements: RS_Diag_04189, RS_Diag_04190

⌈The Dem module shall provide the ability to map Dem-internal data values (e.g. aging counter, occurrence counter) to specific data element (refer to DemInternalDataElement in DemDataElementClass)contained in an extended data records.⌋

Note: If a Dem-internal data element is mapped to e.g. an extended data record (by configuration), this information can simply be requested by UDS Service ReadDTCInformation - Sub-Service reportDTCExtendedDataRecordByDTCNumber (0x19, 06).

[SWS_Dem_00817]

Upstream requirements: RS_Diag_04127, RS_Diag_04190

⌈Internal data elements (refer to DemInternalDataElementClass) shall not be stored, but the current value shall be used instead.⌋

Note: Further customer-specific Dem-internal data elements (according to [SWS_Dem_00470]) can have different semantics (e.g. Dem-internal data elements can freeze a value or provide a current value)

[SWS_Dem_CONSTR_06169] Usage of internal data elements only for extended data records ⌈The DemInternalDataElementClass shall be only allowed for DemDataElementClass if it is only referenced by DemDataElementClassRef from DemExtendedDataRecordClass.⌋

[SWS_Dem_00471]

Upstream requirements: RS_Diag_04127, RS_Diag_04190

⌈If the configuration parameter DemInternalDataElement is set to DEM_OCCCTR, then the Dem-internal value of the occurrence counter shall be mapped to the respective data element.⌋

For details see chapter 7.3.2.

[SWS_Dem_00472]

Upstream requirements: RS_Diag_04190

⌈If the configuration parameter DemInternalDataElement is set to DEM_AGINGCTR_UPCNT or to DEM_AGINGCTR_DOWNCNT, then the Dem-internal value of the aging counter shall be mapped to the respective data element (based on [SWS_Dem_00643] or [SWS_Dem_00673]).⌋

For details see chapter 7.6.1.

[SWS_Dem_00643]

Upstream requirements: RS_Diag_04068, RS_Diag_04190

⌈If the DemAgingCycleCounterThreshold is configured, then the value of the data element DEM_AGINGCTR_UPCNT shall be based on a count-up mechanism from 0 to DemAgingCycleCounterThreshold (refer to ISO-14229-1[2], Annex D).⌋

[SWS_Dem_00673]

Upstream requirements: RS_Diag_04178, RS_Diag_04190

⌈If the DemAgingCycleCounterThreshold is configured, then the value of the data element DEM_AGINGCTR_DOWNCNT shall be based on a count-down mechanism from DemAgingCycleCounterThreshold to 0 (refer to ISO-14229-1[2], Annex D).⌋

[SWS_Dem_01043]

Upstream requirements: RS_Diag_04190

⌈If DemAgingCycleCounterThreshold is not configured the value of the internal data element DEM_AGINGCTR_DOWNCNT shall be '255'.⌋

[SWS_Dem_01044]

Upstream requirements: RS_Diag_04190

⌈If DemAgingCycleCounterThreshold is not configured the value of the internal data element DEM_AGINGCTR_UPCNT shall be '0'.⌋

[SWS_Dem_01219]

Upstream requirements: RS_Diag_04133

⌈If the configuration parameter DemInternalDataElement is set to DEM_AGINGCTR_UPCNT_FIRST_ACTIVE, the value shall be calculated analogous to DEM_AGINGCTR_UPCNT. The difference is, that in case the current aging counter has a value of 0 and the event status bits TestFailedThisOperationCycle and TestNotCompletedThisOperationCycle are set to False, a value of 1 is reported upon reading the Dem internal aging counter.⌋

[SWS_Dem_00473]

Upstream requirements: RS_Diag_04190

⌈If the configuration parameter DemInternalDataElement is set to DEM_OVFLIND, then the Dem-internal value of the overflow indication shall be mapped to the respective data element as boolean (0 = False, 1 = True).⌋

For details see chapter 7.7.2.3.

[SWS_Dem_00592]

Upstream requirements: RS_Diag_04190

⌈If the configuration parameter DemInternalDataElement is set to DEM_SIGNIFICANCE, then the (static) Dem-internal value of the DTC significance shall be mapped to the respective data element with 0 = OCCURRENCE and 1 = FAULT.⌋

For details see chapter 7.4.6.

[SWS_Dem_01084]

Upstream requirements: RS_Diag_04190

⌈If the configuration parameter DemInternalDataElement is set to DEM_CURRENT_FDC, then the Dem-internal value of the current fault detection counter shall be mapped to the respective data element with the following value-translation: [0..127]->[0x00..0x7F] and [-1..-128]->[0xFF..0x80].⌋

For details see chapter 7.7.3.3 and chapter 7.7.3.4.

[SWS_Dem_00818]

Upstream requirements: RS_Diag_04068, RS_Diag_04190

⌈If the configuration parameter DemInternalDataElement is set to DEM_MAX_FDC_SINCE_LAST_CLEAR, then the Dem-internal value of the maximum Fault Detection Counter sincle last clear shall be mapped to the respective data element.⌋

For details see chapter 7.7.3.6.

[SWS_Dem_00819]

Upstream requirements: RS_Diag_04127, RS_Diag_04190

⌈If the configuration parameter DemInternalDataElement is set to DEM_MAX_FDC_DURING_CURRENT_CYCLE, then the Dem-internal value of the maximum Fault Detection Counter during current operation cycle shall be mapped to the respective data element.⌋

For details see chapter 7.7.3.6.

[SWS_Dem_00820]

Upstream requirements: RS_Diag_04127, RS_Diag_04190

⌈If the configuration parameter DemInternalDataElement is set to DEM_CYCLES_SINCE_LAST_FAILED, then the Dem-internal value of the operation cycle counter since last failed shall be mapped to the respective data element.⌋

For details see chapter 7.6.1.1.

[SWS_Dem_00821]

Upstream requirements: RS_Diag_04189, RS_Diag_04190

⌈If the configuration parameter DemInternalDataElement is set to DEM_CYCLES_SINCE_FIRST_FAILED, then the Dem-internal value of the operation cycle counter since first failed shall be mapped to the respective data element.⌋

For details see chapter 7.6.1.2.

[SWS_Dem_00822]

Upstream requirements: RS_Diag_04190

⌈If the configuration parameter DemInternalDataElement is set to DEM_FAILED_CYCLES, then the Dem-internal value of the failed operation cycle counter shall be mapped to the respective data element.⌋

For details see chapter 7.6.1.3.

The Dem module may be extended with further specific Dem-internal data elements, if a specific configuration requires particular data values, which are computed Dem-internally.

[SWS_Dem_01350] Mapping of static event associated identifier to internal data elements

Upstream requirements: RS_Diag_04190

⌈If the DemInternalDataElement is set to DEM_EVENT_ASSOCIATED_IDENTIFICATION, the Dem shall use the static configured value in DemEventAssociatedIdentification as value for this internal data element.⌋

[SWS_Dem_01365] Mapping of IUMPR ratio on internal data element

Upstream requirements: RS_Diag_04252, RS_Diag_04190

⌈If the DemInternalDataElement is set to DEM_IUMPR, the Dem shall use the DTC assigned IUMPR value in SAE J1979-2 format as value for this internal data element.⌋

[SWS_Dem_CONSTR_06175] Prerequisites for use of DEM_IUMPR

Upstream requirements: RS_Diag_04252

⌈If an internal data element uses DEM_IUMPR the corresponding event shall be referenced by DemDiagnosticEventRef.⌋

[SWS_Dem_01366] Mapping of DTR value on internal data element

Upstream requirements: RS_Diag_04252, RS_Diag_04190

⌈If the DemInternalDataElement is set to DEM_DTR, the Dem shall use the DTC assigned DTR value in SAE J1979-2 format as value for this internal data element.⌋

[SWS_Dem_CONSTR_06176] Prerequisites for use of DEM_DTR

Upstream requirements: RS_Diag_04252

⌈If an internal data element uses DEM_DTR the corresponding event shall be referenced by DemDtrEventRef.⌋

[SWS_Dem_01367] Mapping of Monitor Activity Data on internal data element

Upstream requirements: RS_Diag_04252, RS_Diag_04190

⌈If the DemInternalDataElement is set to DEM_MONITOR_ACTIVITY_DATA, the Dem shall support per DTC the Monitor Activity Data according to SAE J1979-2 and map it as value for this internal data element.⌋

[SWS_Dem_01427] Mapping of 3-byte OBD DTC to internal data

Upstream requirements: RS_Diag_04190

⌈If the DemInternalDataElement is set to DEM_OBDDTC_3BYTE and DemSupportedObdUdsDtcSeparation is set to TRUE, the Dem shall use the corresponding 3-byte OBD DTC value configured in DemDtcValue3Byte as value for this internal data element. If no matching DemDtcValue3Byte is configured for the associated DTC, the Dem shall set the value of DEM_OBDDTC_3BYTE to 0x000000.⌋

[SWS_Dem_01428] Mapping of 2-byte OBD DTC to internal data

Upstream requirements: RS_Diag_04190

⌈If the DemInternalDataElement is set to DEM_OBDDTC_2BYTE, the Dem shall use the corresponding 2-byte OBD DTC value configured in DemDtcValue as value for this internal data element. If no matching DemDtcValue is configured for the associated DTC, the Dem shall set the value of DEM_OBDDTC_2BYTE to 0x0000.⌋

[SWS_Dem_CONSTR_06182] Restriction on 3-byte OBD DTC internal data element ⌈The DemInternalDataElement shall not be configured as DEM_OBDDTC_3BYTE if DemSupportedObdUdsDtcSeparation is not configured or is set to FALSE.⌋

[SWS_Dem_CONSTR_06168] Required DEM_EVENT_ASSOCIATED_IDENTIFICATION for events ⌈If an DemEvent refers an DemDTC that itself requires snapshot record or extended data records that contain a DemInternalDataElement = DEM_EVENT_ASSOCIATED_IDENTIFICATION, the DemEventAssociatedIdentification shall be configured for that event.⌋

[SWS_Dem_00470]

Upstream requirements: RS_Diag_04190

⌈If the Dem module implements customer-specific Dem-internal data elements, the configuration parameter DemInternalDataElement shall be extended with the respective enumeration values.⌋

Note: The computation of any Dem-internal data value, which is not configured as data element, can be discarded (if it is not necessary for other internal behavior handling).

[SWS_Dem_01045]

Upstream requirements: RS_Diag_04190

⌈If the configuration parameter DemInternalDataElement is set to DEM_CURRENT_FDC, then the fault detection counter shall be mapped to the respective data element.⌋

For details see chapter 7.7.3.

[SWS_Dem_01334] Usage of monitorData0 in internal data ⌈If DemInternalDataElement is set to DEM_MONITOR_DATA_0, the Dem shall use the 32 bit value

of the provided monitorData0 parameter value of the call Dem_SetEventStatus-WithMonitorData as internal data value.⌋

[SWS_Dem_01314] Usage of monitorData1 in internal data ⌈If DemInternal-DataElement is set to DEM_MONITOR_DATA_1, the Dem shall use the 32 bit value of the provided monitorData1 parameter value of the call Dem_SetEventStatus-WithMonitorData as internal data value.⌋

[SWS_Dem_00918] ⌈The Dem shall treat the non-integer data type uint8[n] either like an integer data type of the matching size or leave the contents uninterpreted in case the DemDataElementEndianness is configured to OPAQUE.⌋

[SWS_Dem_00919] ⌈The Dem shall interpret opaque data as uint8[n] and shall always map it to an n-bytes sized signal.⌋

Note: For opaque data endianness, conversion has to be configured to OPAQUE.

[SWS_Dem_00920] ⌈The Dem shall extend the endianness conversion to signed data types.⌋

[SWS_Dem_01216]

Upstream requirements: RS_Diag_04189

⌈To serialize the required signed- and unsigned integer AUTOSAR data types into the FreezeFrame/ExtendedRecord the target endianness configured in DemData-ElementEndianness shall be considered. If DemDataElementEndianness is not present, the DemDataElementDefaultEndianness shall be used instead.⌋

7.7.7.5 Notification of data changes

The Dem module shall notify other SW-Cs / BSW modules about updates of the event related data in the event memory (refer to chapter 8.5.4.8).

An update of the event related data occurs every time, a new event memory entry is done or an existing is updated.

Note: The Dem does not evaluate the return value (e.g. if other than E_OK) of this callback function.

Note: The configuration container DemCallbackEventDataChanged (in DemEvent-Parameter) is used to specify the related port or c-callback per event.

[SWS_Dem_00475]

Upstream requirements: RS_Diag_04155

⌈If 'event related data' (extended data or freeze frame data) of an event memory entry is added or updated AND notifications on data changes are configured via

DemCallbackEventDataChanged, the Dem shall trigger these configured event-specific notifications as well as the general notification GeneralCallbackEvent-DataChanged.⌋

Note: In case of:

- deletion
- displacement (refer to chapter 7.7.2.4)
- aging (refer to chapter 7.7.8)

the Dem doesn't trigger the datachanged-callbacks.

[SWS_Dem_01062] ⌈The functions Dem_GetEventExtendedDataRecordEx and Dem_GetEventFreezeFrameDataEx shall consider only primary and user defined memory.⌋

Note: The origin of the Event can be derived from the EventId (DemMemory-DestinationRef).

[SWS_Dem_00479]

Upstream requirements: RS_Diag_04024

⌈The function Dem_GetEventFreezeFrameDataEx shall report the data of the DID (defined by parameter DataId) in the requested freeze frame record (defined by parameter RecordNumber, except RecordNumber equal 0xFF) of the requested diagnostic event (EventId). If the RecordNumber is equal to 0xFF and parameter DemType-OfFreezeFrameRecordNumeration is set to DEM_FF_RECNUM_CALCULATED (refer to DemPrimaryMemory or DemUserDefinedMemory) the most recent record shall be used, otherwise E_NOT_OK shall be returned.⌋

[SWS_Dem_01194]

Upstream requirements: RS_Diag_04024

⌈The function Dem_GetEventFreezeFrameDataEx shall return the WWH-OBD freeze frame when called with RecordNumber 0x00. If WWH-OBD is not supported, the function shall return DEM_NO_SUCH_ELEMENT instead (refer to DemOBDSupport).⌋

[SWS_Dem_00991]

Upstream requirements: RS_Diag_04127

⌈The format of the data in the destination buffer (DestBuffer) of the function Dem_GetEventFreezeFrameDataEx is raw hexadecimal values and contains no header-information like RecordNumber or DataId. The size of the buffer equals to the configuration settings of all respective data elements.⌋

[SWS_Dem_00477]

Upstream requirements: RS_Diag_04195

⌈The function Dem_GetEventExtendedDataRecordEx shall report the data of the extended data record of the requested diagnostic event.⌋

[SWS_Dem_00989] ⌈The format of the data in the destination buffer (DestBuffer) of the function Dem_GetEventExtendedDataRecordEx is raw hexadecimal values and contains no header-information like RecordNumber.⌋

Note: The Dcm uses the function Dem_GetNextExtendedDataRecord instead of the function Dem_GetEventExtendedDataRecordEx.

[SWS_Dem_00995]

Upstream requirements: RS_Diag_04189

⌈If the interfaces Dem_GetEventFreezeFrameDataEx and Dem_GetEventExtendedDataRecordEx are called in the context of GeneralCallbackEventDataChanged or CallbackEventDataChanged, the data of the triggering event shall be retrievable (return value E_NOT_OK is not allowed).⌋

[SWS_Dem_00996] ⌈Dem_GetEventFreezeFrameDataEx shall return E_NOT_OK in case the requested FreezeFrame data is currently not accessible (e.g. in case of asynchronous preempted data retrieval from application).⌋

[SWS_Dem_00997] ⌈Dem_GetEventExtendedDataRecordEx shall return E_NOT_OK if the requested data is currently not accessible (e.g.in case of asynchronous preempted data retrieval from application).⌋

7.7.8 Aging of diagnostic events

The Dem module provides the ability to remove a specific event from the event memory, if its fault conditions are not fulfilled for a certain period of time (operation cycles). This process is called as "aging" or "unlearning".

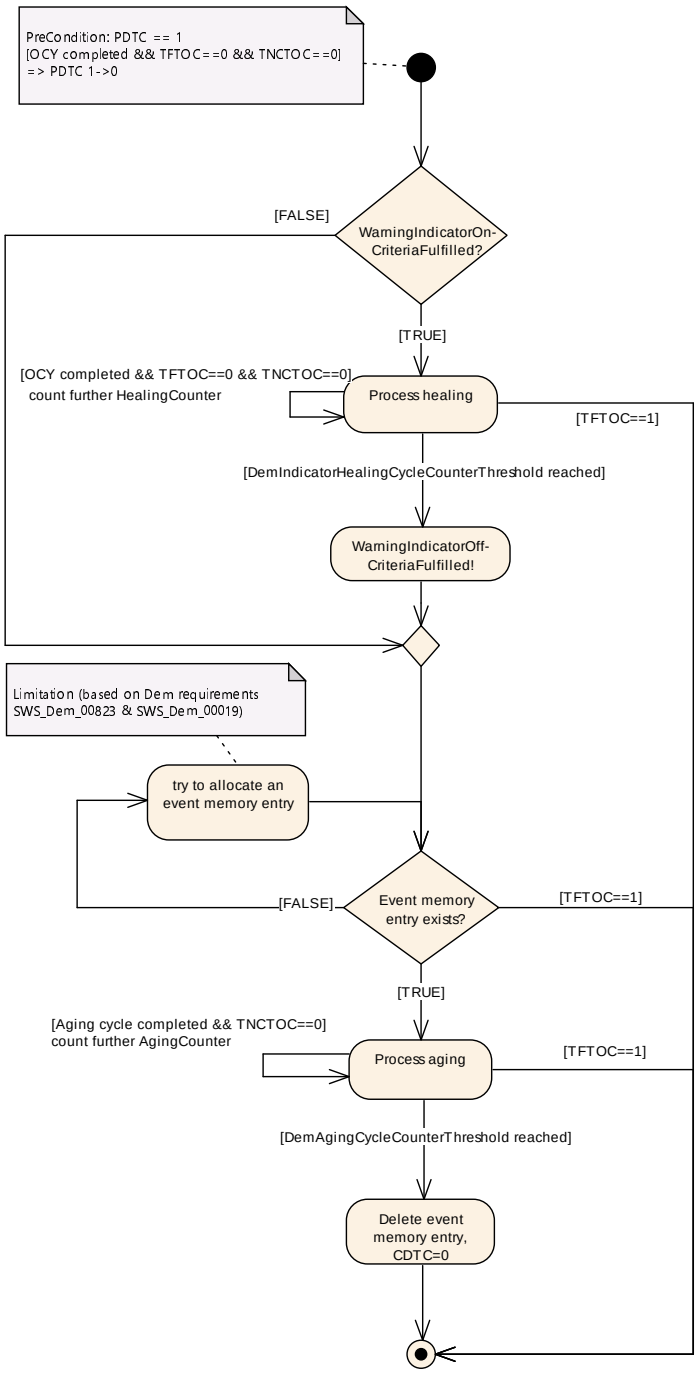


Figure 7.40: General diagnostic event deletion processing

[SWS_Dem_00698]

Upstream requirements: RS_Diag_04133, RS_Diag_04178

⌈The process of `aging` (counting of `aging counter`) starts when `healing` is completed (WarningIndicatorRequested bit == 0).⌋

For details see figure 7.40.

[SWS_Dem_00019]

Upstream requirements: RS_Diag_04133, RS_Diag_04068, RS_Diag_04178

⌈The Dem module shall support at least an aging counter for each event memory entry.⌋

[SWS_Dem_00985]

Upstream requirements: RS_Diag_04133, RS_Diag_04068, RS_Diag_04178

⌈The aging counter shall be calculated based on the referenced aging/operation cycle, if DemAgingCycleCounterThreshold is configured for this event.⌋

[SWS_Dem_00492]

Upstream requirements: RS_Diag_04133

⌈The Dem module shall be able to cover the current value of the aging counter of each individual event memory entry, to support an output.⌋

Note: For extended fault analysis, it is possible to map the current value of the aging counter to a specific data element of an extended data record (refer to chapter 7.7.7.3 Storage of extended data).

[SWS_Dem_00493]

Upstream requirements: RS_Diag_04133

⌈DemAgingCycleCounterThreshold defines the number of completed aging cycles, after which the event memory entry shall be deleted(aged) from the event memory.⌋

Note: For completed aging cycles definition see [SWS_Dem_00489]

[SWS_Dem_00823]

Upstream requirements: RS_Diag_04067

⌈If configuration parameter DemResetConfirmedBitOnOverflow is set to false and in case an event has UDS status bit 3 set and gets qualified as passed and is not stored in an event memory entry the Dem module shall try to allocate an event memory entry to get an aging counter.⌋

Note: If it is not possible to allocate an event memory entry, the aging delays until an event memory entry becomes available. (either by [SWS_Dem_00823] or by [SWS_Dem_00824]).

[SWS_Dem_00824]

Upstream requirements: RS_Diag_04133

⌈If configuration parameter DemResetConfirmedBitOnOverflow is set to false and an event memory entry aging occurs the Dem module shall check for other events having UDS status bit 3 set to 1 and UDS status bit 0 set to 0. If such an

event is found, the Dem shall allocate an event memory location to get an aging counter.⌋

Note: The priorizitation which event is choosen is implementation specific.

[SWS_Dem_00498]

Upstream requirements: RS_Diag_04133

⌈Upon event aging counter reach threshold DemAgingCycleCounterThreshold, the UDS status bit 3 shall be set to 0.⌋

Note: All other UDS status bits are not modified by aging of corresponding event memory entry.

[SWS_Dem_00161]

Upstream requirements: RS_Diag_04195

⌈The Dem module shall handle the reoccurrence of unlearned events like new events, since they were previously deleted from the event memory by aging.⌋

[SWS_Dem_00489]

Upstream requirements: RS_Diag_04133

⌈The Dem module shall only allow processing (counting further) the value of the aging counter, if the related event is stored in the event memory and is qualified as passed.⌋

Note: Aging is independent of the UDS status bit 3 and therefore independent of the fault confirmation (refer to chapter 7.7.4).

[SWS_Dem_01054]

Upstream requirements: RS_Diag_04133

⌈Upon event aging counter reach threshold DemAgingCycleCounterThreshold, the UDS status bit 5 (TestFailedSinceLastClear) shall be set to 0 if DemStatusBitHandlingTestFailedSinceLastClear is set to DEM_STATUS_BIT_AGING_AND_DISPLACEMENT.⌋

[SWS_Dem_01075] ⌈Upon event aging, the Dem shall remove the event related Snapshot data (Freeze frame) and extended data records from the event memory.⌋

[SWS_Dem_01185]

Upstream requirements: RS_Diag_04133

⌈In case there is no memory location available or occupied by an individual event, the UDS status bit TestFailedSinceLastClear of this event will not be aged.⌋

[SWS_Dem_00494] ⌈The Dem module shall provide the configuration parameter DemAgingCycleRef (refer to DemDTCAttributes) defining the event-specific operation/aging cycle, whose status change triggers the processing (counting further) of the aging counter value.⌋

Note: Refer to chapter 7.6 for the handling of the operation cycle.

[SWS_Dem_00490]

Upstream requirements: RS_Diag_04133

⌈If the configuration parameter DemAgingRequiresTestedCycle is set to False, the Dem module shall process (count further) the aging counter value, if the respective aging cycle ends/restarts.⌋

Note: The aging counter in [SWS_Dem_00490] is processed also if no new test result is available in the respective aging cycle.

[SWS_Dem_00826]

Upstream requirements: RS_Diag_04178

⌈If the configuration parameter DemAgingRequiresTestedCycle is set to True, the Dem module shall process (count further) the aging cycle counter value, if the respective aging cycle ends/restarts and the UDS status bit 6 is set to zero.⌋

[SWS_Dem_01214]

Upstream requirements: RS_Diag_04133

⌈If the configuration parameter DemAgingRequiresNotFailedCycle is set to true, the aging counter shall only be processed in an operation cycle without a test failed report.⌋

[SWS_Dem_01215]

Upstream requirements: RS_Diag_04133

⌈If the configuration parameter DemAgingRequiresNotFailedCycle is set to false, the aging counter behavior shall not be influenced.⌋

7.7.9 Healing of diagnostic events

The Dem module provides the ability to activate and deactivate indicators per event stored in the event memory. The process of deactivation is defined as healing of a diagnostic event (refer to Figure 7.40).

[SWS_Dem_01056] ⌈The Dem shall process healing only on passed events using the current operation cycle DemOperationCycleRef.⌋

7.7.9.1 Warning indicator handling

The detailed configuration of the warning indicator handling within the Dem is shown in Figure 7.41.

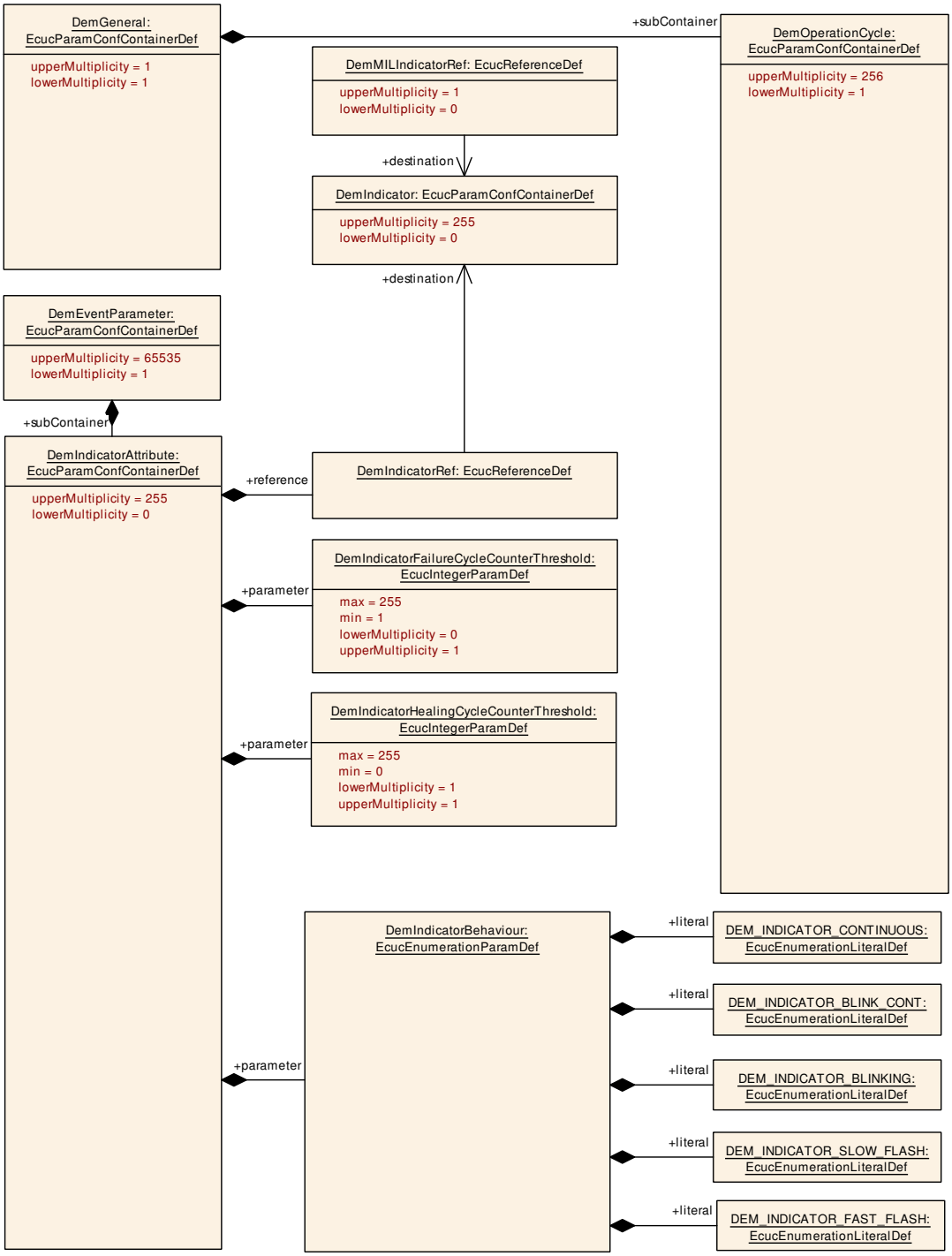


Figure 7.41: Warning indicator configuration

The Dem module supports event specific counters to activate and deactivate indicators. These counters are calculated based on the configured failure and healing process

cycles (e.g. to turn on the MIL upon fault confirmation, and turn off the MIL after subsequent healing over three OBD-driving cycles).

Note: During the integration process of the Dem module, different indicators and behaviors (e.g. indicator lamps, text messages or icons) can be assigned to an event.

[SWS_Dem_00566]

Upstream requirements: RS_Diag_04067

⌈If more than one indicator is configured for a specific event, the Dem module shall use a logical OR operation of all combined warning indicators assigned to this event to calculate the UDS status bit 7 (WarningIndicator).⌋

[SWS_Dem_00501] ⌈The condition WarningIndicatorOnCriteriaFulfilled shall be fulfilled if at least one respective event indicator failure counter has reached its DemIndicatorFailureCycleCounterThreshold of tested and failed DemOperationCycles.⌋

[SWS_Dem_00503] ⌈If any indicator is configured (via DemIndicatorAttribute) the Dem module shall set at the end of an operation cycle the UDS status bit 7 (WarningIndicatorRequested) to 0, if the following conditions are fulfilled:

- at least one DemIndicatorHealingCycleCounterThreshold is greater than 0
- all respective indicator healing process cycle counters, which count the number of tested and passed healing DemOperationCycles, have reached their DemIndicatorHealingCycleCounterThreshold
- WIRbit is not enabled by calling the API Dem_SetWIRStatus

⌋

[SWS_Dem_01233] ⌈If an indicator is configured (via DemIndicatorAttribute) the Dem module shall set when the Event is reported/qualified to PASSED the UDS status bit 7 (WarningIndicatorRequested)to 0, if the following conditions are fulfilled:

- all DemIndicatorHealingCycleCounterThreshold are equal to 0
- WIRbit is not enabled by calling the API Dem_SetWIRStatus

⌋

7.7.9.2 User controlled WarningIndicatorRequested-bit

In some cases (e.g. controlling a failsafe reaction in application) the WIR-bit of a corresponding event in Dem shall be set/reset by an especial “failsafe SW-C”.

The failsafe SW-C has to ensure a proper status of the WIR-bit (e.g. regarding to ISO-14229-1[2] or manufacture specific requirements). Therefore, the failsafe SW-C can

use existing Dem mechanism to get the information about status changes of events in Dem (e.g. Callback EventUdsStatusChanged).

The failsafe SW-C shall report the required WIR-status to Dem (via Dem_SetWIRStatus and has to ensure that the current WIR-status of an event (in Dem) fits to the current failsafe-status in application:

- failsafe running: WIR-bit shall be set to "1"
- failsafe not running: WIR-bit shall be set to "0"

The failsafe SW-C has to report the status after every change of failsafe state.

Each invocation of `Dem_SetWIRStatus` updated the WIR-bit for the corresponding event (see parameter EventId)

Due to not storing the StatusOfDTC-bit 7 (WIR-bit) on ECU shutdown the failsafe SW-C has to ensure that the WIR-bit of an event fit to the current failsafe status after `Dem_Init`.

[SWS_Dem_00831] ⌈Dem shall provide a function to control (set/reset) the WarningIndicatorRequested-bit of a configured event (in Dem) regarding to e.g. failsafe state.⌋

[SWS_Dem_00832] ⌈Setting of the WIR-bit of an event can be controlled via `Dem_SetWIRStatus` OR by the Dem internal WIR-bit handling. (OR-Operation).⌋

Note: The parallel use of `Dem_SetWIRStatus` and the Dem internal WIR-Bit handling (at the same time) is needed for example for OBD systems, where the WIR-Bit needs to be activated according to the legislation (Dem-Internal) and a system reaction is controlled by the same event. Therefore the WIR-Bit needs to stay active until the system reaction is deactivated (`Dem_SetWIRStatus`).

[SWS_Dem_00833] ⌈The WIR-bit of the corresponding event shall be set to "1" if `Dem_SetWIRStatus` is called with parameter WIRStatus = TRUE .⌋

[SWS_Dem_00834] ⌈The WIR-bit of the corresponding event shall be set to "0" if `Dem_SetWIRStatus` is called with WIRStatus = FALSE and no referenced Dem Indicator(s) are set.⌋

[SWS_Dem_00836] ⌈During disabled ControlDTCSettings the WIR-bit of an event shall not be changed via `Dem_SetWIRStatus` and the function shall return E_NOT_OK.⌋

Note: In case the failsafe application is not able to set the WIR bit (`Dem_SetWIRStatus` returned E_NOT_OK), the failsafe application needs to observe the general status of the event and coordinate the retry of `Dem_SetWIRStatus` itself e.g. by using the callback function `InitMonitorForEvent` with parameter `DEM_INIT_MONITOR_REENABLED`.

[SWS_Dem_01303] Asynchronous behavior of Dem_SetWirStatus ⌈The Dem shall process a call of the Dem_SetWIRStatus asynchronously. This means that the final result is available at a later point in time.⌋

.

A later point in time is meant to be implementation specific, it could be the next main function and after Dem_SetWIRStatus has returned.

7.7.9.3 Handling of the warning indicator lamp (MIL)

[SWS_Dem_00546]

Upstream requirements: RS_Diag_04110

⌈For OBD-relevant ECUs, the Dem module shall provide the configuration parameter DemMILIndicatorRef to indicate that the configured indicator controls the MIL activation and deactivation.⌋

[SWS_Dem_00567] ⌈If an indicator is configured for controlling the MIL of an OBD-relevant ECU, the Dem module shall use the configured event failure cycle counter of this event (refer to 0 Fault confirmation) to define the maximum number of tested and failed cycles, before the stored event activates the respective indicator.⌋

Note: For OBD systems, the activation of the Malfunction Indicator Lamp (MIL) is linked with the entry to confirmed state. Therefore the event specific fault confirmation counter (refer to the configuration parameter DemEventConfirmationThreshold and DemOperationCycleRef) has to be consistent with the indicator failure cycle counter.

Note: Leaving Pending state and the deactivation of the MIL is controlled by the configuration of the indicator healing process cycle counter.

[SWS_Dem_00701] ⌈If the MIL is deactivated and the event is confirmed the MIL shall be reactivated according to [SWS_Dem_00567] i.e. again according to the indicator failure cycle counter is reaching it's threshold.⌋

[SWS_Dem_00535] ⌈In case of OBD-relevant events the indicator cycles shall be based on cycles defined by OBD legislation.⌋

[SWS_Dem_CONSTR_06165] Dependency for DemMILIndicatorRef ⌈DemMILIndicatorRef shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

7.7.9.4 Notification and Set of the warning indicator status

[SWS_Dem_00046] ⌈The Dem module shall provide the API Dem_GetIndicator-Status that a software component can get information about the calculated indicator status.⌋

[SWS_Dem_01361]

Upstream requirements: RS_Diag_04110, RS_Diag_04204

⌈If an indicator is activated in multiple modes at the same time, the API Dem_Get-IndicatorStatus shall return the status considering the following priority:

- DEM_INDICATOR_BLINK_CONT (highest priority)
- DEM_INDICATOR_BLINKING
- DEM_INDICATOR_FAST_FLASH
- DEM_INDICATOR_SLOW_FLASH
- DEM_INDICATOR_CONTINUOUS
- DEM_INDICATOR_SHORT
- DEM_INDICATOR_ON_DEMAND
- DEM_INDICATOR_OFF (lowest priority)

⌋

Note: The returned indicator status type in [SWS_Dem_01361] is the most natural order of the indicator status in case multiple events trigger the status. At the time this specification was written this is compliant to known legislated requirements as well as to OEMs. AUTOSAR explicitly allows the option to change the order of indicator status according to [SWS_Dem_01361] to fulfil manufacture specific indicator states as well as legislated requirements to come.

7.8 BSW Error Handling

Beside application software components also the basic software (BSW) can detect errors (e.g. hardware driver faults), especially during startup (ref. to document [11] for further details). For these errors (only a small number compared to application specific events), some additional aspects apply:

- Errors can be detected at startup before Dem is fully initialized
- Errors can be reported during startup, information needs to be buffered until Dem is fully available

- Errors can be reported between startup and shutdown, information needs to be buffered and need to be processed by the Dem main function (RTE related call tree requirement)
- Entries in the event memory can have a different configuration (e.g. no emphasis on freeze frame data for the workshop)

Dem_SetEventStatus is used by BSW modules to report errors from the point in time when the Dem module is pre-initialized. Within Dem_Init, the queued events are processed. During normal operation (after full initialization), the queuing mechanism of the API Dem_SetEventStatus is necessary to process the reported fault within the main function of the Dem module.

Dem_SetEventStatus can be used by BSW modules to report errors from the point in time when the Dem module is pre-initialized. Within Dem_Init, the queued events are processed.

[SWS_Dem_01212]

Upstream requirements: SRS_BSW_00339

⌈After Dem_PreInit, the Dem shall support debouncing by calling API Dem_SetEventStatus with EventStatus set to DEM_EVENT_STATUS_PREPASSED or DEM_EVENT_STATUS_PREFAILED.⌋

The Fim can be run in trigger or polling mode. For the initialization sequence this mode is not important, the Dem always initializes the Fim.

[SWS_Dem_01213]

Upstream requirements: RS_Diag_04068

⌈The Dem-internal debouncing (counters and timers) shall be initialized by calling Dem_PreInit.⌋

[SWS_Dem_00167]

Upstream requirements: SRS_BSW_00339

⌈The Dem module shall provide a buffer mechanism to queue events which are reported before Dem_Init via Dem_SetEventStatus as qualified as Passed or Failed or reaching the qualification (debouncing). Reporting of DEM_EVENT_STATUS_FDC_THRESHOLD_REACHED shall not be queued.⌋

[SWS_Dem_00207]

Upstream requirements: SRS_BSW_00339

⌈The size of the queue of the function Dem_SetEventStatus which is used while Dem is not initialized is configurable by the configuration parameter DemBswErrorBufferSize.⌋

[SWS_Dem_01079]

Upstream requirements: SRS_BSW_00339

⌈The Dem shall determine the queue size depending on the number of events that can be reported before Dem_Init, i.e., those events with undefined DemReportBehavior or DemReportBehavior = REPORT_BEFORE_INIT. In case DemBswErrorBufferSize is configured, it overwrites the calculated size of the queue.⌋

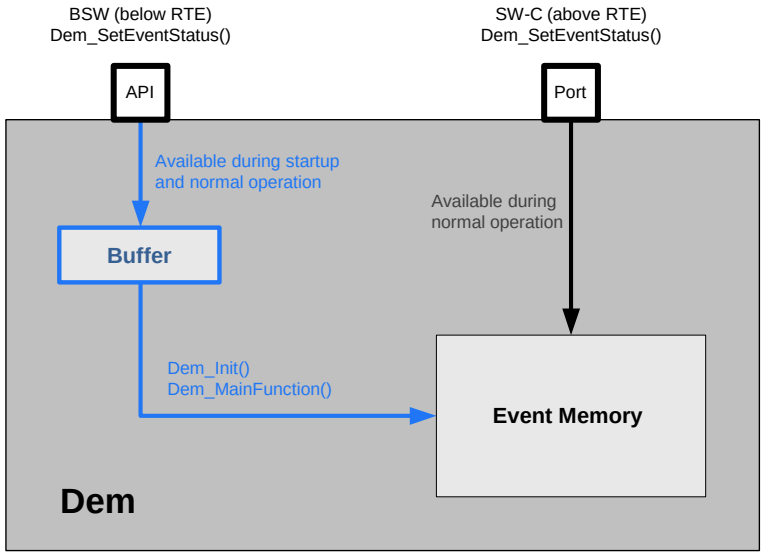


Figure 7.42: Dem_SetEventStatus buffering behavior

[SWS_Dem_01289]

Upstream requirements: SRS_BSW_00339

⌈All events which are reported via Dem_SetEventStatus after Dem_PreInit and before Dem_Init shall not trigger any monitor status change callback.⌋

7.9 OBD-specific functionality

7.9.1 General overview and restrictions

In the following, a specification of the OBD handling in AUTOSAR is introduced. Herein, “OBD“ is used for automotive OBD with respect to different target markets. For SW-sharing and distributed development reasons as well as aspects of packaging and responsibility of releases, the OBD-relevant information / data structures need to be reported via Standardized AUTOSAR interfaces.

In a vehicle there can be 3 different kinds of OBD ECUs:

- Master ECU (one per vehicle), in WWH-OBD [3] referenced as VOBD
- Primary ECU (several per vehicle)

- Dependent / Secondary ECUs (several per vehicle)

From the Basic Software point of view Dependent / Secondary ECUs doesn't need any specific OBD functionality. In Dependent / Secondary ECUs are always related to a Master or a Primary ECU. In Dependent / Secondary ECUs OBD-relevant information will not be stored in the Basic Software (e.g. OBD events will be forwarded to the respective Master or Primary ECU via the Bussystem). In Dependent / Secondary ECUs this "reported errors" and other OBD functionality might be handeled by a SW component.

The following table will give an overview about which OBD functionality must be supported in a Master ECU, Primary ECU or Dependent / Secondary ECU:

Functionality	***Master ECU***	***Primary ECU***	***Dependent / Secondary ECU***
OBD Event Memory	Own and "reported errors" of Dep. / Sec ECUs	Own and "reported errors" of Dep. / Sec ECUs	No
MIL Master	Yes	No	No
Readiness per vehicle (PID $90)	Yes	No	No
Readiness per ECU (PID $91)	Yes	Yes	No
Vehicle OBD counters (PID $93)	Yes	No	No
Calulation and provision of General Information (DYC, General Nominator, PFC cycle,..)	Yes	No	No
Reception and execution of General Information (DYC, General Nominator, PFC cycle,..)	(Yes)	Yes	No
Continuous-MI counter	Yes	No	No
Cumulative Continuous-MI counter	Yes	No	No
Calibration Identification (CAL-ID)	Not in BSW	Not in BSW	Not in BSW
Calibration Verification Number (CVN)	Not in BSW	Not in BSW	Not in BSW

Table 7.4: Overview about OBD functionality in different OBD ECUs

The following OBD requierments are only valid for Master and Primary ECUs. If necessary the OBD requierments differenciate between Master and Primary Requierment. Master and Primary ECUs should have the same interface to the SWComponents. To build a sufficant and lean Master ECU there is no compulsion to use this interfaces.

Some details on the interaction between Dem and specific SW-C might remain open, since they are dependent on the Dem and SW-C implementation. The following functionality is not defined:

- Malfunction Indicator Lamp (MIL)-activation (interface Dem to MIL handler, MIL-bulb check, readiness blinking, blinking in case of catalyst damaging misfire, Continuous-MI, Short-MI, On-demand MI,...)
- Misfire fault handling (common debouncing, filtering single / multiple misfire faults).

However, this Dem SWS does not prescribe implementation details on how OBD compliance can be achieved within the Dem module, e.g. concerning state handling. Furthermore, the Dem SWS does not prescribe implementation details on the diagnostic algorithms of the SW-C necessary to achieve OBD compliance (how to detect a fault, when to trigger incrementation of IUMPR-numerator...).

In the following chapters, OBD relevant functionality and interfaces are described. It is important to note, that independent of standard Autosar mechanisms (e.g. communication via RTE), response codes and timing constraints need to fulfill OBD requirements (refer to [12] and [17]).

[SWS_Dem_01248] ⌈The configuration parameter DemOBDEventMemorySetRef shall be used to reference to the one and only DemEventMemorySet that contains OBD relevant information. All OBD operations of the Dem shall be only executed on this DemEventMemorySet.⌋

[SWS_Dem_00584] ⌈While applying standard mechanisms (like Std_ReturnType), the Dem module shall only return values ensuring OBD compliant behavior with regard to permitted response codes and timing constraints.⌋

Calulation and provision of General Information Data:
The Master ECU shall calculate and provide the following General Information Data via the Bussystem to the Primary ECUs:

- OBD Driving cycle information (DYC)
- General Nominator / Rate-based monitoring - driving cycle (RBM cycle)
- Warm up cycle (WUC)
- Ignition cycle
- Qualified OBD Driving cycle
- Permanent fault code - driving cycle (PFC cycle)

OBD Driving cycle information (DYC)
The Master ECU will provide the driving cycle information (DYC) via the Busystem. The driving cycle information (DYC) shall not be computed internally in the Primary ECUs.

General Nominator / Rate-based monitoring - driving cycle (RBM cycle)
Included in the IUMPR-Cycle Flag

Warm up cycle (WUC)
The warm up cycle (WUC) is a legally prescribed cycle and is computed by the Master ECU. The Master ECU will provide the warm-up cycle information (WUC) via the Busystem.

Ignition cycle
An ignition cycle describes the cycle between the “Terminal 15 on“ status and “Terminal 15 off“, including the shut down/after-run phase of the electronic control unit if the

engine start condition was met for at least 2 seconds after “Terminal 15 on“. The Master ECU will provide the Ignition cycle information via the Busystem.

Qualified OBD Driving cycle
The Qualified OBD Driving cycle is a legally prescribed cycle and has the following definition: For combustion engines (no hybrid): A driving cycle starts with “ignition on“ and ends with the next time “ignition on“ if the engine has reached the first engine start conditions for at least 2 seconds (+/-a sec.) in between. Engine stops which are not caused by the driver or ECU don’t end the driving cycle.

For hybrids and engines with shutoff strategies, the driving cycle is defined as the time between “ignition on“ and the next time “ignition on“ after the driving cycle is qualified.

If a driving cycle is qualified every “ignition off > ignition on - EVENT“ results in the beginning of a new driving cycle. This new driving shall be qualified. A driving cycle is qualified when the engine speed has exceeded the minimum speed for engine start conditions for at least 2 seconds (+/-a sec.) or condition ready for driving for at least 2 seconds (+/-a sec.)

Permanent fault code - driving cycle (PFC cycle)
(refer to chapter 7.9.5.8)

Note: Calibration Identification (CAL-ID) and Calibration Verification Number (CVN)
The Calulation of the Calibration Identification (CAL-ID) and Calibration Verification Number (CVN) is not a BSW Task and will not handled within the Dem.

7.9.2 PIDs provided by Dem

[SWS_Dem_00293] ⌈For the following PIDs the data layout shall be compliant to SAE J1979DA [18].

- PID $01 monitor status since DTCs cleared (4 byte)
- PID $02[1] DTC that caused required freeze frame storage (2 byte)
- PID $1C OBD requirements to which vehicle or engine is certified (1 byte)
- PID $1E auxiliary input status (`PTO` status) (1 byte)
- PID $21 distance traveled while MIL is activated (2 byte)
- PID $30 number of warm-ups (WUC) since DTCs cleared (1 byte)
- PID $31 distance traveled since DTCs cleared (2 byte)
- PID $41 monitor status this driving cycle (4 byte)

[1] PID $02 is only required for service $02 and therefore no interface (like Dem_DcmReadDataOfPID02) is necessary. Instead the API `Dem_DcmGetDTCOfOBDFreezeFrame` is used (refer to[SWS_Dem_00623])

- PID $4D engine run time while MIL is activated (2 byte)
- PID $4E engine run time since DTCs cleared (2 byte)
- PID $91 ECU OBD System Information (5 byte)
- PID $F501 I/M Readiness Data

⌋

Note: PID $90 and $93 are calculated in the application of the VOBD. Primary ECUs only provides interfaces to retrieve the related data.

[SWS_Dem_CONSTR_06150] Dependency on container DemPidClass ⌈The container DemPidClass and aggregated sub-container shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_00351] ⌈The Dem module shall compute and provide the number of confirmed faults (PID $01, Byte A).⌋

[SWS_Dem_01137] ⌈For WWH-OBD PID$01 shall not provide the number of confirmed faults.⌋

[SWS_Dem_01138] ⌈The Dem module shall compute and provide the ECU MIL status (PID $01, Byte A).⌋

[SWS_Dem_01139] ⌈For WWH-OBD PID$01 shall not provide the ECU MIL status.⌋

[SWS_Dem_00748] ⌈The function Dem_DcmReadDataOfPID1C shall return the appropriate value “OBD requirements to which vehicle or engine is certified.“ according to the respective standards [12], e.g. OBD, OBDII, JOBD etc. The value PID1Cvalue to return is in configuration parameter DemOBDCompliancy defined.⌋

[SWS_Dem_01368] Support of PID 0xF501 data

Upstream requirements: RS_Diag_04252

⌈The Dem shall calculate the data for PID 0xF501 and provide the data within the parameter PIDF501value of Dem_DcmReadDataOfPIDF501.⌋

7.9.2.1 Centralized PID $21 / $31 / $4D / $4E handling

[SWS_Dem_00703] ⌈The Dem of the Master ECU shall calculate information for PID $31 / $4D / $4E vehicle wide.⌋

A Primary ECU is not allowed to calculate information for PID $31 / $4D / $4E.

Note: Therefore a Software component of the Master ECU will use API Dem_DcmReadDataOfPID<NN> (where NN is 31 / 4D / 4E) to read out the information and will provide information about mileage / time for PID $31 / $4D / $4E to the Primary ECU via the bussystem.

On receiving the information for PID $31 / $4D / $4E in a software component in the Primary ECU via the bussystem, the software component will set the PID $31 / $4D / $4E in calling the API Dem_SetDataOfPID<NN> (where NN is 31 / 4D / 4E) of the Dem of the Primary ECU.

[SWS_Dem_00704] ⌈Only the Master ECU is allowed to report information for PID $31 / $4D / $4E to the scan tool.⌋

Primary ECUs are not allowed report informations for PID $31 / $4D / $4E to the scan tool.

On receiving the information for PID $21 in a software component in the Primary ECU via the bussystem, the software component will set the PID $21 in calling the API Dem_SetDataOfPID21 of the Dem of the Primary ECU.

[SWS_Dem_00346] ⌈The Dem module shall use PID $0D to calculate PID $21 and PID $31.⌋

For details see chapter 7.11.7.

[SWS_Dem_00304] ⌈A Dem delivery shall provide function call interfaces to the Dcm and the respective ServiceNeeds to declare to the Dcm that these PIDs are supported.⌋

A special configuration is applied for the computation of PID $01 and $41:

[SWS_Dem_00349] ⌈The Dem module shall support the configuration parameter DemEventOBDReadinessGroup for OBD systems, to assign individual events to one specific readiness group.⌋

According to SAEJ1979 [17], the group AirCondition Component shall not be supported anymore. However, it is still included in ISO 15031-5 [12]. The groups distinguish between spark ignition engines (spark) and compression ignition engines (compr.). However, it is necessary to configure per event, to which readiness group the monitor contributes (if at all).

PID $21 handling

PID $21 is either reported only by the OBD Master ECU, or a synchronized valued by all OBD ECUs (centralized PID handling).

If the configuration switch DemOBDCentralizedPID21Handling is enabled:

A Software component of the OBD Master ECU could retrieve the current PID $21 value via the interface GetDataOfPID21 from Dem module with the goal to distribute the value to all other Primary ECUs in the vehicle.

A Software component of the OBD Primary ECU receiving the information for PID $21 could forward this information to the Dem module via the interface SetDataOfPID21 to synchronize the value with the OBD Master ECU.

[SWS_Dem_01095] ⌈On invocation of Dem_SetDataOfPID21 the internal PID $21 value shall be updated.⌋

Note: AUTOSAR defines that OBD Primary ECUs do not calculate the PID $21 value by its own.

[SWS_Dem_01096] ⌈An OBD Master ECU shall calculate the PID $21 value by its own (considering [SWS_Dem_00346]).⌋

[SWS_Dem_01097] ⌈On invocation of Dem_GetDataOfPID21 the internal calculated PID $21 value shall be returned.⌋

Note: AUTOSAR defines that OBD Primary ECUs do not calculate the PID $21 value by its own.

[SWS_Dem_01098] ⌈On invocation of Dem_DcmReadDataOfPID21 the Dem shall return the current value of PID $21.⌋

[SWS_Dem_01099] ⌈If Dem_DcmReadDataOfPID21 is called before Dem_SetDataOfPID21, the Dem shall return 0xFFFF as PID $21 value.⌋

7.9.2.2 PID $1E handling

The power take-off (PTO) mode is a mode in which the vehicle is not driven on the road and the engine powers auxiliary equipment, such as mowers or generators. In active PTO mode, some OBD monitors cannot be executed, and monitor results could otherwise lead to incorrect OBD states. Details on PTO are available in CCR 1971.1 [19].

The PTO state is reported to the Dem via Dem_SetPtoStatus and maintained by the Dem. The Dem provides the PTO state as part of PID $1E to the Dcm via Dem_DcmReadDataOfPID1E, which can be mapped to the Dcm read function for a configured PID $1E.

For monitors that consider the PTO state, the Dem may perform PTO-specific calculations (e.g., event status and MIL status) in accordance with CCR 1971.1. AUTOSAR does not specify these calculations in detail. The Dem provides a general switch to enable PTO support, a per-DTC configuration to identify DTCs affected by PTO, and an API for the application to report the current PTO state.

[SWS_Dem_00347] PID $1E support based on PTO support ⌈If DemPTOSupport is set to true, the Dem shall provide PID $1E data representing the current PTO state via Dem_DcmReadDataOfPID1E.⌋

[SWS_Dem_00377] PID $1E value according to reported PTO state ⌈If DemPTOSupport is set to true and Dem_SetPtoStatus is called with PtoStatus, the Dem shall use PtoStatus as the current PTO state for the PID $1E value returned by Dem_DcmReadDataOfPID1E.⌋

[SWS_Dem_01431] Initial PTO state for PID $1E ⌈If DemPTOSupport is set to true, after Dem_Init and until the first call to Dem_SetPtoStatus, the Dem shall use the PTO state inactive.⌋

[SWS_Dem_00378] DTC processing for PTO-affected DTCs ⌈If DemConsiderPtoStatus is set to true for a DTC and the current PTO state is active, the Dem shall process the DTC in accordance with CCR 1971.1 [19].⌋

7.9.3 Readiness status

[SWS_Dem_00354] ⌈The Dem module shall compute for PID $01 the readiness status (if all events of a DemEventOBDReadinessGroup are reported as OK tested since last clear, or the event has caused MIL on). Suppressed Events shall be ignored for this computation.⌋

For details see chapter 7.4.8.

Note: DemEventOBDReadinessGroup has a regulation background and special computations based on ModelYear (MY) and Market.

[SWS_Dem_CONSTR_06147] Limit readiness to OBD ECUs only ⌈DemEventOBDReadinessGroup shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

[SWS_Dem_00355] ⌈The Dem module shall compute for PID $41 the readiness group complete for current driving cycle (if all events of a group are tested in the current driving cycle). OBD Events Suppression shall be ignored for this computation.⌋

For details see chapter 7.4.8.

Note: For calculation of the group readiness, the Dem module considers all events assigned to the specific readiness group.

[SWS_Dem_00356] ⌈The Dem module shall compute the readiness group disabled (if the disabled status is reported by the monitor for any event of a group). OBD Events Suppression shall be ignored for this computation.⌋

For details see chapter 7.4.8.

[SWS_Dem_00348] ⌈The Dem module shall provide the disabling of events (refer to Dem_SetEventDisabled). OBD Events Suppression shall be ignored for this computation.⌋

For details see chapter 7.4.8.

[SWS_Dem_00294] ⌈In order to allow a monitor to report that the event cannot be computed in the driving cycle (aborted e.g. due to physical reasons), the Dem shall provide the API Dem_SetEventDisabled.⌋

Note: For the computation of PID $41, the monitor has to report its event as disabled, if the test cannot be carried out anymore until the end of this driving cycle.

Note: SetEventDisabled does report an Event as “uncompletable“ during the current driving cycle. It does not turn the Event into a Events Suppression according to chapter 7.4.8

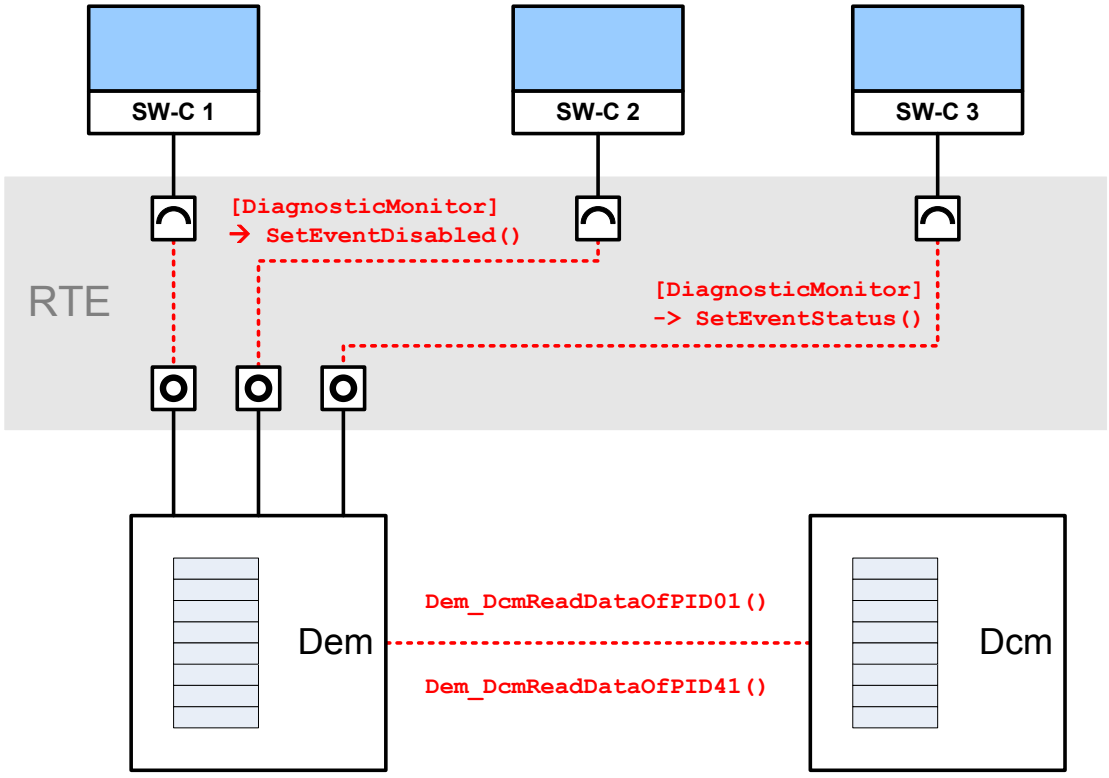


Figure 7.43: Dem calculates PID$01 and PID$41 data based on specific port operations

7.9.4 In-Use-Monitor Performance Ratio (IUMPR) Support

The In-Use-Monitor Performance Ratio (IUMPR) indicates how often the OBD system monitors particular components, compared to the amount of the vehicle operation. It is defined as the number of times a fault could have been found (=numerator), divided by the number of times the vehicle operation conditions have been fulfilled (=denominator) as defined in the respective OBD regulations.

The IUMPR will be calculated within the ECU using the following formula: In-Use-Monitor Performance Ratio = NUMERATOR / DENOMINATOR

[SWS_Dem_00709] ⌈Ratios which refer to a Suppressed Event shall not be computed.⌋

[SWS_Dem_00710] ⌈The numerator shall be calculated in the ECU (Master and Primary).⌋

[SWS_Dem_00711] ⌈The Dem of the Master ECU shall calulate the IUMPR-Cycle Flag.⌋

[SWS_Dem_00966] ⌈The Dem shall provide an API Dem_GetIUMPRDenCondition to give a software component the possibility to get the General Denominator status information.⌋

The Master ECU (software component) will provide the IUMPR-Cycle Flag (included in the General Denominator signal) via the bus system.

[SWS_Dem_00712] ⌈In master and optionally primary OBD ECUs, the Dem shall increment ECU global internal general denominator depending on the denominator DEM_IUMPR_GENERAL_OBDCOND, if available. Depending on the system definition the global general denominator (and therefore the DEM_IUMPR_GENERAL_OBDCOND) may not be needed in some or even all primary ECUs. In such cases the DEM_IUMPR_GENERAL_OBDCOND is never set, and the general denominator is stuck at 0 (and reporting of the InfoType $08 / $0B is disabled in the Dcm).⌋

[SWS_Dem_01379] ⌈If DemIUMPRDenGroup is configured as DEM_IUMPR_DEN_NONE, the Dem shall increment the denominator of the ratio depending on DEM_IUMPR_GENERAL_INDIVIDUAL_DENOMINATOR a maximum of once per driving cycle.⌋

[SWS_Dem_01380] ⌈If DemIUMPRDenGroup is configured as DEM_IUMPR_DEN_500MILL, the Dem shall increment the denominator of the ratio depending on DEM_IUMPR_DEN_COND_500MI_CMPL a maximum of once per driving cycle.⌋

[SWS_Dem_01381] ⌈If DemIUMPRDenGroup is configured as DEM_IUMPR_DEN_COLDSTART, the Dem shall increment the denominator of the ratio depending on DEM_IUMPR_DEN_COND_COLDSTART_CMPL a maximum of once per driving cycle.⌋

[SWS_Dem_01382] ⌈If DemIUMPRDenGroup is configured as DEM_IUMPR_DEN_EVAP, the Dem shall increment the denominator of the ratio depending on DEM_IUMPR_DEN_COND_EVAP_CMPL a maximum of once per driving cycle.⌋

[SWS_Dem_01383] ⌈If DemIUMPRDenGroup is configured as DEM_IUMPR_DEN_PHYS_API, the Dem shall increment the denominator of the ratio depending on DEM_IUMPR_GENERAL_INDIVIDUAL_DENOMINATOR a maximum of once per driving cycle and shall consider monitor specific conditions only for the cycle the Release operation was called in.⌋

[SWS_Dem_01394] ⌈If DemIUMPRDenGroup is configured as DEM_IUMPR_DEN_CSERS the Dem shall increment the denominator of the ratio depending on DEM_IUMPR_DEN_COND_CSERS a maximum of once per driving cycle.⌋

[SWS_Dem_01393] ⌈If DemIUMPRDenGroup is configured as DEM_IUMPR_DEN_CSERS_API the Dem shall increment the denominator of the ratio depending on DEM_IUMPR_DEN_COND_CSERS a maximum of once per driving cycle and shall consider monitor specific conditions only for the cycle the Release operation was called in.⌋

[SWS_Dem_01396] ⌈If DemIUMPRDenGroup is configured as DEM_IUMPR_DEN_EVAPPURGEFLOW, the Dem shall increment the denominator of the ratio depending on DEM_IUMPR_DEN_COND_EVAPPURGEFLOW_CMPL a maximum of once per driving cycle.⌋

[SWS_Dem_00714] ⌈The Dem shall provide an API Dem_SetIUMPRDenCondition to get informed about the IUMPR-Cycle Flag status by a software component.⌋

Note: A Timeout of the IUMPR-Cycle Flag / General Denominator signal of the Master will lead to an event memory entry in the COM Stack (timeout of signal) in the Primary ECU.

Further details concerning In-Use-Monitor Performance Ratio (IUMPR) can be found in regulation documentation.

Differencation InfoType $08 / InfoType $0B

The IUMPR-data collected need to be provided upon a service $09 request. For gasoline engines, the Info Type $08 is used and for diesel engines the Info Type $0B is used (refer to [12] and [17]).

[SWS_Dem_00298] ⌈In order to support the data requests in service $09 as described above, the Dem shall provide the API Dem_DcmGetInfoTypeValue08 or Dem_DcmGetInfoTypeValue0B to the Dcm.⌋

[SWS_Dem_00357] ⌈If the DemOBDEngineType is set to DEM_IGNITION_SPARK the Dem module shall provide the API Dem_DcmGetInfoTypeValue08 for InfoType $08 IUMPR data.⌋

Note: The service Dem_DcmGetInfoTypeValue08 will be used by the Dcm, to request the IUMPR-data according to the InfoType $08 data format used for the output to service $09.

[SWS_Dem_00358] ⌈If the DemOBDEngineType is set to DEM_IGNITION_COMPRESSION the Dem module shall provide the API Dem_DcmGetInfoTypeValue0B for Info Type $0B IUMPR data.⌋

Note: The API Dem_DcmGetInfoTypeValue0B will be used by the Dcm, to request the IUMPRdata according to the InfoType $0B data format used for the output to service $09.

The type of OBD system is defined by using the configuration switch DemOBDSupport (refer to DemGeneral).

In-Use-Monitor Performace Ratio (IUMPR) groups or components
The relevant In-Use-Monitor Performace Ratio (IUMPR) data recording is allocated in the Dem based on FIDs and events. The IUMPR data are recorded for different groups or components respectively.

Typically, one or more events serve for the monitoring of these components, e.g. the oxygen sensor. Hence, in order to record the in-use performance of the OBD-system, the test performance of all the relevant events for all the IUMPR-groups needs to be recorded. For that purpose, certain data structures need to be configured by the Dem.

However, in order to stop the incrementing of numerator and denominator in the case a monitor is stopped, due to the occurrence of another event preventing the monitor from running, it is necessary to list all events that can affect the computation of a particular IUMPRrelevant event. However, this information is already embodied in a FID (refer to FiM SWS) representing a function which serves for the computation e.g. of a particular even. A FID is inhibited in case of certain events and thus, this relation can also be used to stop the IUMPR record.

This leads to a combination of an event to be recorded and a primary FID (and optionally multiple secondary FIDs) representing all the events stopping the computation of the IUMPR-event. For the purpose of classifying the events into groups, an IUMPR-group is also part of this combination.

For the configuration of data structure per EventId / FID(s) / IUMPR-group combination, a new data object is introduced, namely, the RatioId (refer to DemRatioId). Thus, the container DemRatio contains the EventId, primary FID, secondary FIDs, IUMPR-group and an interface option to configure “APIuse“ vs. “observer“, whereas the interface option is explained in more detail in the following.

Also for the purpose of the port configuration, an ObdRatioServiceNeeds is offered to the SW-C. If a SW-C is IUMPRrelevant, this ServiceNeed is filled out.

If the monitor is “symmetric“, i.e. upon completion of the test a fault could have been found, even if there is currently no fault in the system, then the numerator can be incremented just by observing the TESTED-status of the assigned event.

[SWS_Dem_00359] ⌈Only for monitors being configured with the option “observer“, the Dem module shall increment the numerator of the corresponding monitor, if the assigned event gets tested/qualified (as passed or failed).⌋

If the diagnostic is asymmetric and it takes more efforts to detect a malfunction than to detect an OK status, the monitor needs to call an API in order to report that a malfunction could have been found, because this may require some simulation within the monitor and can therefore not be purely derived from the TESTED status.

[SWS_Dem_00296] ⌈The Dem module shall provide the API Dem_RepIUMPRFaultDetect for the asymmetric monitor to report that a malfunction could have been found.⌋

Note: This service shall be used by the monitor to report that a fault could have been found even if there is currently no fault at all according to the IUMPR regulations that all conditions are met for the detection of a malfunction.

[SWS_Dem_01188] ⌈The Dem shall increment the ratio-individual IUMPR-numerator at maximum of one time per OBD driving cycle.⌋

[SWS_Dem_00361] ⌈The Dem module shall provide the configuration parameter DemRatioKind, to indicate per RatioId if the numerator is calculated based on the TESTED-status or the API call.⌋

Additional denominator conditions:
For some particular monitors (e.g. for secondary air system, comprehensive components), there are additional conditions defined on the denominator: Their denominator will only be incremented if certain temperature or activity conditions are met. Then, the monitor needs unlock (Release) the denominator when the additional conditions are met in a driving cycle.

In case of a fault being detected that prevents further computations of these additional denominator conditions, the numerator is required to be frozen as well. For that reason it is necessary to assign particular denominator conditions if applicable. Upon the report of an inhibited computation of a particular denominator condition, the affected ratio are set to frozen,

For this handling of commonly used denominator conditions within the Dem, further interfaces and configuration items are introduced.

[SWS_Dem_00715] ⌈The Dem module shall provide configuration parameter DemIUMPRDenGroup to offer several conditions to be applied to the denominator per RatioId.⌋

[SWS_Dem_00362] Release of IUMPR Denominator ⌈If DemOBDSupport is not set to DEM_OBD_NO_OBD_SUPPORT, the Dem module shall provide the APIs for releasing (refer to Dem_RepIUMPRDenRelease) the denominator under special conditions if the DemIUMPRDenGroup is configured as DEM_IUMPR_DEN_PHYS_API or DEM_IUMPR_DEN_CSERS_API.⌋

[SWS_Dem_00308] ⌈The Dem shall provide the API Dem_RepIUMPRDenRelease to IUMPRrelevant SW-C, to control the denominator specific to the respective RatioId.⌋

Note: This service shall be used by the monitor to report, that a denominator of a specific monitor (represented by FID and EventId) is released for physical reasons.

In systems with multiple ECUs being OBD relevant and even IUMPR relevant, the status information on the additional denominator conditions needs to be communicated

to synchronize the behavior of the counters. For that reason, an additional interface is introduced to read out and to report the status of the individual denominator conditions.

[SWS_Dem_00716] ⌈The Dem shall provide the API `Dem_GetIUMPRDenCondition` to read out the status of a particular condition.⌋

[SWS_Dem_00717] ⌈The Dem shall provide the API `Dem_SetIUMPRDenCondition` to set the status of a particular condition (mainly in depending ECUs).⌋

Note: To comuincate the received general denominator status, the Primary ECU should make use existing API as offered for SWCs.

Legislation requires that IUMPR tracking shall be stopped for a specific monitor, if it is inhibited by another service $07 visible fault.

[SWS_Dem_01392] ⌈The IUMPR ratio shall be blocked whenever any of the Events linked (via FunctionInhibitionConfiguration or through SummaryEvents) to one of the FIDs referenced by `DemFunctionIdRef` is in state PENDING, or any other event causal to this event (according to component dependency structure) is in state PENDING (see `DemComponentClassRef` referenced by `DemDiagnosticEventRef`).⌋

[SWS_Dem_00299] ⌈As long as an event has Pending status, the Dem module shall stop increasing the numerator and denominator.⌋

Based on the related FID (FiM access) and RatioId, the Dem module can determine which numerator and denominator has to be stopped.

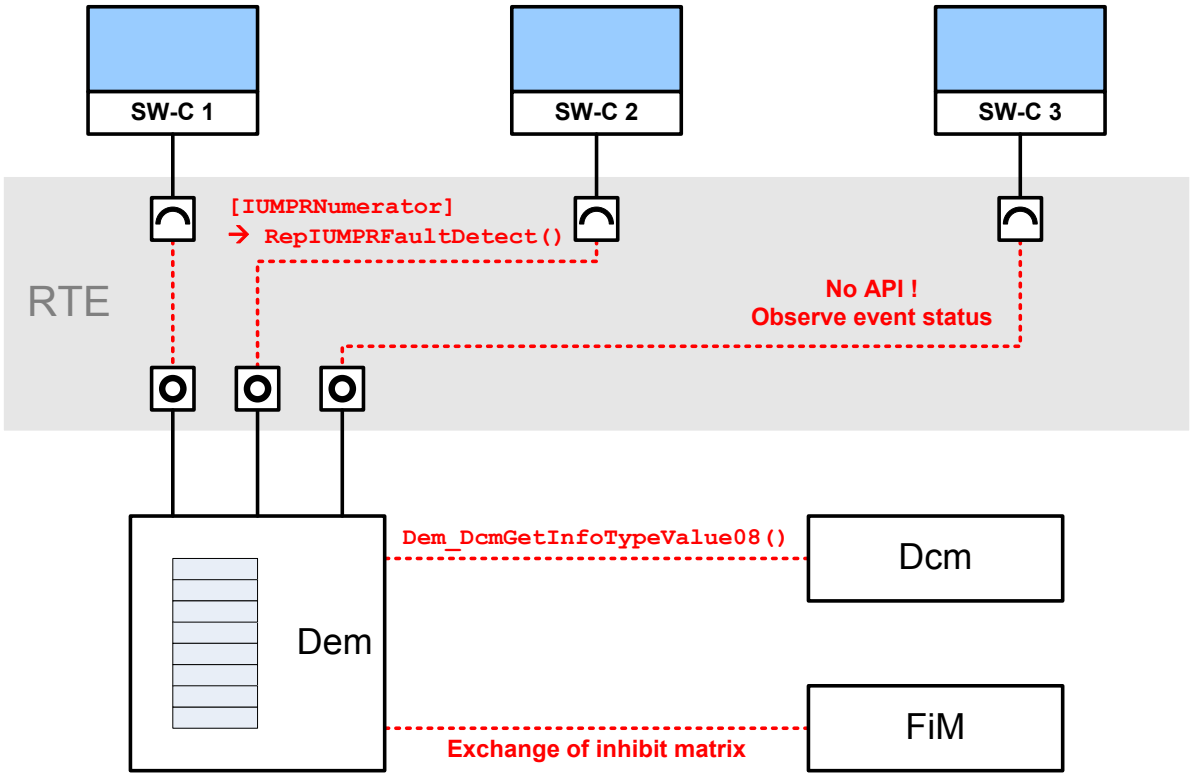


Figure 7.44: Dem calculates IUMPR data based on specific interface operations

[SWS_Dem_01369] Support of Monitor Activity Denominator

Upstream requirements: RS_Diag_04252

⌈The Dem shall calculate the data for ITID "Monitor Activity Denominator" and provide it in the parameter of `DataValueBuffer` of the API `Dem_DcmGetInfoTypeValue79`.⌋

[SWS_Dem_01297] Freeze monitor activity numerator if the general denominator is inhibited

Upstream requirements: RS_Diag_04252

⌈If `DemFreezeMANWhenGeneralDenominatorInhibited` is set to true and the general denominator is inhibited via `Dem_SetIUMPRDenCondition` reporting a status set to `DEM_IUMPR_DEN_STATUS_INHIBITED`, all monitor activity numerator (and implicitly denominator) values shall be frozen throughout the end of the current driving cycle.⌋

7.9.5 OBD for light duty (OBD2)

Together with the Dcm, this Dem SWS provides standardized AUTOSAR interfaces to support the OBD services $01 - $0A defined in SAE J1979 [17]. With these services, Autosar OBD functionality shall be capable of meeting all light duty OBD regulations worldwide (California OBDII, EOBD, Japan OBD, and all others.)

[SWS_Dem_CONSTR_06148] Dependency on container DemRation ⌈The container `DemRatio` shall only be available if `DemOBDSupport` is set to `DEM_OBD_MASTER_ECU`.⌋

7.9.5.1 Service $01 Read Current Powertrain Diagnostic Data

With Service $01, the current value of any PID (except PID $02) can be read. For each PID calculated by the Dem, a dedicated API function Dem_DcmReadDataOfPID<NN> as described in chapter 7.9.2 is provided.

7.9.5.2 Service $02 Read Powertrain Freeze Frame Data

In order to retrieve relevant data a fault entry, the Dem needs access to current data, addressed via PIDs. For that purpose, a Client (=Dcm/Dem)/Server (=SW-C) interface is assigned based on configuration items.

[SWS_Dem_00291] ⌈The Dem module shall support only the legislative freeze frame (record number 0). This will be a single list of PIDs assigned to this freeze frame (refer to `DemPidClass`).⌋

This means a request by a generic scan tool for record number one and above will be ignored.

Upon the entry of a fault in the memory, the values of these PIDs/DIDs are requested through the RTE via a client server interface.

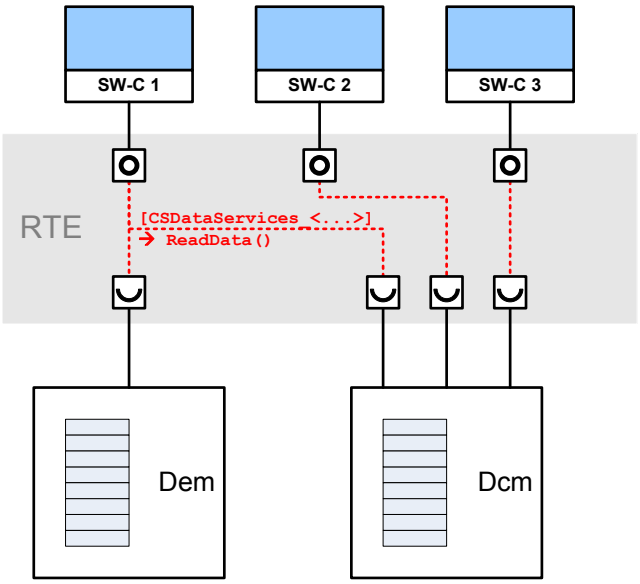


Figure 7.45: Dem and Dcm module requests PID data elements of SW-C via ReadData operation

[SWS_Dem_00596] ⌈The Dem module shall provide access on PID data elements of the most important freeze frame being selected for the output of service $02 (OBD freeze frame of the event which caused MIL on) to the Dcm module (refer to `Dem_DcmReadDataOfOBDFreezeFrame`).⌋

Note: The Dem processes and stores only the raw data for service $02. Any PID header-information and fill bytes are added by the Dcm. The Dcm configuration defines the individual PID layout, which also defines an order of the contained data elements. Each data element references to its linked element in the Dem configuration (refer to Figure 47).

The individual data elements of a PID are selected via an index by the Dcm.

[SWS_Dem_00597] ⌈The index values (refer to API parameter DataElementIndexOfPID) of `Dem_DcmReadDataOfOBDFreezeFrame` shall be assigned zerobased and consecutive. Their order shall be derived from the position of the data elements (refer to DcmDspPidDataPos, of the referencing DcmDspPidData containers) in the Dcm's PID layout.⌋

Note: This index is not configured explicitly (to be able to avoid resource overhead for e.g. mapping tables in the implementation).

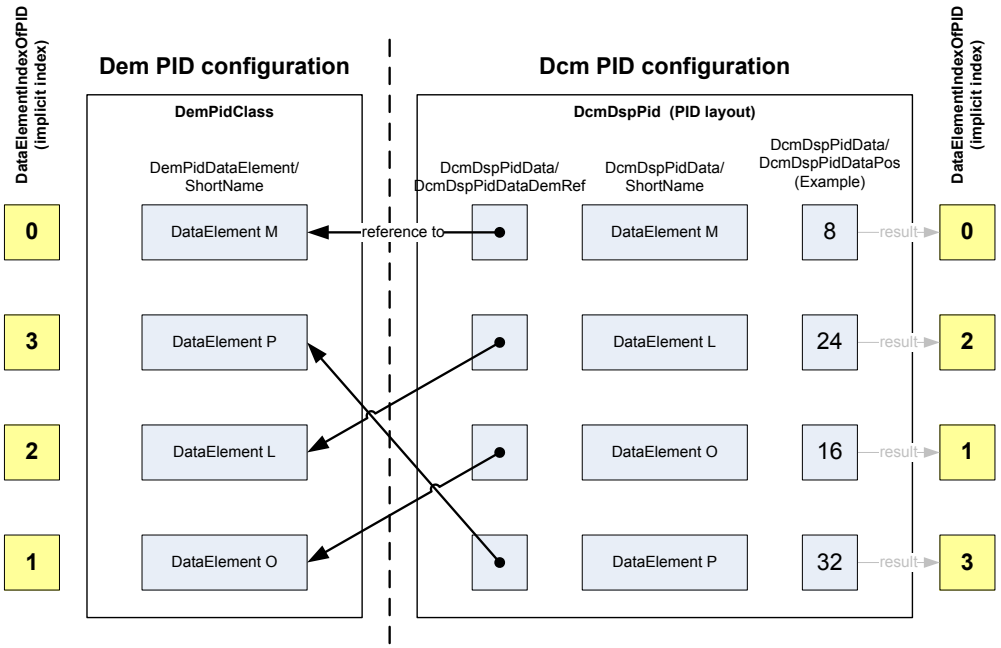


Figure 7.46: PID layout configuration within Dcm (master) and Dem

[SWS_Dem_00623] ⌈The function `Dem_DcmGetDTCOfOBDFreezeFrame` shall return the DTC associated with the most important freeze frame being selected for the output of service $02 (PID $02).⌋

7.9.5.3 Service $03 - Read Emission-Related Powertrain Diagnostic Trouble Codes

The diagnostic service provides the OBD DTCs where the `UDS status bit 3` is currently set. Therefore the interfaces described in chapter 8.3.3.1 will be used by `Dcm`.

7.9.5.4 Service $04 - Clear Reset Emission-Related Diagnostic Information

[SWS_Dem_00718] Clearing emission related DTCs ⌈In Master, Primary and `Dependent / Secondary ECUs` executing service $04 shall clear all emission-related diagnostic information in primary memory.⌋

[SWS_Dem_00719] ⌈Executing service $04 shall include non emissionrelated diagnostic information of primary event memory in Master, Primary and `Dependent / Secondary ECUs`.⌋

7.9.5.5 Service $06 - Support of central DTR handling

With Service $06, an external test tool can request an emissionrelated ECU to return the supported OBDMIDs (“OBD Monitor Identifier”) and / or the standardized diagnostic test results associated with the OBDMID.

The OBD related standards (refer to ISO 15031-5 [12] and SAE J1979 [17]) reserve certain OBDMIDs for the special purpose of obtaining the list of supported OBDMIDs within the vehicle. These OBDMIDs are called “availability OBDMIDs” and are numbered with $00, $20, $40, $60, $80, $A0, $C0 and $E0.

In contrast to earlier Autosar releases, the Dem additionally provides interfaces to report latest test results handled per particular identifier (DTR “Diagnostic Test Result“) introduced for that purpose.

Within the Service $06, the response (apart from the support information) consists of a data triple with test result, lower and upper limit representing the latest result of a conducted and qualified monitoring check. In order to refer to a particular component and a particular certified test, the response also contains the OBDMID and the TID (“Test Identifier“). For the external representation of the data, a socalled UnitandScalingID (UaSID) is passed along with the response onto the tester. From OBD regulation point of view it is required to have consistent data in Service$06 and Service$07. And in this scenario, a test has failed if the test result is either less than the lower limit or higher than the upper limit. Along with this violation of the thresholds, a corresponding Event has to be reported and vice versa. Note that this consistency is valid strictly only upon first detection of the malfunction, since a healing from Service$07 takes one entire cycle without a failure detected while the Service$06 contents simply represent the latest test result.

In order to support this consistency requirement, the Dem provides a specific interface to report the results per DTR whereas an Event is associated to a DTR on configuration level to refer to.

Note that the Event serves as master and within its status additional effects such as enable / storage conditions are encountered. Given a specific DTR update type (by configuration), the UDS status (Tested, Failed, debounce status etc.) decides whether a DTR result being reported is further processed or suppressed/ignored.

The Dem offers a service of conversion for the SWC to support a report of test results and thresholds within the ECU-internal resolution. The conversion onto the target format (standardized by [17, 19]: uint16, formula according to UaSID) including a correction in case of rounding effects is done within the Dem. For that reason the configuration includes a conversion formula similar to the one used by the RTE to describe physical data (but simpler) where only a linear mapping is supported. Note that the conversion does not respect the UaSId. The service Need for the DTR provides a reference to a physical unit (see SWC Template chapter “Physical Units“) and a linear conversion (see SWC Template chapter “Computation Methods“) of CompuRationalCoeffs with a maximum of two numerator coefficients and one denominator coefficient,

as well as the desired UaSId During the configuration process, this information is processed into a single linear conversion, which is then stored inside the ECUC of the Dem.

By this central data handling, the aspect of data consistency for Service$07 / $06 is supported as well as reset in case of fault clear.

[SWS_Dem_00751] ⌈The Dem shall provide data structures and functionality to receive, store and report Service$06 test results persistently.⌋

[SWS_Dem_00752]

Upstream requirements: RS_Diag_04129

⌈The `DemDtrId` is assigned by the `Dem` during the `BSW` configuration step.⌋

Note: The Dem refers to a BSW configuration `DemDtr` which associates

- a `DemDtrId` to an EventId
- an indication whether the Monitor will provide data for the minimum limit, maximum limit, or both
- OBDMID
- TID
- UaSID
- `DemDtrUpdateKind`
- Coefficients for linear conversion analogous to the SW-C Description (chapter 5.5.)

[SWS_Dem_00753] ⌈For ensuring data consistency, an EventId can only be referenced by at maximum one `DTR`.⌋

[SWS_Dem_00754] ⌈The `Dem` shall use the `DemDtrUpdateKind` for the evaluation and processing of reported `DTR`-values:

- `DEM_DTR_UPDATE_ALWAYS`: State of the event is not considered (default, and used if `DemDtrEventRef` is not configured).
- `DEM_DTR_UPDATE_STEADY`: Only when the events pre-debouncing within `Dem` is “stuck“ at the FAIL or PASS limit, and the latest result matches the debouncing direction.

⌋

[SWS_Dem_00755] ⌈The Dem shall only support a linear mapping, i.e. only support values for the coefficients b, c, and f.⌋

Note: ASAM formula coefficients: $(ax^2 + bx + c) / (dx^2 2 + ex + f)$ supporting on b, c and f.

[SWS_Dem_00756]

Upstream requirements: RS_Diag_04179

⌈The Dem shall provide an API Dem_SetDTR within an interface DTRCentralReport to provide a report mechanism to the SWC for their test results. Arguments shall be the DemDtrId, the test value, the minimum and maximum limit, and a state indicating whether the monitor provides a valid minimum limit, maximum limit, or both, or whether the reported values shall be reset to zero.⌋

[SWS_Dem_00757] ⌈If DemDtrEventRef references a DemEventParameter, the Dem shall process the reported DTR values only if all the enable/storage conditions of the referenced event are fulfilled, independent of DemDtrUpdateKind. If any of the enable/storage conditions of the referenced event are not fulfilled, then the Dem shall ignore the reported DTR values.⌋

Note the example with DEM_DTR_UPDATE_STEADY: If an event has reached that status TESTED starting from an initialized state but has not detected a FAILED, the received DTR values within the limits are processed (=converted and stored). If the test result violates its associated limit (cf. control value) while the eventis indeed TESTED and FAILED, the DTR report is processed as well. But if the test result violates its associated limit (cf. control value) while the event is TESTED and PASSED, the DTR report is ignored assuming that the EventId is also not updated for good reasons (cf. enable / storage conditions).

Applying this validation check within the DTR report, the data consistency in Service$06 and $07 is ensured.

[SWS_Dem_00758] ⌈The Dem shall support a conversion of the ECUinternal resolution into the standardized external uint16 size and resolution according to the configured and standardized UaSID. For that reason, the conversion formula is configured per DTR. It is assumed that the conveyed parameter of the API call contain the values in the internal size and resolution. The Dem does offer sint32 as parameter for that reason to cover uint16 as well as sint16 (or even sint32) internal data types.⌋

[SWS_Dem_00759]

Upstream requirements: RS_Diag_04179

⌈The Dem shall ensure the validity of the test results vs. threshold(s). If the parameter TestResult passed in Dem_SetDTR is inside/outside the thresholds given by UpperLimit and LowerLimit, Testvalue of Dem_DcmGetDTRData shall also be inside/outside Upplimvalue and Lowlimvalue. If rounding effects lead to an unintended change of the meaning, the thresholds shall be shifted by one increment accordingly to restore the result.⌋

[SWS_Dem_00760] Computing available OBD MID values ⌈If Dem_DcmGetAvailableOBDMIDs is called with an "availablity Obdmid" value of 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0 or 0xE0 and the Dem has configured DemDtrMid, the Dem

shall compute and provide the “Supported-OBDMID” information in the OUT parameter Obdmidvalue and return E_OK. The computed Obdmids parameter is according to ISO 15031-5[12] and sets a bit for each supported Obdmid in the request range (starting with Obdmid+1) and having the last bit of the Obdmidvalue set, if any subsequent range has configured Obdmids.⌋

[SWS_Dem_01301] Behavior on invalid Obdmid value ⌈If Dem_DcmGetAvailableOBDMIDs is called with an invalid Obdmid value which is different to 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0 or 0xE0, the Dem shall report the Det development error DEM_E_INVALID_OBDMID.⌋

[SWS_Dem_00761] ⌈Upon request by the Dcm, the Dem shall repond with the number of TIDs per requested OBDMID using the API Dem_DcmGetNumTIDsOfOBDMID. This value can be used by the Dcm to iteratively request for the DTR data per OBDMID / TID-index whereas the TID-index loops from 0 to number-of-TIDs minus one.⌋

[SWS_Dem_00762] ⌈Upon request by the Dcm, the Dem shall respond with the data available for a particular OBDMID / TIDindex per requested OBDMID using the API Dem_DcmGetDTRData. This value can be used by the Dcm to iteratively request for the DTR data per OBDMID / TIDindex starting from 0 to numberofTIDs minus one.⌋

[SWS_Dem_00763] ⌈The Dem shall reset the DTR data using the reference to the Event per DTR, whenever the Event is affected by a fault clear command.⌋

[SWS_Dem_00764] ⌈If no Event is assigned per DTR, the Dem shall reset the DTR data upon a Service$04 clear command by the Dcm, i.e. a Dem_ClearDTC with:

DTC = DEM_DTC_GROUP_ALL_DTCS

DTCFormat = DEM_DTC_FORMAT_OBD

DTCOrigin = DEM_DTC_ORIGIN_PRIMARY_MEMORY

⌋

[SWS_Dem_CONSTR_06149] Dependency on container DemDtr ⌈The container DemDtr shall only be available if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU.⌋

A DTR can refer to an event via DemDtrEventRef. The referenced event can be unavailable according to chapter chapter 7.4.8. In this case the Dem treats the DTR as not supported and will neither compute the DTR or report a result of it.

[SWS_Dem_01335] Setting DTR results for unavailable events

Upstream requirements: RS_Diag_04126

⌈If Dem_SetDTR is called for an DTR that references via DemDtrEventRef an event that itself is configured with DemEventAvailable to FALSE, the Dem shall return E_NOT_OK and not compute the DTR.⌋

[SWS_Dem_01336] Reporting DTRs for unavailable events

Upstream requirements: RS_Diag_04126

⌈If Dem_DcmGetDTRData is called for an DTR that references via DemDtrEventRef an event that itself is configured with DemEventAvailable to FALSE, the Dem shall return E_NOT_OK. In this case all the out-parameters of Dem_DcmGetDTRData are undefined.⌋

7.9.5.5.1 DTR visibility

This chapter describes a run-time (re-)configuration method for variant handling to set DTRs to invisible without removing them from the configuration.

It is recommended to set DTRs to invisible in the ECU startup phase before diagnostic requests are processed.

Interleaving of any run-time reconfiguration and diagnostic request processing may lead to temporary "inconsistencies" in diagnostic protocols.

The non-volatile storage of the visibility state of a DTR is a concept to minimize this issue.

The Dem support the visibility of DTRs.

DTRs which are invisible are treated as if not configured.

[SWS_Dem_01410]

Upstream requirements: RS_Diag_04274

⌈The visibility state shall be set by calling the interface Dem_SetDTR. If Dem_SetDTR is called with Dem_DTRControlType set to DEM_DTR_CTL_INVISIBLE the DTR state is set to "invisible". If Dem_SetDTR is called with any other defined Dem_DTRControlType (DEM_DTR_CTL_NORMAL, DEM_DTR_CTL_NO_MAX, DEM_DTR_CTL_NO_MIN, DEM_DTR_CTL_RESET) the DTR state is set to "visible".⌋

[SWS_Dem_01411]

Upstream requirements: RS_Diag_04274

⌈The visibility state of DTRs shall be stored non-volatile. The initial visibility state of DTRs shall be "visible".⌋

7.9.5.6 Service $07 - Read emission-related diagnostic trouble codes detected during during current or last completed driving cycle

The diagnostic service provides the OBD DTCs where the DTC Status bit 2 (PendingDTC) is currently set. Therefore the interfaces described in chapter 8.3.3.1 will be used by Dcm.

7.9.5.7 Read Emission-Related Diagnostic Trouble Codes with Permanent Status

OBD regulations require to store DTCs in a permanent memory that is not cleared with a ClearDTC command or on power fail. DTCs entered in the permanent memory are removed under special conditions. These special conditions are defined by the OBD regulations.

An OBD related DTC entering the permanent fault memory with illuminating the MIL and removed with deactivating the MIL.

[SWS_Dem_00300] Storing permanent DTCs ⌈The Dem shall store Permanent DTCs according to the regulation as defined by [20] within the DTCOrigin DEM_DTC_ORIGIN_PERMANENT_MEMORY.⌋

For details refer to OBD relevant DTCs in chapter 7.4.1.

[SWS_Dem_00301] Reporting stored permanent DTCs ⌈On a set DTC filter via Dem_SetDTCFilter and DTCOrigin = DEM_DTC_ORIGIN_PERMANENT_MEMORY, the Dem shall report in Dem_GetNextFilteredDTC the stored permanent DTCs.⌋

PFC Cycle

A SWC can use the Dem to get the state of the PFC cycle. The DemOBDPFCCycleRef defines the cycle used for the PFC cycle, it's state can be polled by the application via Dem_GetCycleQualified. The SWC may use this information on the MIL Master Ecu to forward it as "Minimum trip conditions satisfied" via the vehicle network to the primary ECUs.

In a primary ECU the interface Dem_GetCycleQualified can be used to get informed that the current OBD driving cycle has met the criteria for the PFC cycle by a SWC.

7.9.5.8 MIL Handling

Typically, the "MIL Master functionality" is located in the OBD master ECU. Only the MIL master functionality is allowed to send the MIL request to the instrument cluster.

Therefore in primary ECUs the function Dem_GetIndicatorStatus can be used to poll the MIL indicator status and send the information to the "MIL Master functionality" (e.g. located in the master ECU).

7.9.6 UDS based OBD protocols

This chapter describes the Dem provided functionality with respect to the UDS based OBD protocols: OBDonUDS as defined in SAE J1979-2 [17] and WWH-OBD as defined in ISO 27145-3 [21].

UDS based OBD DTCs are stored within the primary fault memory and may also store manufacturer specific non-OBD information such as snapshot records or extended data records. The FunctionalGroupIdentifier with a value of 0x33 is used to make a DTC relevant for the Dem as UDS based OBD DTC with a DTC origin set to DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY.

7.9.6.1 DTC WWHOBD class

[SWS_Dem_01140] ⌈The Dem shall provide a class value per DTC (to characterize the impact of a malfunction OBD system's monitoring capability) according to ISO 14229-1 [1], Annex D.3 "DTCSeverityMask and DTC class information bit definitions" (refer to Dem_GetSeverityOfDTC and Dem_GetNextFilteredDTCAndSeverity), only if configured for at least one DTC.⌋

[SWS_Dem_CONSTR_06109] ⌈The DTC class is only available for ISO 14229-1 [1] DTCs. It is configurable per DTC optionally (refer to DemWWHOBDDTCClass).⌋

[SWS_Dem_CONSTR_06110] ⌈The WWH-OBD DTC priority shall be according table Table 7.5.⌋

WWH-OBD DTC class	*Priority*
A	> B1 DTC
B1	> B2 DTC
B2	> C DTC
C	> no OBD DTC

Table 7.5: WWH-OBD DTC priority

Note : ISO 27145-3 [21]defines the following DTC Classes utilizing the severity levels: No Class, Class A, Class B1, Class B2 and Class C.

7.9.6.2 Service $19 42 - Read WWH-OBD DTC By Mask Record

This diagnostic service returns the OBD related DTCs that will match the given DTC Status mask and DTC Severity mask.

[SWS_Dem_01177] ⌈The interface Dem_GetDTCSeverityAvailabilityMask shall provide the DtcSeverityAvailabilityMask to the DCM.⌋

Note: The DTCSeverityAvailabilityMask can be derived from the configured DTC severity and the WWH-OBD DTC class.

7.9.6.3 Service $14 FFFF33 - Clear Emission Related DTCs

[SWS_Dem_01178] ⌈In addition to clearing event memory entries (refer to [SWS_Dem_01203]) the Dem shall clear/reset the following values:

- Activation Mode
- Readiness (PID $01, PID $41 and PID $90)
- Continuous-MI Counter
- Highest ECU B1 Counter
- B1 Counter
- Legislative Freeze Frame

⌋

For details see chapter 7.7.2.2.

[SWS_Dem_01179] ⌈Service $14 FFFF33 shall clear all DTCs (WWH-OBD DTCs and UDS DTCs)⌋

7.9.6.4 WWH-OBD

7.9.6.4.1 Continuous-MI counter

The counter Continuous-MI counter is representing the number of engine hours during which the MIL was continuously commanded to be on.

[SWS_Dem_01141] ⌈If Activation Mode 4 becomes active the counter shall be reset to 0.⌋

[SWS_Dem_01142] ⌈The counter shall increment for each engine operating hour during which the event is detected and the Continuous-MI is commanded to be on.⌋

[SWS_Dem_01143] ⌈The counter shall halt its value when the Continuous-MI is no longer commanded to be on.⌋

[SWS_Dem_01144] ⌈The counter shall continue its incrementation if the Continuous-MI is again commanded on within three operation sequences.⌋

[SWS_Dem_01145] ⌈The counter shall reset to zero and start incrementing its value if the Continuous-MI is again commanded on after more than three operation sequences.⌋

[SWS_Dem_01146] ⌈The counter shall reset to zero if the Continuous-MI is not commanded on after more than 40 warm-up cycles or after more than 200 engine operating hours or upon a ClearDTC request.⌋

[SWS_Dem_01147] ⌈The counter shall be reported as a two byte value. If any count operation occurs which would cause a counter to roll over past 0x00FF then the count value shall instead be maintained at 0x00FF.⌋

[SWS_Dem_01148] ⌈The counter shall be stored non-volatile.⌋

7.9.6.4.2 Cumulative Continuous-MI counter (Master ECU)

The counter `Cumulative Continuous-MI counter` is representing the number of engine hours during which MIL has been continuously commanded to be on during its lifetime. The counter shall only be implemented in the `VOBD`.

[SWS_Dem_01149] ⌈The counter shall increment for each engine operating hour during which the event is detected and the Continuous-MI is commanded to be on,⌋

[SWS_Dem_01150] ⌈The counter shall halt its value when the Continuous-MI is no longer commanded to be on,⌋

[SWS_Dem_01151] ⌈The counter shall never be reset.⌋

[SWS_Dem_01152] ⌈The counter shall be reported as a two byte value. If any count operation occurs which would cause a counter to roll over past 0x00FF then the count value shall instead be maintained at 0x00FF.⌋

[SWS_Dem_01153] ⌈The counter shall be stored non-volatile.⌋

7.9.6.4.3 Class B1 counter

The `Class B1 counter` is representing the number of engine hours during which a Class B1 has been Confirmed and TestFailed. The counter can be implemented as global, i.e. one counter per ECU or local, i.e. one counter per B1 Event. The `Dem` module supports only the global `B1 counter`.

[SWS_Dem_01154] ⌈The B1 counter shall increment for each 1 hour engine operating hour during which at least one Class B1 event is detected as Confirmed and TestFailed.⌋

[SWS_Dem_01155] ⌈The B1 counter shall latch its value when no Class B1 event is no longer detected as Confirmed and TestFailed or after a ClearDTC request.⌋

The B1 counter continue its incrementation if a Class B1 event is again detected as Confirmed and TestFailed.

[SWS_Dem_01156] ⌈The B1 counter shall reset to zero after three consecutive operating sequences where no Class B1 event have been detected.⌋

[SWS_Dem_01157] ⌈If B1 Counter exceeds 200 engine operating hours and no Class B1 event have been detected the counter shall be set to 190.⌋

[SWS_Dem_01158] ⌈The B1 counter shall be reported as a two byte value. If any count operation occurs which would cause a counter to roll over past 0x00FF then the count value shall instead be latched at 0x00FF.⌋

[SWS_Dem_01159] ⌈The B1 counter shall be stored non-volatile.⌋

7.9.6.4.4 Activation Mode

Activation Mode Escalation

[SWS_Dem_01160] ⌈If the B1 counter exceeds 200 engine operating hours the `Activation Mode 4` shall be active.⌋

[SWS_Dem_01161] ⌈If at least one Class A event is Confirmed and TestFailed the `Activation Mode 4` shall be active.⌋

Activation Mode Deescalation/ Healing

[SWS_Dem_01162] ⌈If `Activation Mode 4` is active and no `Activation Mode 4` relevant condition is present anymore, then the `Activation Mode 3` shall be activated.⌋

[SWS_Dem_01163] ⌈If `Activation Mode 4` has been degraded to `Activation Mode 3` and an `Activation Mode 4` relevant event is confirmed the `Activation Mode 4` shall be reactivated if the configured value of EventFailureCycleCounterThreshold is reached.⌋

[SWS_Dem_01164] ⌈If `Activation Mode 3` is active and no `Activation Mode 3` and `Activation Mode 4` relevant condition is present, then the `Activation Mode 1` shall be set.⌋

[SWS_Dem_01165] ⌈If an Class C event is Confirmed and TestFailed and no `Activation Mode 3` or `Activation Mode 4` is active, the `Activation Mode 2` shall be set to active.⌋

[SWS_Dem_01166] ⌈The function `Dem_GetIndicatorStatus` shall return the current "indicator status"(refer to Table 7.6) of those events referenced by the `DemMILIndicatorRef`. The conditions for the activation modes are defined in Table 7.6.⌋

Condition	*Activation Mode*	*Indicator Status*
No `WWH-OBD` event active	1	`DEM_INDICATOR_OFF`

Class C event active	2	DEM_INDICATOR_ON_DEMAND
Class B event active and B1 counter < 200h	3	DEM_INDICATOR_SHORT
Class A event active or B1 counter >= 200h	4	DEM_INDICATOR_CONTINUOUS

Table 7.6: Mapping of conditions to the activation mode

7.9.6.4.5 Freeze Frame 0x00

The Freeze Frame shall provide the operating conditions of the vehicle at the time of malfunction detection.

[SWS_Dem_01170] ⌈The legislative Freeze Frame 0x00 shall be stored on transition of `UDS status bit 2` from 0 to 1 (via `DemFreezeFrameRecordTrigger` set to `DEM_TRIGGER_ON_PENDING`).⌋

[SWS_Dem_01171] ⌈The update behavior if the Freeze Frame 0x00 is updated on transition of `UDS status bit 3` from 0 to 1, can be configured using `DemFreeze-FrameRecordUpdate`.⌋

[SWS_Dem_01172] ⌈The storage/reporting of freeze frame information associated with a Class A malfunction shall take precedence over a Class B1 malfunction which shall take precedence over a Class B2 malfunction and likewise for information associated with a Class C malfunction.⌋

[SWS_Dem_01173] ⌈The first malfunction detected shall take precedence over the most recent malfunction unless the most recent malfunction is of a higher class.⌋

[SWS_Dem_01174] ⌈The Freeze Frame 0x00 shall be erased if the event was removed from fault memory due to aging.⌋

[SWS_Dem_01175] ⌈The legislative freeze frame 0x00 is configured analogues to the non-emission related freeze frames.⌋

For details see chapter 7.7.7.1.

7.9.6.4.6 Extended Data Record 0x90

Extended Data Record 0x90 is not supported due to a single B1 counter only supported in Dem.

7.9.6.4.7 Aging

[SWS_Dem_01176] ⌈Additional to event aging (refer to [SWS_Dem_00698]) the event shall be aged after 200 hours of engine operation while the aging conditions are met.⌋

For details see chapter 7.7.8.

[SWS_Dem_CONSTR_06111] ⌈An OBD related DTC shall have an aging counter threshold of 40.⌋

[SWS_Dem_CONSTR_06112] ⌈An OBD related DTC shall have the Warm-Up cycle as aging cycle.⌋

7.9.6.5 OBDonUDS

7.9.6.5.1 Legislated FreezeFrame storage for OBDonUDS

[SWS_Dem_01370] Legislated FreezeFrame support ⌈The Dem shall support legislated freeze frames according to SAE J1979-2 [17].⌋

[SWS_Dem_01371] Data content of legislated freeze frames ⌈If a DTC assigned to the primary fault memory is configured and refers to a DemFreezeFrameRecordNumber with 0x00 and 0xF0, the Dem shall sample the legislated freeze frame based on the data configured via DemUDSOBDFreezeFrameClassRef.⌋

[SWS_Dem_CONSTR_06177] ⌈DemUDSOBDFreezeFrameClassRef shall not be configured if DemPidClass is configured in a specific configuration variant.⌋

[SWS_Dem_01409] UDS relevant OBD DTCs

Upstream requirements: RS_Diag_04252

⌈If DemDtcOBDRelevance is set to true for a DTC, the Dem shall assign this DTC to the DTC origin DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY.⌋

[SWS_Dem_CONSTR_06183] Restriction of DemObdDTCRef for non-OBD-relevant DTCs ⌈The DemObdDTCRef shall not be configured for a DemDTC where DemDtcOBDRelevance is set to false.⌋

[SWS_Dem_CONSTR_06184] Restriction of DemDtrEventRef for non-OBD-relevant DTCs ⌈The DemDtrEventRef shall not reference an event that is linked to a DemDTC where DemDtcOBDRelevance is set to false.⌋

[SWS_Dem_CONSTR_06185] Restriction of DemOBDGroupingAssociativeEventsRef for non-OBD-relevant DTCs ⌈The DemOBDGroupingAssociativeEventsRef shall not reference an event that is linked to a DemDTC where DemDtcOBDRelevance is set to false.⌋

This effectively means that a DemDtc with DemDtcOBDRelevance set to true will make the DTC OBD relevant. This DTC is reported by OBD related APIs where DTCs with origin of DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY are requested. OBD related data is calculated by the Dem and reported to the Dcm in that case.

7.10 J1939 specific functionality

The Dem module is capable to support J1939 networks. On these, it uses a J1939 specific DTC.

[SWS_Dem_00845]

Upstream requirements: RS_Diag_04112

⌈The J1939 support shall be available in case the configuration container DemGeneralJ1939 (see DemGeneral) is present.⌋

Note: The Dem is designed to distinguish 'Service only DTCs' and 'all other DTCs' assigning DTCs to certain event memories. In order to work correctly, with the J1939 Dcm APIs 'Service only DTCs' need to be assigned to the user defined memory. All other DTCs need to be assigned to primary memory.

7.10.1 Read DTC

The J1939-73 [14] standard defines various DM messages to report DTC information. The differences among these DM messages is mostly due to different filters on DTC states, such as active or pending DTCs, or on the OBD DTC class. The Dem provides a uniform way to read out the stored DTC information for these messages. The sequence diagram in figure 9.9 shows the general approach to retrieving the J1939 DTC information. The reporting content for different J1939 DM messages is accessed by using different input filter parameters.

[SWS_Dem_00855]

Upstream requirements: RS_Diag_04112

⌈The function Dem_J1939DcmSetDTCFilter shall set the filter mask attributes to be used for the subsequent calls of Dem_J1939DcmGetNumberOfFilteredDTC, Dem_J1939DcmGetNextFilteredDTC, and Dem_J1939DcmGetNextFilteredDTCWithSeverity, and reset an internal counter to the first event.⌋

[SWS_Dem_00856]

Upstream requirements: RS_Diag_04112

⌈The filter mask attributes set via Dem_J1939DcmSetDTCFilter shall be used until the next call of Dem_J1939DcmSetDTCFilter or Dem initialization.⌋

[SWS_Dem_00857] ⌈The function Dem_J1939DcmSetDTCFilter shall return the current composite status of the J1939 lamps. The matching filter criteria are defined in Table 7.7.⌋

SetFilter Parameter	*UDS Status filter*	*further filter criteria*	*DM representation*	
			AllDTCs	*EmissionDTCs*
DEM_J1939DTC_ACTIVE	(ConfirmedDTC == 1 AND TestFailed == 1) OR MIL_ON	n/a	DM01	DM12
DEM_J1939DTC _PREVIOUSLY_ACTIVE	ConfirmedDTC == 1 AND TestFailed == 0 AND MIL_OFF	n/a	DM02	DM23
DEM_J1939DTC_PENDING	PendingDTC == 1	n/a	DM27	DM06
DEM_J1939DTC_ PERMANENT	n/a	permanent memory entry available	n/a	DM28
DEM_J1939DTC _CURRENTLY_ACTIVE	TestFailed == 1	n/a	DM35	n/a

Table 7.7: Types of errors which can be detected by the Dem module

Each reading of DTC returns a composite lamp status of the following lamps:

a) Malfunction Indicator Lamp

b) Red Stop Lamp

c) Amber Warning Lamp

d) Protect Lamp

7.10.1.1 Composite Malfunction Indicator Lamp Status

[SWS_Dem_00858]

Upstream requirements: RS_Diag_04110

⌈The composite “Malfunction Indicator Lamp” shall be set to “Lamp Off” in case the Indicator referenced by DemMILIndicatorRef has an IndicatorStatus DEM_INDICATOR_OFF. All other IndicatorStatus states than “DEM_INDICATOR_OFF” shall set the composite Malfunction Indicator Lamp to “Lamp On”.⌋

[SWS_Dem_00859]

Upstream requirements: RS_Diag_04110

⌈The composite “Flash Malfunction Indicator Lamp” shall be set to “Unavailable / Do Not Flash” (0x03) in case the Indicator referenced by DemMILIndicatorRef has an IndicatorStatus DEM_INDICATOR_OFF or DEM_INDICATOR_CONTINUOUS.⌋

[SWS_Dem_00860]

Upstream requirements: RS_Diag_04110

⌈The composite “Flash Malfunction Indicator Lamp” shall be set to “Slow Flash” (0x00) in case the Indicator referenced by DemMILIndicatorRef has an IndicatorStatus DEM_INDICATOR_SLOW_FLASH.⌋

[SWS_Dem_00861]

Upstream requirements: RS_Diag_04110

⌈The composite “Flash Malfunction Indicator Lamp” shall be set to “Fast Flash” (0x01) in case the Indicator referenced by DemMILIndicatorRef has an IndicatorStatus DEM_INDICATOR_FAST_FLASH.⌋

7.10.1.2 Composite Red Stop Lamp Status

[SWS_Dem_00862]

Upstream requirements: RS_Diag_04110

⌈The composite “Red Stop Lamp” shall be set to “Lamp Off” in case the Indicator referenced by DemRedStopLampIndicatorRef has an IndicatorStatus DEM_INDICATOR_OFF. All other IndicatorStatus states than “DEM_INDICATOR_OFF” shall set the composite Red Stop Lamp to “Lamp On”.⌋

[SWS_Dem_00863]

Upstream requirements: RS_Diag_04110

⌈The composite “Flash Red Stop Lamp” shall be set to “Unavailable / Do Not Flash” (0x03) in case the Indicator referenced by DemRedStopLampIndicatorRef has an IndicatorStatus DEM_INDICATOR_OFF or DEM_INDICATOR_CONTINUOUS.⌋

[SWS_Dem_00864]

Upstream requirements: RS_Diag_04112

⌈The composite “Flash Red Stop Lamp” shall be set to “Slow Flash” (0x00) in case the Indicator referenced by DemRedStopLampIndicatorRef has an IndicatorStatus DEM_INDICATOR_SLOW_FLASH.⌋

[SWS_Dem_00865] ⌈The composite “Flash Red Stop Lamp” shall be set to “Fast Flash” (0x01) in case the Indicator referenced by DemRedStopLampIndicatorRef has an IndicatorStatus DEM_INDICATOR_FAST_FLASH.⌋

7.10.1.3 Composite Amber Warning Lamp Status

[SWS_Dem_00866]

Upstream requirements: RS_Diag_04110

⌈The composite “Amber Warning Lamp” shall be set to “Lamp Off” in case the Indicator referenced by DemAmberWarningLampIndicatorRef has an IndicatorStatus DEM_INDICATOR_OFF. All other IndicatorStatus states than DEM_INDICATOR_OFF shall set the composite Amber Warning Lamp to “Lamp On”.⌋

[SWS_Dem_00867]

Upstream requirements: RS_Diag_04110

⌈The composite “Amber Warning Lamp” shall be set to “Unavailable / Do Not Flash” (0x03) in case the Indicator referenced by DemAmberWarningLampIndicatorRef has an IndicatorStatus DEM_INDICATOR_OFF or DEM_INDICATOR_CONTINUOUS.⌋

[SWS_Dem_00868]

Upstream requirements: RS_Diag_04110

⌈The composite “Flash Amber Warning Lamp” shall be set to “Slow Flash” (0x00) in case the Indicator referenced by DemAmberWarningLampIndicatorRef has an IndicatorStatus DEM_INDICATOR_SLOW_FLASH.⌋

[SWS_Dem_00869]

Upstream requirements: RS_Diag_04110

⌈The composite “Flash Amber Warning Lamp” shall be set to “Fast Flash” (0x01) in case the Indicator referenced by DemAmberWarningLampIndicatorRef has an IndicatorStatus DEM_INDICATOR_FAST_FLASH.⌋

7.10.1.4 Composite Protect Lamp Status

[SWS_Dem_00870]

Upstream requirements: RS_Diag_04110

⌈The composite “Protect Lamp” shall be set to “Lamp Off” in case the Indicator referenced by DemProtectLampIndicatorRef has an IndicatorStatus DEM_INDICATOR_OFF. All other IndicatorStatus states than DEM_INDICATOR_OFF shall set the composite Protect Lamp to “Lamp On”.⌋

[SWS_Dem_00871] ⌈The composite “Flash Protect Lamp” shall be set to “Unavailable / Do Not Flash” (0x03) in case the Indicator referenced by DemProtectLampIndicatorRef has an IndicatorStatus DEM_INDICATOR_OFF or DEM_INDICATOR_CONTINUOUS.⌋

[SWS_Dem_00872]

Upstream requirements: RS_Diag_04110

⌈The composite “Flash Protect Lamp” shall be set to “Slow Flash” (0x00) in case the Indicator referenced by DemProtectLampIndicatorRef has an IndicatorStatus DEM_INDICATOR_SLOW_FLASH.⌋

[SWS_Dem_00873]

Upstream requirements: RS_Diag_04110

⌈The composite “Flash Protect Lamp” shall be set to “Fast Flash” (0x01) in case the Indicator referenced by DemProtectLampIndicatorRef has an IndicatorStatus DEM_INDICATOR_FAST_FLASH.⌋

7.10.1.5 DTC data acqusition

[SWS_Dem_00874]

Upstream requirements: RS_Diag_04112

⌈The function Dem_J1939DcmGetNumberOfFilteredDTC shall return the number of J1939 DTCs matching the defined filter criteria of Table 7.7 for the corresponding primary event memory of the requested ClientId by the function call of Dem_J1939DcmSetDTCFilter.

- In case the function has calculated the total number of matching DTCs, the return value shall be E_OK.
- In case the search needs to be interrupted due to internal implementations or limitations, the return value can be set to DEM_PENDING. The out parameter needs not to be valid in this case. The next call of Dem_J1939DcmGetNumberOfFilteredDTC should continue after the interrupted element.

⌋

[SWS_Dem_01412] Setting a filter for J1939 DTCs

Upstream requirements: RS_Diag_04113

⌈A call of Dem_J1939DcmSetDTCFilter shall set the filter criteria for subsequent calls of Dem_J1939DcmGetNextFilteredDTC and Dem_J1939DcmGetNextFilteredDTCWithSeverity.⌋

[SWS_Dem_01413] Selecting all J1939 DTCs

Upstream requirements: RS_Diag_04113

⌈If Dem_J1939DcmSetDTCFilter is called with Dem_DTCKindType equal to DEM_DTC_KIND_ALL_DTCS, the Dem shall consider all J1939 DTCs.⌋

[SWS_Dem_01414] Selecting all J1939 emission related DTCs

Upstream requirements: RS_Diag_04113

⌈If Dem_J1939DcmSetDTCFilter is called with Dem_DTCKindType equal to DEM_DTC_KIND_EMISSION_REL_DTCS, the Dem shall consider all OBD emission related J1939 DTCs.⌋

[SWS_Dem_01415] Selecting all J1939 emission related Class_A DTCs

Upstream requirements: RS_Diag_04113

⌈If Dem_J1939DcmSetDTCFilter is called with Dem_DTCKindType equal to DEM_DTC_KIND_WWHOBD_CLASS_A, the Dem shall consider all OBD emission related J1939 DTCs with DemWWHOBDDTCClass set to DEM_DTC_WWHOBD_CLASS_A.⌋

[SWS_Dem_01416] Selecting all J1939 emission related Class_B1 DTCs

Upstream requirements: RS_Diag_04113

⌈If Dem_J1939DcmSetDTCFilter is called with Dem_DTCKindType equal to DEM_DTC_KIND_WWHOBD_CLASS_B1, the Dem shall consider all OBD emission related J1939 DTCs with DemWWHOBDDTCClass set to DEM_DTC_WWHOBD_CLASS_B1.⌋

[SWS_Dem_01417] Selecting all J1939 emission related Class_B2 DTCs

Upstream requirements: RS_Diag_04113

⌈If Dem_J1939DcmSetDTCFilter is called with Dem_DTCKindType equal to DEM_DTC_KIND_WWHOBD_CLASS_B2, the Dem shall consider all OBD emission related J1939 DTCs with DemWWHOBDDTCClass set to DEM_DTC_WWHOBD_CLASS_B2.⌋

[SWS_Dem_01418] Selecting all J1939 emission related Class_C DTCs

Upstream requirements: RS_Diag_04113

⌈If Dem_J1939DcmSetDTCFilter is called with Dem_DTCKindType equal to DEM_DTC_KIND_WWHOBD_CLASS_C, the Dem shall consider all OBD emission related J1939 DTCs with DemWWHOBDDTCClass set to DEM_DTC_WWHOBD_CLASS_C.⌋

[SWS_Dem_00875]

Upstream requirements: RS_Diag_04112

⌈Each call of the function Dem_J1939DcmGetNextFilteredDTC shall search for the next event having a J1939DTC assigned and matching the filter criteria of Table 7.7 for the node provided by the function call of Dem_J1939DcmSetDTCFilter. In case no more events are matching the filter criteria, the function return value shall be DEM_NO_SUCH_ELEMENT. The out parameters need not to be valid in this case.⌋

[SWS_Dem_00877]

Upstream requirements: RS_Diag_04111

⌈In case the function Dem_J1939DcmGetNextFilteredDTC has found an event matching the filter criteria it shall return the corresponding J1939DTC (refer to Dem-DTCAttributes for details) and the corresponding occurrence counter. The return

value shall be E_OK. In case the search needs to be interrupted due to internal implementations or limitations, the return value can be set to DEM_PENDING. The out parameter needs not to be valid in this case. The next call of Dem_J1939DcmGetNextFilteredDTC should continue after the interrupted element. In case the occurrence counter is above +126 (0x7E), the returned values shall be set to +126 (0x7E).⌋

[SWS_Dem_01432] Searching for the next filtered J1939 DTC with severity

Upstream requirements: RS_Diag_04112

⌈Each call of Dem_J1939DcmGetNextFilteredDTCWithSeverity shall search for the next event with an assigned J1939DTC that matches the filter criteria of Table 7.7 for the node configured via Dem_J1939DcmSetDTCFilter. If no further matching events exist, the function shall return DEM_NO_SUCH_ELEMENT; the out parameters need not be valid in this case.⌋

[SWS_Dem_01433] Returning a matching J1939 DTC, occurrence counter, and severity

Upstream requirements: RS_Diag_04111

⌈If Dem_J1939DcmGetNextFilteredDTCWithSeverity finds a matching event, it shall set the out parameters to the corresponding J1939DTC (refer to Dem_DTCAttributes for details), occurrence counter, and severity, and shall return E_OK. If the occurrence counter exceeds 126 (0x7E), the returned occurrence counter shall be clamped to 126 (0x7E). If the search must be interrupted due to internal limitations, the function shall return DEM_PENDING; the out parameters need not be valid in this case, and the next call shall resume after the interrupted element.⌋

[SWS_Dem_01434] Returning configured severity for a J1939 DTC

Upstream requirements: RS_Diag_04071

⌈If Dem_J1939DcmGetNextFilteredDTCWithSeverity is called and for the selected DTC a severity value in DemDTCSeverity is configured, the Dem shall set the value configured in DemDTCSeverity as DTCSeverity.⌋

[SWS_Dem_01435] Returning no-severity when no J1939 DTC severity is configured

Upstream requirements: RS_Diag_04071

⌈If Dem_J1939DcmGetNextFilteredDTCWithSeverity is called and for the selected DTC no severity value in DemDTCSeverity is configured, the Dem shall set DEM_SEVERITY_NO_SEVERITY as DTCSeverity.⌋

An example how to use Dem_J1939DcmSetDTCFilter to provide data for the DM41 throughout DM52 is given in table Table 7.8.

DM message	DTCStatusFilter	DTCKind	DTCOrigin
DM41	DEM_J1939DTC_PENDING	DEM_DTC_KIND_WWHOBD_CLASS_A	DEM_DTC_ORIGIN_PRIMARY_MEMORY

DM42	DEM_J1939DTC_ACTIVE	DEM_DTC_KIND_WWHOBD_CLASS_A	DEM_DTC_ORIGIN_PRIMARY_MEMORY
DM43	DEM_J1939DTC_PREVIOUSLY_ACTIVE	DEM_DTC_KIND_WWHOBD_CLASS_A	DEM_DTC_ORIGIN_PRIMARY_MEMORY
DM44	DEM_J1939DTC_PENDING	DEM_DTC_KIND_WWHOBD_CLASS_B1	DEM_DTC_ORIGIN_PRIMARY_MEMORY
DM45	DEM_J1939DTC_ACTIVE	DEM_DTC_KIND_WWHOBD_CLASS_B1	DEM_DTC_ORIGIN_PRIMARY_MEMORY
DM46	DEM_J1939DTC_PREVIOUSLY_ACTIVE	DEM_DTC_KIND_WWHOBD_CLASS_B1	DEM_DTC_ORIGIN_PRIMARY_MEMORY
DM47	DEM_J1939DTC_PENDING	DEM_DTC_KIND_WWHOBD_CLASS_B2	DEM_DTC_ORIGIN_PRIMARY_MEMORY
DM48	DEM_J1939DTC_ACTIVE	DEM_DTC_KIND_WWHOBD_CLASS_B2	DEM_DTC_ORIGIN_PRIMARY_MEMORY
DM49	DEM_J1939DTC_PREVIOUSLY_ACTIVE	DEM_DTC_KIND_WWHOBD_CLASS_B2	DEM_DTC_ORIGIN_PRIMARY_MEMORY
DM50	DEM_J1939DTC_PENDING	DEM_DTC_KIND_WWHOBD_CLASS_C	DEM_DTC_ORIGIN_PRIMARY_MEMORY
DM51	DEM_J1939DTC_ACTIVE	DEM_DTC_KIND_WWHOBD_CLASS_C	DEM_DTC_ORIGIN_PRIMARY_MEMORY
DM52	DEM_J1939DTC_PREVIOUSLY_ACTIVE	DEM_DTC_KIND_WWHOBD_CLASS_C	DEM_DTC_ORIGIN_PRIMARY_MEMORY

Table 7.8: J1939 Diagnostic Message Mapping to DTC Filters and Kinds

7.10.2 Clear DTCs

For definition of locking the clearing process, refer to chapter 7.7.2.2.

[SWS_Dem_00878] ⌈The Dem module shall provide the Dem_J1939DcmClearDTC to the J1939Dcm [7] for deleting all active or previously active DTCs from the event memory. This shall trigger also initializing of related SW-Cs / BSW modules according to [SWS_Dem_00003].⌋

When one of the diagnostic messages DM03 and DM11 of SAE J1939-73 [14] are requested, all active (DM11) or previously active (DM03) DTCs have to be cleared, along with additional information like freeze frames.

[SWS_Dem_00879]

Upstream requirements: RS_Diag_04117

⌈The function Dem_J1939DcmClearDTC shall clear the status of all event(s) related to the specified DTC(s), as well as all associated event memory entries for these event(s).⌋

For DM22 support, the Dem module shall provide the Dem_J1939DcmClearSingle-DTC to the J1939Dcm [7] for deleting an individual active or previously active DTC from the event memory. This shall also trigger initialization of related SW-Cs / BSW modules according to [SWS_Dem_00003].

[SWS_Dem_01430] Clearing an individual J1939 DTC

Upstream requirements: RS_Diag_04113

⌈If the function Dem_J1939DcmClearSingleDTC is called, the Dem shall clear the status of all event(s) related to the specified DTC, as well as all associated event memory entries for these event(s).⌋

Note: Exceptions may apply due to [SWS_Dem_00515].

7.10.3 DM31

The DM31 provides the applicable lamp(s) and their status for each individual DTC, wherefore the DTC-specific lamp status needs to be returned by the function Dem_J1939DcmFirstDTCwithLampStatus.

[SWS_Dem_00880]

Upstream requirements: RS_Diag_04113

⌈Each call to Dem_J1939DcmFirstDTCwithLampStatus shall set the internal counter to the first event having a J1939DTC for this particular “ClientId” assiged.⌋

[SWS_Dem_00881]

Upstream requirements: RS_Diag_04113

⌈Each call to Dem_J1939DcmGetNextDTCwithLampStatus shall search within the event memory (set by the function Dem_J1939DcmFirstDTCwithLampStatus) for the next event having a J1939DTC assigned. In case no more events are available that have a J1939DTC assigend, the function return value shall be DEM_NO_SUCH_ELEMENT. The out parameter needs not to be valid in this case.⌋

[SWS_Dem_00882]

Upstream requirements: RS_Diag_04113

⌈In case Dem_J1939DcmGetNextDTCwithLampStatus has found an event matching the filter criteria it shall return the J1939DTC specific status of the lamps (refer to following sections for details), the corresponding J1939DTC (refer DemDTCAttributes for details) and the corresponding occurrence counter. The return value shall be E_OK. In case the search needs to be interrupted due to internal implementations or limitations, the return value can be set to DEM_PENDING. The out parameter needs not to be valid in this case. The next call of Dem_J1939DcmGetNextDTCwithLampStatus should continue after the interrupted element. In case the occurrence counter is above +126 (0x7E), the returned values shall be set to +126 (0x7E).⌋

7.10.3.1 Malfunction Indicator Lamp

[SWS_Dem_00883]

Upstream requirements: RS_Diag_04110

⌈The DTC-specific "Malfunction Indicator Lamp" returned in the function `Dem_J1939DcmGetNextDTCwithLampStatus` shall be set to "Lamp On" if the Indicator referenced by `DemMILIndicatorRef` is configured for the corresponding event and the UDS status bit 7 (WarningIndicator) is active (set to 1). It shall be set to "Lamp Off" if the Indicator referenced by `DemMILIndicatorRef` is configured for the corresponding event and the UDS status bit 7 (WarningIndicator) is passive (set to 0). It shall be set to "Unavailable" if the Indicator referenced by `DemMILIndicatorRef` is not configured for the corresponding event.⌋

[SWS_Dem_00884]

Upstream requirements: RS_Diag_04110

⌈The DTC-specific "Flash Malfunction Indicator Lamp" shall be set to "Unavailable / Do Not Flash" (0x03) if the corresponding event has not set "Lamp On", or if the corresponding event has set "Lamp On" and the `DemIndicatorBehaviour` of the Indicator referenced by `DemMILIndicatorRef` is set to `DEM_INDICATOR_CONTINUOUS`.⌋

[SWS_Dem_00885]

Upstream requirements: RS_Diag_04110

⌈The DTC-specific "Flash Malfunction Indicator Lamp" shall be set to "Slow Flash" (0x00) in case the corresponding event has set "Lamp On" and the `DemIndicatorBehaviour` of the Indicator referenced by `DemMILIndicatorRef` is set to `DEM_INDICATOR_SLOW_FLASH`.⌋

[SWS_Dem_00886] ⌈The DTC-specific "Flash Malfunction Indicator Lamp" shall be set to "Fast Flash" (0x01) in case the corresponding event has set "Lamp On" and the `DemIndicatorBehaviour` of the Indicator referenced by `DemMILIndicatorRef` is set to `DEM_INDICATOR_FAST_FLASH`.⌋

7.10.3.2 Red Stop Lamp

[SWS_Dem_00887]

Upstream requirements: RS_Diag_04110

⌈The DTC-specific "Red Stop Lamp" returned in the function `Dem_J1939DcmGetNextDTCwithLampStatus` shall be set to "Lamp On" if the Indicator referenced by `DemRedStopLampIndicatorRef` is configured for the corresponding event and the UDS status bit 7 (WarningIndicator) is active (set to 1). It shall be set to "Lamp Off" if the Indicator referenced by `DemRedStopLampIndicatorRef` is configured for the corresponding event and the UDS status bit 7 (WarningIndicator) is passive (set to

0). It shall be set to “Unavailable” if the Indicator referenced by DemRedStopLamp-IndicatorRef is not configured for the corresponding event.⌋

[SWS_Dem_00888]

Upstream requirements: RS_Diag_04110

⌈The DTC-specific “Flash Red Stop Lamp” shall be set to “Unavailable / Do Not Flash” (0x03) if the corresponding event has not set “Lamp On”, or if the corresponding event has set “Lamp On” and the DemIndicatorBehaviour of the Indicator referenced by DemRedStopLampIndicatorRef is set to DEM_INDICATOR_CONTINUOUS.⌋

[SWS_Dem_00889]

Upstream requirements: RS_Diag_04110

⌈The DTC-specific “Flash Red Stop Lamp” shall be set to “Slow Flash” (0x00) in case the corresponding event has set “Lamp On” and the DemIndicatorBehaviour of the Indicator referenced by DemRedStopLampIndicatorRef is set to DEM_INDICATOR_SLOW_FLASH.⌋

[SWS_Dem_00890] ⌈The DTC-specific “Flash Red Stop Lamp” shall be set to “Fast Flash” (0x01) in case the corresponding event has set “Lamp On” and the DemIndicatorBehaviour of the Indicator referenced by DemRedStopLamp-IndicatorRef is set to DEM_INDICATOR_FAST_FLASH.⌋

7.10.3.3 Amber Warning Lamp

[SWS_Dem_00891]

Upstream requirements: RS_Diag_04110

⌈The DTC-specific “Amber Warning Lamp” returned in the function Dem_J1939Dcm-GetNextDTCwithLampStatus shall be set to “Lamp On” if the Indicator referenced by DemAmberWarningLampIndicatorRef is configured for the corresponding event and the UDS status bit 7 (WarningIndicator) is active (set to 1). It shall be set to “Lamp Off” if the Indicator referenced by DemAmberWarningLampIndicatorRef is configured for the corresponding event and the UDS status bit 7 (WarningIndicator) is passive (set to 0). It shall be set to “Unavailable” if the Indicator referenced by Dem-AmberWarningLampIndicatorRef is not configured for the corresponding event.⌋

[SWS_Dem_00892]

Upstream requirements: RS_Diag_04110

⌈The DTC-specific “Flash Amber Warning Lamp” shall be set to “Unavailable / Do Not Flash” (0x03) if the corresponding event has not set “Lamp On”, or if the corresponding event has set “Lamp On” and the DemIndicatorBehaviour of the Indicator referenced by DemAmberWarningLampIndicatorRef is set to DEM_INDICATOR_CONTINUOUS.⌋

[SWS_Dem_00893]

Upstream requirements: RS_Diag_04110

⌈The DTC-specific “Flash Amber Warning Lamp” shall be set to “Slow Flash” (0x00) in case the corresponding event has set “Lamp On” and the DemIndicatorBehaviour of the Indicator referenced by DemAmberWarningLampIndicatorRef is set to DEM_INDICATOR_SLOW_FLASH.⌋

[SWS_Dem_00894]

Upstream requirements: RS_Diag_04110

⌈The DTC-specific “Flash Amber Warning Lamp” shall be set to “Fast Flash” (0x01) in case the corresponding event has set “Lamp On” and the DemIndicatorBehaviour of the Indicator referenced by DemAmberWarningLampIndicatorRef is set to DEM_INDICATOR_FAST_FLASH.⌋

7.10.3.4 Protect Lamp

[SWS_Dem_00895]

Upstream requirements: RS_Diag_04110

⌈The DTC-specific “Protect Lamp” returned in the function Dem_J1939DcmGetNext-DTCwithLampStatus shall be set to “Lamp On” if the Indicator referenced by Dem-ProtectLampIndicatorRef is configured for the corresponding event and the UDS status bit 7 (WarningIndicator) is active (set to 1). It shall be set to “Lamp Off” if the Indicator referenced by DemProtectLampIndicatorRef is configured for the corresponding event and the UDS status bit 7 (WarningIndicator) is passive (set to 0). It shall be set to “Unavailable” if the Indicator referenced by DemProtectLampIndicator-Ref is not configured for the corresponding event.⌋

[SWS_Dem_00896] ⌈The DTC-specific “Flash Protect Lamp” shall be set to “Unavailable / Do Not Flash” (0x03) if the corresponding event has not set “Lamp On”, or if the corresponding event has set “Lamp On” and the DemIndicatorBehaviour of the Indicator referenced by DemProtectLampIndicatorRef is set to DEM_INDICATOR_CONTINUOUS.⌋

[SWS_Dem_00897]

Upstream requirements: RS_Diag_04110

⌈The DTC-specific “Flash Protect Lamp” shall be set to “Slow Flash” (0x00) in case the corresponding event has set “Lamp On” and the DemIndicatorBehaviour of the Indicator referenced by DemProtectLampIndicatorRef is set to DEM_INDICATOR_SLOW_FLASH.⌋

[SWS_Dem_00898]

Upstream requirements: RS_Diag_04110

⌈The DTC-specific “Flash Protect Lamp” shall be set to “Fast Flash” (0x01) in case the corresponding event has set “Lamp On” and the DemIndicatorBehaviour of the Indicator referenced by DemProtectLampIndicatorRef is set to DEM_INDICATOR_FAST_FLASH.⌋

7.10.3.5 J1939 lamp status

[SWS_Dem_01310]

Upstream requirements: RS_Diag_04190

⌈If the configuration parameter DemInternalDataElement is set to DEM_J1939LAMP_STATUS, then the lamp status in the format of type Dem_J1939DcmLampStatusType shall be mapped to the respective data element.⌋

For details see chapter 7.10.3.

[SWS_Dem_01311]

Upstream requirements: RS_Diag_04190

⌈The DemInternalDataElement for DEM_J1939LAMP_STATUS shall contain the lamp status before the event was processed Dem internally.⌋

7.10.4 FreezeFrame

[SWS_Dem_00899]

Upstream requirements: RS_Diag_04111

⌈The function Dem_J1939DcmSetFreezeFrameFilter shall set the filter mask attributes to be used for the subsequent calls of Dem_J1939DcmGetNextFreezeFrame, and reset an internal counter to the first freeze frame. The filter mask attributes shall be used until the next call of Dem_J1939DcmSetFreezeFrameFilter or Dem initialization. The matching filter citreria are DEM_J1939DCM_FREEZEFRAME (filtering the data in DemJ1939FreezeFrameClassRef) or DEM_J1939DCM_EXPANDED_FREEZEFRAME (filtering the data in DemJ1939ExpandedFreezeFrameClassRef).⌋

[SWS_Dem_00900]

Upstream requirements: RS_Diag_04111

⌈Each function call of Dem_J1939DcmGetNextFreezeFrame shall step to the next (Expanded-)FreezeFrame for a specific event memory defined by the filter criteria (FreezeFrameKind and ClientId) of Dem_J1939DcmSetFreezeFrameFilter. In case no more (Expanded-)FreezeFrames are matching the filter criteria, the function

shall return DEM_NO_SUCH_ELEMENT. The out parameters need not to be valid in this case.⌋

[SWS_Dem_00901]

Upstream requirements: RS_Diag_04111

⌈The function Dem_J1939DcmGetNextFreezeFrame shall trigger the Det error DEM_E_WRONG_CONDITION in case of not supported FreezeFrameKind. Valid values are DEM_J1939DCM_FREEZEFRAME and DEM_J1939DCM_EXPANDED_FREEZEFRAME.⌋

[SWS_Dem_00902]

Upstream requirements: RS_Diag_04111

⌈In case the search of [SWS_Dem_00901] needs to be interrupted due to internal implementations or limitations, the return value can be set to DEM_PENDING. The out parameter needs not to be valid in this case. The next call of Dem_J1939DcmGetNextFreezeFrame should continue after the interrupted element.⌋

[SWS_Dem_01384]

Upstream requirements: RS_Diag_04111

⌈The function Dem_J1939DcmGetNextFreezeFrame shall return DEM_BUFFER_TOO_SMALL if the buffer in the DestBuffer parameter is not big enough to hold the (Expanded-)FreezeFrame.⌋

[SWS_Dem_01385]

Upstream requirements: RS_Diag_04111

⌈The function Dem_J1939DcmGetNextFreezeFrame shall copy the (Expanded-)FreezeFrame data into the buffer provided by the parameter DestBuffer (in case of Expanded FreezeFrames without any SPN informations).⌋

[SWS_Dem_01386]

Upstream requirements: RS_Diag_04111

⌈The function Dem_J1939DcmGetNextFreezeFrame shall set the parameter J1939DTC to the corresponding J1939DTC value (refer DemDTCAttributes for details).⌋

[SWS_Dem_01387]

Upstream requirements: RS_Diag_04111

⌈The function Dem_J1939DcmGetNextFreezeFrame shall set the parameter OccurenceCounter to the corresponding occurrence counter value.⌋

[SWS_Dem_01388]

Upstream requirements: RS_Diag_04111

⌈If the occurrence counter is above +126 (0x7E), the function Dem_J1939DcmGetNextFreezeFrame shall set the parameter OccurenceCounter to +126 (0x7E).⌋

7.10.4.1 SPNs in ExpandedFreezeFrame

[SWS_Dem_00904]

Upstream requirements: RS_Diag_04111

⌈Each call to Dem_J1939DcmGetNextSPNInFreezeFrame shall step to the next SPN of the ExpandedFreezeFrame for this DemEventMemory. In case no more SPN of the ExpandedFreezeFrame definition is available, the function return value shall be DEM_NO_SUCH_ELEMENT. The out parameter needs not to be valid in this case.⌋

[SWS_Dem_00905]

Upstream requirements: RS_Diag_04111

⌈In case the search of [SWS_Dem_00904] needs to be interrupted due to internal implementations or limitations, the return value can be set to DEM_PENDING. The out parameter needs not to be valid in this case. The next call of Dem_J1939DcmGetNextSPNInFreezeFrame should continue after the interrupted element.⌋

[SWS_Dem_00906]

Upstream requirements: RS_Diag_04057, RS_Diag_04111

⌈The function Dem_J1939DcmGetNextSPNInFreezeFrame shall trigger the Det error DEM_E_WRONG_CONDITION in case of not supported FreezeFrameKind. Valid value is Dem_SPNsInExpandedFreezeFrame.⌋

[SWS_Dem_00907]

Upstream requirements: RS_Diag_04111

⌈If Dem_J1939DcmGetNextSPNInFreezeFrame is called and a further SPN of the ExpandedFreezeFrame is found, the Dem shall copy the SPN number into SPNSupported and corresponding data length of the SPN into SPNDataLength and return E_OK.⌋

7.10.5 Diagnostic Readiness

The Dem module needs to calculate internally the (Non-)Continuously Monitored Systems support and status for “Diagnostic Readiness 1” (DM05), “Diagnostic Readiness 2” (DM21), and “Diagnostic Readiness 3” (DM26).

[SWS_Dem_00908]

Upstream requirements: RS_Diag_04113

⌈The J1939 events shall use the configuration parameter DemEventOBDReadinessGroup (refer to DemObdDTC) to assign individual events to a Continuously or Non-Continuously Monitored System.⌋

Dem_J1939DcmReadDiagnosticReadiness1 reports the diagnostics information that relates to diagnostic readiness according DM05.

[SWS_Dem_00909]

Upstream requirements: RS_Diag_04113

⌈A call of Dem_J1939DcmReadDiagnosticReadiness1 shall report a response message based on the J1939-73 [14] DM05 definition:

- "Active Trouble Codes" shall report the number of active DTCs identical to the number of filtered DTCs by Dem_J1939DcmSetDTCFilter (Dem_Active, DEM_DTC_KIND_ALL_DTCS).
- "Previously Active Diagnostic Trouble Codes" shall report the number of previously active DTCs identical to the number of filtered DTCs by Dem_J1939DcmSetDTCFilter (Dem_PreviouslyActive, DEM_DTC_KIND_ALL_DTCS).
- "OBD Compliance" shall be based on the configuration parameter DemOBDCompliancy (in DemGeneralOBD). For nonOBD ECUs the value five (5) shall be reported.
- The (Non-)Continuously Monitored Systems support and status shall be reported according to SAE J1939-73 chapter 5.7.5.4 to 5.7.5.6. The calculation shall be based on those events referencing the corresponding configuration parameter DemEventOBDReadinessGroup. In case there is no reference by any event parameter, the support bit shall be set to zero (0). The "monitoring status" bit shall be set to "test complete" (0) if all assigned event parameter have the TestNotCompletedSinceLastClear status bit set to zero (0), otherwise the "monitoring status" bit shall be set to "test not complete" (1).

⌋

Dem_J1939DcmReadDiagnosticReadiness2 reports the diagnostic information relevant to a second PGN conveying diagnostic readiness according to DM21 (see also DM05).

[SWS_Dem_00910]

Upstream requirements: RS_Diag_04112

⌈A call of Dem_J1939DcmReadDiagnosticReadiness2 shall report a response message based on the J1939-73 [14] DM21 definition:

- "Distance Traveled While MIL is Activated" shall be identical to PID 0x21, except the maximum value is 64255 (all values above 64255 shall be reported as 64255).
- "Distance Since Diagnostic Trouble Codes Cleared" shall be identical to PID 0x31, except the maximum value is 64255 (all values above 64255 shall be reported as 64255).

- “Minutes Run by Engine While MIL is Activated” shall be identical to PID 0x4D, except the maximum value is 64255 (all values above 64255 shall be reported as 64255).
- “Time Since Diagnostic Trouble Codes Cleared” shall be identical to PID 0x4E, except the maximum value is 64255 (all values above 64255 shall be reported as 64255).

⌋

`Dem_J1939DcmReadDiagnosticReadiness3` reports the diagnostic information conveying the pending status of OBD system monitors for the current drive cycle according to DM26 (See also DM05 and DM21).

[SWS_Dem_00911]

Upstream requirements: RS_Diag_04112

⌈A call of `Dem_J1939DcmReadDiagnosticReadiness3` shall report a response message based on the J1939-73 [14] DM26 definition:

- “Time Since Engine Start” shall be retrieved by the DemOBDTimeSinceEngineStart (in `DemGeneralOBD`)
- “Number of Warmups Since DTCs Cleared” shall be identical to PID 0x30, except the maximum value is 250 (all values above 250 shall be reported as 250).
- The (Non-)Continuously Monitored Systems Enable/Completed status shall be reported according to SAE J1939-73 [14] chapter 5.7.26.3 to 5.7.26.5. The calculation shall be based on those events referencing the corresponding configuration parameter `DemEventOBDReadinessGroup`. In case there is no reference by any event parameter it shall shall indicate disabled and complete. Otherwise the disable state shall be based on the readiness group disabled status according to [SWS_Dem_00356], and the complete status shall be set to “monitor complete” (0) if all assigned event parameter have the TestNotCompletedSinceThisOperationCycle status bit set to zero (0); otherwise the status bit shall be set to “monitor not complete” (1).

⌋

7.10.6 Monitor Performance Ratio

For J1939 the In-Use-Monitor Performace Ratio (IUMPR) requires a reporting for each supported Applicable System Monitor only.

[SWS_Dem_00912]

Upstream requirements: RS_Diag_04112

⌈The function `Dem_J1939DcmSetRatioFilter` shall reset an internal counter to the first valid SPN with DemRatioId defined to be used for the subsequent calls of `Dem_J1939DcmGetNextFilteredRatio`. Furthermore, it shall return the "Ignition Cycle Counter" according SAEJ193973 chapter 5.7.20.1, as well as "OBD Monitoring Conditions Encountered" according SAEJ193973 chapter 5.7.20.2 (CARB defines this as the general denominator).⌋

[SWS_Dem_00913] Reading IUMPR data for J1939

Upstream requirements: RS_Diag_04113

⌈With each call to the function `Dem_J1939DcmGetNextFilteredRatio` the Dem module shall return the next valid SPN and its associated Numerator and Denominator matching the filter criteria defined by the function call `Dem_J1939DcmSetRatioFilter`(refer to [SWS_Dem_00912]).⌋

[SWS_Dem_01397]

Upstream requirements: RS_Diag_04113

⌈If `Dem_J1939DcmGetNextFilteredRatio` is called and no more SPNs are available, the Dem shall return `DEM_NO_SUCH_ELEMENT`.⌋

7.11 Interaction with other Software Modules

7.11.1 Interaction with Software Components (SW-C)

In the AUTOSAR ECU architecture, the Diagnostic Event Manager implements an AUTOSAR Service. This means that the Dem module does not exclusively communicate with other BSW modules, but also with SWC via the RTE (refer to Figure 5.1).

From the viewpoint of the BSW "Cmodule" Dem, there are three kinds of dependencies between the Service and the AUTOSAR Software Components above the RTE:

- The application accesses the API (implemented as Cfunctions) of the Dem (by accessing the Dem Service Component).
- The application is optionally notified upon the outcome of requested asynchronous activity (via direct/indirect RTE API by the Dem).
- An initialization function of the SWC is invoked by the Dem.

These dependencies must be described in terms of the AUTOSAR metamodel, which will contribute to the SWC Description of the application component as well as to the Service Component Description of the Dem Service.

The service component description of the Dem Service will define the ports below the RTE. Each SWC, which uses the Dem Service, must contain respective ports in its own

SWC description, which will be typed by the same (or compatible) interfaces and must be connected to the ports of the Dem, so that the RTE can be generated. Ids used in these Dem-ports are abstracted with portdefined arguments.

[SWS_Dem_00512]

Upstream requirements: SRS_BSW_00300

⌈The name of the Dem Service Component shall be “Dem”.⌋

The callbacks to SWCs do not use the mechanism of the portdefined arguments. Instead, the Dem configuration mechanism must ensure that the callback is delivered to the configured port and invokes the correct operation at this port using an RTE (direct or indirect) API call. For example, the EventId must not be passed as the first argument of the operation, because the monitors (and other SW-Cs) do not cope with EventIds explicitly.

In contrast to that, there are some special SWCs, that need to handle event status/data changes uniformly. These SWCs are notified by the Dem, which provides then also the EventId for the application, to be able, to request the respective data from the Dem. Therefore, general interfaces are provided (refer to `GeneralCallbackEventUdsStatusChanged`, `GeneralDiagnosticInfo` and `CallbackEventUdsStatusChanged`), which shall only be used in the way, that the EventIds (given by the Dem) shall not be interpreted by SWCs in any way, as well as the EventId values (used by the SW-Cs to request the event-specific data) have always to be provided by the Dem. Here only a handthrough mechanism is allowed to be used by SWCs for EventIds.

In the current Autosar release, the RTE analyzes the complete calltree of triggered runnables. Therefore, some limitations in the Dem API are introduced to avoid function calls from BSW modules, which rely on callbacks to SWCs via RTE.

7.11.2 Interaction with Diagnostic Clients

The `Dem` module provides interfaces to diagnostic clients. A diagnostic client is a `BSW` module, which is related to processing diagnostic requests. The most common diagnostic client is the `Dcm`. There are further used cases, where a diagnostic client is a `SW-C` interacting with the `Dem` via the `RTE` interfaces. A diagnostic client accesses the `Dem` module in order to delegate diagnostic services, which are related to the event memory.

For detailed description of the interface usage between `Dcm` and `Dem` consult the Dcm SWS document [9, SWS Dcm]. There, especially the handling of freeze frame data is described.

[SWS_Dem_00171]

Upstream requirements: RS_Diag_04058

⌈If the `Dcm` has requested an unavailable `event memory` / `DTC` origin, the `Dem` module functions with the respective return value shall return `DEM_WRONG_DTCORIGIN`.⌋

[SWS_Dem_00172] ⌈If the Dcm has requested a DTC that is not available at all or that is available but has a different event memory / DTC origin than the requested one, the Dem module functions with the respective return value shall return DEM_WRONG_DTC.⌋

The ClientID is also used to filter for specific contents (like UDS instances , J1939 nodes, OBD,..).

7.11.2.1 Parallel event memory access

There are multiple clients that can request access to the fault memory. To allow parallel access to the clients, the Dem uses the concept of a client ID. The client IDs are configured in the container DemClient. Each client which requests access to the fault memory of the Dem has an assigned unique client ID. Examples for these clients are:

- Parallel OBD and UDS communication, where each protocol is handled by an own Dcm client
- Multiple Dcm instances in virtual ECUs, where Dcm instances are independent from each other
- SWCs or complex device driver, as proxy in OBD systems for secondary ECUs
- Clearing DTC by different clients

The Dem supports the client id as parameter 'ClientId' in various APIs to:

- Access fault memory information
- Read event related data
- Clear DTC and fault memory
- Control fault memory behavior

[SWS_Dem_01251]

Upstream requirements: RS_Diag_04162

⌈The Dem shall ensure that clients with a different client Id can call all APIs with 'ClientId' as parameter in parallel. An API call from one client shall be processed independently from calls of the same API from a client with a different ID. E.g. if one client is setting a filter to iterate over extended data records, another client can set a different filter and iterate the extended data records.⌋

[SWS_Dem_01252]

Upstream requirements: RS_Diag_04162

⌈The Dem shall accept in all APIs with 'ClientId' as parameter only a ClientId value matching to any of the configured DemClientId. If a not configured ClientId value is passed, the Dem shall report the Det error DEM_E_WRONG_CONFIGURATION.⌋

Note: A client is a BSW/SW-C module or a CDD that is processing diagnostic requests or is used to support a distributed diagnostic environment such as events from a secondary ECU that are handled on a primary ECU. The client id concept is explicitly not designed to provide any arbitrary BSW module access to the fault memory, especially if the modules purpose is different than processing diagnostic requests. The design focuses on Dcm for OBD and UDS processing, as well as remote event communication modules for distributed OBD systems.

For each DemClient with the parameter DemClientUsesRte set to TRUE, the Dem will provide the C/S interfaces ClearDTC and EvMemOverflowIndication.

7.11.2.2 Accessing diagnostic fault memory

The Dem provides a range of APIs controlling and accessing event memory and its behavior or configuration and states. Many of those APIs follow a 'select and request' mechanism. Initially a client selects a DTC or a group of DTCs and then calls APIs working on the active selection. A selection is client local and remains until a new selection is made.

[SWS_Dem_01253] ⌈If the API Dem_SelectDTC is called, the Dem shall use the selected DTC or group of DTC in the requested DTCOrigin and DTCFormat as target for the following API calls:

- Dem_ClearDTC
- Dem_DisableDTCRecordUpdate
- Dem_GetDTCSelectionResultForClearDTC
- Dem_GetDTCSelectionResult
- Dem_GetFunctionalUnitOfDTC
- Dem_GetSeverityOfDTC
- Dem_GetStatusOfDTC
- Dem_SelectFreezeFrameData
- Dem_SelectExtendedDataRecord
- Dem_SetDTCSuppression
- Dem_GetDTCSuppression.

⌋

[SWS_Dem_01306] Behavior if one client calling Dem_SelectDTC before the previous operation finished ⌈If Dem_SelectDTC is called by one client and the same client is calling Dem_SelectDTC again and the Dem is currently still processing the

previous operation based on Dem_SelectDTC according to [SWS_Dem_01253], the Dem shall return DEM_BUSY.⌋

[SWS_Dem_01305] Behavior if one client starting a new operation while the previous one is still running ⌈In case any of the Dem_SelectDTC related APIs according to [SWS_Dem_01253] is called by one client and the same client has already started another Dem_SelectDTC related operation according to [SWS_Dem_01253] and this different operation is not completed, the Dem shall return DEM_BUSY.⌋

[SWS_Dem_01304] Life cycle of Dem_Select dependent operations ⌈The Dem shall consider an operation based on Dem_SelectDTC according to [SWS_Dem_01253] to be finished independently if the client has requested the E_OK result from the Dem or not.⌋

A client calling Dem_SelectDTC while another operation of the same client is still running, would indicate that the client is no longer interested in the result of the running operation. The result can be dropped and new select can be processed after Dem is ready.

[SWS_Dem_01198] ⌈For addressing all DTCs, the Dem shall provide the symbol DEM_DTC_GROUP_ALL_DTCS which is selecting all configured DTCs (representation is 0xFFFFFF).⌋

[SWS_Dem_01254] ⌈The API Dem_SelectDTC triggers the Dem internal DTC selection process on the event memory assigned to the ClientId. The result of the selection shall be reflected in the return value of the API calls following the Dem_SelectDTC call, see [SWS_Dem_01253].⌋

Note:The behaviour of Dem_SelectDTC depending APIs with calling Dem_SelectDTC is described in [SWS_Dem_01299].

Note: The Dem_SelectDTC selects internally a DTC or group of DTC. In case a client is interested only in the result of the selection without a further action behind, the API Dem_GetDTCSelectionResult can be used. One use case of this API is the NRC handling for the UDS service 0x14 ClearDTC. The Dcm needs to perform further NRC relevant verifications between selecting and deleting the DTC.

[SWS_Dem_01372] Selection of 3 byte OBD DTCs

Upstream requirements: RS_Diag_04253

⌈If Dem_SelectDTC is called with DTCFormat equal to DEM_DTC_FORMAT_OBD_3BYTE, the Dem shall select only DTCs based on the 3 byte OBD Format defined by DemDtcValue3Byte.⌋

[SWS_Dem_01261] API Behaviour on undefined DTC or format

Upstream requirements: SRS_BSW_00369

⌈If any of the APIs requiring a Dem_SelectDTC according to [SWS_Dem_01253] is called while the selection with Dem_SelectDTC is invalid, then the Dem shall return

DEM_WRONG_DTC. A selection is invalid in case the provided DTC does not specify a valid DTC or GroupOfDTCs in the corresponding DTCFormat. The selection also is invalid, if Dem_SelectDTC was not called before.⌋

[SWS_Dem_01299] API behavior without selected DTC

Upstream requirements: SRS_BSW_00369

⌈If any of the APIs requiring a Dem_SelectDTC according to [SWS_Dem_01253] is called without a prior call of Dem_SelectDTC for this client, the Dem shall report the Det DEM_E_WRONG_CONDITION.⌋

[SWS_Dem_01262] API Behaviour on undefined DTCOrigin ⌈If any of the APIs requiring a Dem_SelectDTC (see [SWS_Dem_01253]) is called on a not existing selected DTCOrigin, the Dem shall return DEM_WRONG_DTCORIGIN on this API.⌋

[SWS_Dem_01373] API behavior on not supported OBD 3 byte DTC

Upstream requirements: RS_Diag_04253

⌈If DemSupportedObdUdsDtcSeparation is not configured or is set to FALSE and any Dem API having DTC format as a parameter is called with DTCFormat set to DEM_DTC_FORMAT_OBD_3BYTE, the Dem shall report the Det DEM_E_PARAM_DATA.⌋

[SWS_Dem_01256] ⌈The API Dem_GetDTCSelectionResult shall return E_OK, if the selection of Dem_SelectDTC was successful and the DTC or group of DTC are ready for further processing.⌋

Note: The call of Dem_GetDTCSelectionResult after a Dem_SelectDTC is optional. APIs depending on Dem_SelectDTC have the same behaviour with or without a call of Dem_GetDTCSelectionResult.

[SWS_Dem_01257] ⌈If the API Dem_GetDTCSelectionResult is called and the DTC or group of DTC from Dem_SelectDTC is invalid, the Dem shall return DEM_WRONG_DTC.⌋

[SWS_Dem_01258] ⌈If the API Dem_GetDTCSelectionResult is called and the Origin provided to Dem_SelectDTC is invalid, the Dem shall return DEM_WRONG_DTCORIGIN.⌋

[SWS_Dem_01296] Definition of Dem_GetDTCSelectionResultForClearDTC functionality ⌈The API Dem_GetDTCSelectionResultForClearDTC shall behave exactly as Dem_GetDTCSelectionResult with the only difference that, DEM_WRONG_DTC is returned in case of a single DTC was selected and DemClearDTCLimitation is set to DEM_ONLY_CLEAR_ALL_DTCS for that client.⌋

7.11.2.3 Access DTCs and Status Information

The following chapter defines the APIs, which shall be used to access the number of DTCs, and DTCs matching specific filter criteria and the associated status information.

[SWS_Dem_00231]

Upstream requirements: RS_Diag_04201

⌈The API Dem_GetTranslationType shall return the translation format of the Dem-EventMemorySet configured in DemTypeOfDTCSupported.⌋

[SWS_Dem_00060]

Upstream requirements: RS_Diag_04067

⌈The function Dem_GetDTCStatusAvailabilityMask shall set the out parameter DTCStatusMask to the value configured in DemDtcStatusAvailabilityMask of the DemEventMemorySet for the assigned client ID.⌋

[SWS_Dem_01342]

Upstream requirements: RS_Diag_04067

⌈If the function Dem_GetDTCStatusAvailabilityMask is called with a DTCOrigin different to DEM_DTC_ORIGIN_PRIMARY_MEMORY or DEM_DTC_ORIGIN_USERDEFINED_MEMORY_<Name>, the Dem shall report the development error DEM_E_PARAM_DATA.⌋

[SWS_Dem_01181] ⌈The Dem module shall be able to return the DTC severity availability mask (refer to Dem_GetDTCSeverityAvailabilityMask) in accordance to ISO 14229-1 [1].⌋

[SWS_Dem_01182] ⌈The function Dem_GetDTCSeverityAvailabilityMask shall calculate the DTC severity availability mask in accordance to ISO 14229-1 [1] from the severity configuration of each supported DTC.⌋

[SWS_Dem_00657] ⌈The Dem module shall mask all DTC status bytes (API parameters DTCStatus) provided to the Dcm with the DTC status availability mask (by performing a bit-wise AND operation).⌋

[SWS_Dem_00059]

Upstream requirements: RS_Diag_04067

⌈The function Dem_GetStatusOfDTC shall copy the status of the selected DTC to the parameter DTCStatus. The DTCStatus format is according to ISO-14229-1[2].⌋

[SWS_Dem_01259] ⌈If the API Dem_GetStatusOfDTC is called on a selected DTC that does not exist in the selected format or if the selected DTC is a group of DTC or no DTC was selected, the Dem shall return DEM_WRONG_DTC.⌋

[SWS_Dem_01260] ⌈If the API Dem_GetStatusOfDTC is called on a not existing selected DTCOrigin, the Dem shall return DEM_WRONG_DTCORIGIN.⌋

[SWS_Dem_01275] ⌈If the API Dem_GetStatusOfDTC is called on a selected DTC that does not have an assigned DTC status, the Dem shall return DEM_NO_SUCH_ELEMENT.⌋

[SWS_Dem_00057]

Upstream requirements: RS_Diag_04057

⌈The function Dem_SetDTCFilter shall set the filter criteria for the client with the same client ID for subsequent calls of::

- Dem_GetNumberOfFilteredDTC
- Dem_GetNextFilteredDTC
- Dem_GetNextFilteredDTCAndFDC
- Dem_GetNextFilteredDTCAndSeverity.

⌋

Note: A DTC filter for a different ClientID can be set in parallel (see [SWS_Dem_01251]) without interfering each other. This way multiple clients can access the fault memory data in parallel.

In figure 7.47 a situation is shown, where two clients (e.g. an Dcm for OBD and an Dcm for UDS) are reading the number of filtered DTCs in parallel.

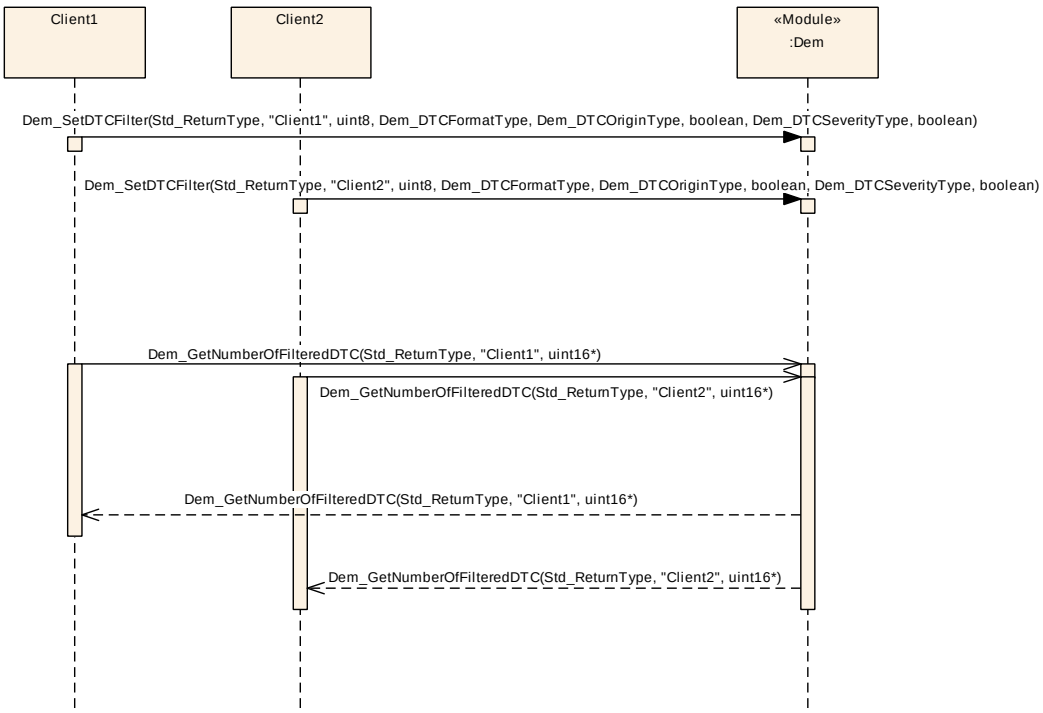


Figure 7.47: Multiple clients requesting event memory information in parallel

[SWS_Dem_01374] Selecting OBD 3 Byte DTCs for readiness group filter

Upstream requirements: RS_Diag_04253

⌈If Dem_SetDTCFilterByReadinessGroup is called with DTCFormat equal to DEM_DTC_FORMAT_OBD_3BYTE and DemSupportedObdUdsDtcSeparation is set to True, the Dem shall use the DTCs configured via DemDtcValue3Byte as filter target.⌋

[SWS_Dem_01375] Selecting UDS DTCs for readiness group filter

Upstream requirements: RS_Diag_04253

⌈If Dem_SetDTCFilterByReadinessGroup is called with DTCFormat equal to DEM_DTC_FORMAT_UDS and DemSupportedObdUdsDtcSeparation is set to FALSE, the Dem shall use the 3-byte UDS DTC configured via DemDtcValue as filter target.⌋

[SWS_Dem_01376] Selecting not supported DTC format for readiness group filter

Upstream requirements: RS_Diag_04253

⌈If Dem_SetDTCFilterByReadinessGroup is called with DTCFormat equal to DEM_DTC_FORMAT_UDS and DemSupportedObdUdsDtcSeparation is set to TRUE, the Dem shall report the Det DEM_E_PARAM_DATA.⌋

[SWS_Dem_01377] Selecting OBD 3 Byte DTCs with extended data records

Upstream requirements: RS_Diag_04253

⌈If Dem_SetDTCFilterByExtendedDataRecordNumber is called with DTCFormat equal to DEM_DTC_FORMAT_OBD_3BYTE and DemSupportedObdUdsDtcSeparation is set to True, the Dem shall use the DTCs configured via DemDtcValue3Byte as filter target.⌋

[SWS_Dem_01378] Selecting UDS DTCs with extended data records

Upstream requirements: RS_Diag_04253

⌈If Dem_SetDTCFilterByExtendedDataRecordNumber is called with DTCFormat equal to DEM_DTC_FORMAT_UDS, the Dem shall use the 3-byte UDS DTC configured via DemDtcValue as filter target.⌋

[SWS_Dem_01362] Functionality of Dem_SetDTCFilterByExtendedDataRecordNumber

Upstream requirements: RS_Diag_04057, RS_Diag_04250

⌈The function Dem_SetDTCFilterByExtendedDataRecordNumber shall set a DTC filter criteria for supported ExtendedDataRecords on DTCs based on the parameter ExtendedDataRecordNumber for the client with the same client ID for subsequent calls of:

- Dem_GetNumberOfFilteredDTC
- Dem_GetNextFilteredDTC.

An extended data record for a DTC is considered as supported if the DTC refers to an extended data record that has DemExtendedDataRecordNumber set to the number of the extended data record.⌋

The sequence diagram in chapter 9.12 depicts the interaction of Dem_SetDTCFilterByExtendedDataRecordNumber and Dem_GetNextFilteredDTC for UDS service processing.

[SWS_Dem_01364] Functionality of Dem_SetDTCFilterByReadinessGroup

Upstream requirements: RS_Diag_04057, RS_Diag_04250

⌈The function Dem_SetDTCFilterByReadinessGroup shall set a DTC filter criteria for DTCs that have DemDtcOBDRelevance set to true and are assigned to a given DTC readiness group for the client with the same client ID for subsequent calls of:

- Dem_GetNumberOfFilteredDTC
- Dem_GetNextFilteredDTC.

⌋

[SWS_Dem_01363] Life cycle of set filter criteria

Upstream requirements: RS_Diag_04057, RS_Diag_04250

⌈DTC filter criteria set via

- Dem_SetDTCFilter
- Dem_SetDTCFilterByExtendedDataRecordNumber
- Dem_SetDTCFilterByReadinessGroup

shall be valid until the next call of any of these APIs by the same client or Dem initialization.⌋

This chapter defines filters on DTCs and data retrieval based on the filter. A DTC filter is set by one of the APIs:

- Dem_SetDTCFilter
- Dem_SetDTCFilterByExtendedDataRecordNumber
- Dem_SetDTCFilterByReadinessGroup

The term "DTC filter" is used in requirements and in this chapter and defines that a client has set a DTC filter by calling any of these APIs.

[SWS_Dem_01300] API behavior without set DTC filter

Upstream requirements: SRS_BSW_00369

⌈If any of the APIs depending on a set DTC filter is called without a set DTC filter for this client, the Dem shall report the Det DEM_E_WRONG_CONDITION.⌋

[SWS_Dem_01266] ⌈If any of the APIs depending on a set DTC filter is called while the call to set the DTC filter did not return E_OK, the Dem shall return E_NOT_OK on this subsequent function call.⌋

[SWS_Dem_01263]

Upstream requirements: RS_Diag_04162

⌈A call of a DTC filter setting API

- Dem_SetDTCFilter
- Dem_SetDTCFilterByExtendedDataRecordNumber
- Dem_SetDTCFilterByReadinessGroup

shall set the requested filter on the event memory set assigned to ClientID of the call.⌋

[SWS_Dem_00649] Reset the filter/data retrieval sequence

Upstream requirements: RS_Diag_04057

⌈Each call of a DTC filter setting API

- Dem_SetDTCFilter
- Dem_SetDTCFilterByExtendedDataRecordNumber
- Dem_SetDTCFilterByReadinessGroup

shall lead to a reset of the sequence for that client.⌋

[SWS_Dem_01058] ⌈The function Dem_SetDTCFilter shall ignore unsupported bits retrieved in DTCStatusMask.⌋

The parameter DTCFormat of Dem_SetDTCFilter is relevant to decide about specific protocol characteristics like OBD vs. UDS vs. J1939. The DTCOrigin only references the fault memory location.

[SWS_Dem_01066]

Upstream requirements: RS_Diag_04057

⌈The Det error DEM_E_WRONG_CONFIGURATION shall be reported if the function Dem_SetDTCFilter is called with a value of the parameter DTCFormat that is not supported per configuration (e.g. if DTCFormat = DEM_DTC_FORMAT_OBD is passed, but OBD is not supported per configuration).⌋

[SWS_Dem_01067] ⌈The Det error DEM_E_WRONG_CONFIGURATION shall be reported if the function Dem_SetDTCFilter is called with a value of the parameter DTCOrigin that is not supported per configuration.⌋

[SWS_Dem_00061] Number of DTCs matching a DTC filter criteria ⌈If Dem_GetNumberOfFilteredDTC is called, the Dem shall provide the number of DTCs matching the filter criteria defined by a DTC filter call within the out-parameter NumberOfFilteredDTC.⌋

[SWS_Dem_00216] ⌈With each call to the function Dem_GetNextFilteredDTC the Dem module shall return the next DTC and its associated status matching a set DTC filter.⌋

A client can call Dem_GetNextFilteredDTC continuously to receive all DTCs matching the filter criteria. When the return value of the function is DEM_NO_SUCH_ELEMENT no further DTC matching the filter criteria is available and the client has retrieved all the filtered data.

[SWS_Dem_00653] ⌈

The API Dem_GetNextFilteredDTC shall not return the value DEM_PENDING when DTCFormat = DEM_DTC_FORMAT_OBD (indicating emission-related services).⌋

Rationale: The API Dem_GetNextFilteredDTC is used for UDS service 0x19 as well as OBD services $03/$07/$0A (refer to [SWS_Dem_00301]). UDS service 0x19 is allowed to respond with NRC 0x78 (response pending) to the tester, while the OBD services are not.

[SWS_Dem_00228] ⌈With each call to the function Dem_GetNextFilteredDTCAndFDC the Dem module shall return the next DTC and its associated fault detection counter (FDC, refer to [SWS_Dem_00264]) matching the filter criteria defined by the function call Dem_SetDTCFilter (refer to [SWS_Dem_00057]).⌋

[SWS_Dem_00513] ⌈If the callback-function Dem_GetFaultDetectionCounter returns other than E_OK or is not configured for a DTC filtered by Dem_GetNextFilteredDTCAndFDC, this FDC value shall be returned as 0.⌋

The Dcm calls the function Dem_GetNextFilteredDTCAndFDC continuously until the return value of the function is DEM_NO_SUCH_ELEMENT, to receive all DTCs matching the filter criteria.

Note: Non-Dem-internal calculated fault detection counters are typically requested from SWCs through the RTE. To indicate an equivalent calltree for these runnables, a workaround is used: The Dcm main function specifies a trigger to the Dem interface GeneralEvtInfo (operation GetFaultDetectionCounter), which triggers the respective runnable (refer to RunnableEntity Dem_GetFaultDetectionCounter).

[SWS_Dem_00287] ⌈With each call to the function Dem_GetNextFilteredDTCAndSeverity the Dem module shall return the next DTC and its associated fault severity matching the filter criteria defined by the function call Dem_SetDTCFilter (refer to [SWS_Dem_00057]).⌋

The Dcm calls the function Dem_GetNextFilteredDTCAndSeverity continuously until the return value of the function is DEM_NO_SUCH_ELEMENT, to receive all DTCs matching the filter criteria.

For information about the functions Dem_GetSeverityOfDTC and Dem_GetFunctionalUnitOfDTC, refer to chapter 7.4.

[SWS_Dem_00595] ⌈The function Dem_SetFreezeFrameRecordFilter shall set the static filter criteria attribute "all freeze frame records currently stored in the event memory" to be used for the subsequent calls of Dem_GetNextFilteredRecord and Dem_GetNumberOfFreezeFrameRecords. This filter always belongs to primary memory.⌋

[SWS_Dem_00650]

Upstream requirements: RS_Diag_04205

⌈The filter criteria attributes of Dem_SetFreezeFrameRecordFilter shall be used until the next call of Dem_SetFreezeFrameRecordFilter or Dem initialization. Each call of Dem_SetFreezeFrameRecordFilter shall lead to a reset of the sequence iterating over freezeframes.⌋

[SWS_Dem_01312]

Upstream requirements: RS_Diag_04205

⌈Dem_GetNumberOfFreezeFrameRecords shall retrieve the number of freeze frame records in primary memory that match the current filter.⌋

[SWS_Dem_01313] API behavior without set DTC filter

Upstream requirements: SRS_BSW_00369

⌈If any of the APIs requiring a Dem_SetFreezeFrameRecordFilter is called without a prior call of Dem_SetFreezeFrameRecordFilter for this client, the Dem shall report the Det DEM_E_WRONG_CONDITION.⌋

[SWS_Dem_00225] ⌈With each call to the function Dem_GetNextFilteredRecord the Dem module shall return the next DTC and its associated relative addressed freeze frame record numbers matching the filter criteria defined by the function call Dem_SetFreezeFrameRecordFilter.⌋

The client calls the function Dem_GetNextFilteredRecord continuously until the return value of the function is DEM_NO_SUCH_ELEMENT, to receive all records matching the filter criteria.

[SWS_Dem_00219]

Upstream requirements: RS_Diag_04195

⌈The function Dem_GetDTCByOccurrenceTime shall provide the capability to get one DTC stored in the primary event memory of the addressed DemEventMemorySet according to the API parameter DTCRequest.⌋

[SWS_Dem_00221]

Upstream requirements: RS_Diag_04195

⌈The API Dem_GetDTCByOccurrenceTime shall return DEM_NO_SUCH_ELEMENT, if no DTC is matching the requested occurrence time in the parameter DTCRequest.⌋

7.11.2.4 Access event related data

This section defines the APIs, to get access to the event related data (refer to chapter 7.7.7.1 and chapter 7.7.7.3) stored with the DTCs in the event memory of the Dem via diagnostics. Furthermore, access to WWH-OBD relevant PIDs stored in a freeze frame is made available. Details concerning freeze frame handling can be found in ISO-14229-1[2] and ISO 27145-3[21].

Example for the mapping of a PID to a DID: PID $04 –> DID 0xF404

Note: The data returned include the DTCSnapshotRecordNumberOfIdentifiers (Num of DIDs) as defined by ISO-14229-1[2] for the response message to service 0x19 0x05.

[SWS_Dem_01268] ⌈If the API Dem_SelectFreezeFrameData is called, the Dem shall use the DTC selected by Dem_SelectDTC and the RecordNumber as a target for the following API calls:

- Dem_GetNextFreezeFrameData
- Dem_GetSizeOfFreezeFrameSelection

⌋

[SWS_Dem_01269] ⌈The API Dem_SelectFreezeFrameData triggers the Dem internal Freeze Frame selection process on the event memory assigned to the ClientId. The Dem shall provide the result of the selection process in the return value of the API calls following the Dem_SelectFreezeFrameData call, see [SWS_Dem_01268].⌋

[SWS_Dem_01270]

Upstream requirements: SRS_BSW_00369

⌈If any of the APIs requiring a Dem_SelectFreezeFrameData (see [SWS_Dem_01268]) are called without a prior call to Dem_SelectFreezeFrameData, the Dem shall report the Det error DET_E_WRONG_CONDITION.⌋

[SWS_Dem_01271]

Upstream requirements: SRS_BSW_00369

⌈If any of the APIs requiring a Dem_SelectFreezeFrameData is called without a prior call to Dem_SelectFreezeFrameData, the Dem shall report the Det error DEM_E_WRONG_CONDITION.⌋

[SWS_Dem_00071] ⌈The function Dem_GetNextFreezeFrameData shall copy the specific data identifiers (DIDs) of the freezeframe record selected by the API Dem_SelectFreezeFrameData for the specified ClientId to the destination buffer provided by the DestBuffer parameter. The Dem shall transmit these data as a complete record with the Buffer format.⌋

For Buffer format see figure 7.48.

[SWS_Dem_00576] ⌈If the RecordNumber selected by the API Dem_SelectFreeze-FrameData is set to 0x00, Dem_GetNextFreezeFrameData shall provide the event/DTC specific OBD equivalent freeze frame record.⌋

Note: Retrieving record 0x00 with Dem_SelectFreezeFrameData and Dem_Get-NextFreezeFrameData is only supported if the Dem module supports OBD (refer to DemOBDSupport)

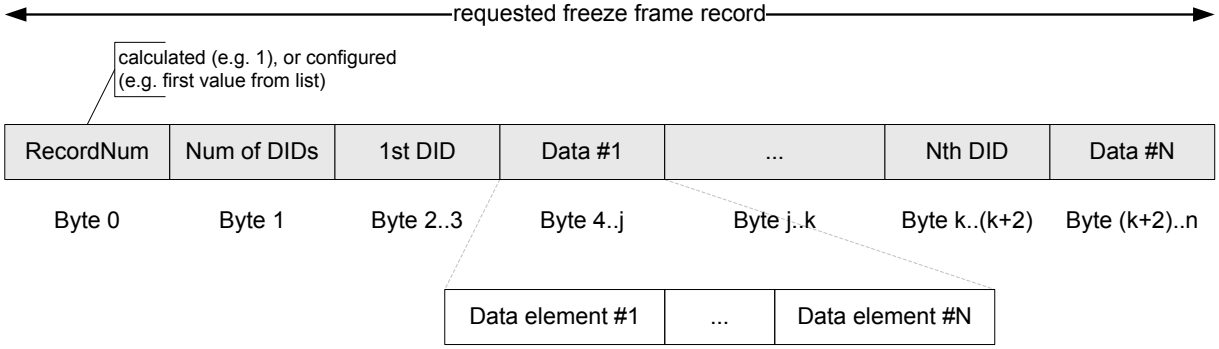


Figure 7.48: Buffer format used by Dem_GetNextFreezeFrameData

Note: The data returned include the DTCSnapshotRecordNumber (RecordNum) and DTCSnapshotRecordNumberOfIdentifiers (Num of DIDs) as defined by ISO-14229-1[2] for the response message to service 0x19 0x04. In contradiction to the Dcm PIDstructure handling, the DIDstructure is handled within the Dem module. Therefore, the Dem returns already full freeze frame records according to the response layout. This results in an optimized Dem/Dcm interface.

[SWS_Dem_00630] ⌈If a single DTC is selected and the API Dem_SelectFreeze-FrameData is called with a valid RecordNumber which is not stored, Dem_GetNext-FreezeFrameData shall return E_OK and BufSize 0 (empty buffer).⌋

[SWS_Dem_00074]

Upstream requirements: RS_Diag_04205

⌈The function Dem_GetSizeOfFreezeFrameSelection shall return the size of the requested freeze frame record(s) by Dem_SelectFreezeFrameData for the specified ClientId, which represents the number of user data bytes, including any freeze frame headerinformation (according to the format defined in [SWS_Dem_00071]).⌋

Note: If the record number value 0xFF is requested, the Dem considers the size of all stored freeze frame records according to [SWS_Dem_00074].

[SWS_Dem_01389] Requested size for not configured snapshot records

Upstream requirements: RS_Diag_04205

⌈If Dem_GetSizeOfFreezeFrameSelection is called and the selected freeze frame record number is not configured, the Dem shall return DEM_NO_SUCH_ELEMENT.⌋

[SWS_Dem_01348] Select freeze frame data for single DTCs only

Upstream requirements: RS_Diag_04244

⌈In case the function Dem_SelectFreezeFrameData is called and the DTC selected by Dem_SelectDTC is not a single DTC but a group of DTC, the Dem shall return DEM_WRONG_DTC.⌋

[SWS_Dem_01349] Select extended data record for single DTCs only

Upstream requirements: RS_Diag_04245

⌈In case the function Dem_SelectExtendedDataRecord is called and the DTC selected by Dem_SelectDTC is not a single DTC but a group of DTC, the Dem shall return DEM_WRONG_DTC.⌋

[SWS_Dem_01272] ⌈If the API Dem_SelectExtendedDataRecord is called, the Dem shall use the DTC selected by Dem_SelectDTC and the ExtendedDataNumber as a target for the following API calls:

- Dem_GetNextExtendedDataRecord
- Dem_GetSizeOfExtendedDataRecordSelection

⌋

[SWS_Dem_01273] ⌈The API Dem_SelectExtendedDataRecord triggers the Dem internal extended data record selection process on the event memory assigned to the ClientId. The Dem shall provide the result of the selection process in the return value of the API calls following the Dem_SelectExtendedDataRecord, see [SWS_Dem_01272].⌋

[SWS_Dem_01274]

Upstream requirements: SRS_BSW_00369

⌈If any of the APIs requiring a Dem_SelectExtendedDataRecord (see [SWS_Dem_01272]) are called without a prior call to Dem_SelectExtendedDataRecord, the Dem shall report the Det error DEM_E_WRONG_CONDITION.⌋

[SWS_Dem_00075] ⌈The API Dem_GetNextExtendedDataRecord shall copy the complete data of the requested extended data record by Dem_SelectExtended-DataRecord API for the specified ClientId to the destination buffer (DestBuffer). The

function shall transmit these data as a complete record with the Buffer format. The extended data record number is placed as the first byte in the copied data.⌋

For Buffer format see figure 7.49.

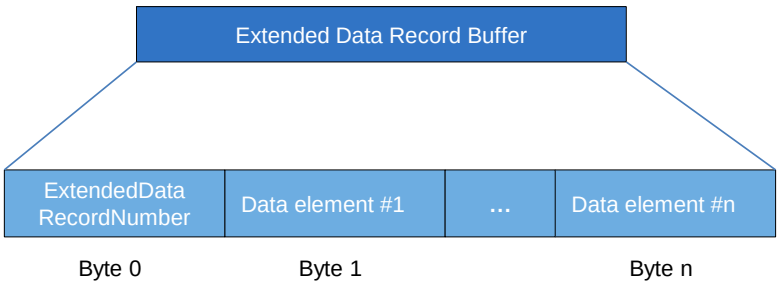


Figure 7.49: Buffer format used by Dem_GetNextExtendedDataRecord

[SWS_Dem_00631] ⌈If the API Dem_SelectExtendedDataRecord is called with a valid DTC and a valid extended data record number which is not stored, the API shall return E_OK and Dem_GetNextExtendedDataRecord shall return BufSize 0 (empty buffer).⌋

[SWS_Dem_00076] ⌈The function Dem_GetSizeOfExtendedDataRecord-Selection shall return the size of the requested 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, different to the format defined in [SWS_Dem_00075]).⌋

ISO 14229-1 defines the DTCExtendedDataRecordNumber wildcards 0xFE and 0xFF to request multiple extended data records with a single UDS request. The Dem always supports these extended data records without requiring explicit configuration of them.

[SWS_Dem_01398] Support of DTCExtendedDataRecordNumber 0xFE

Upstream requirements: RS_Diag_04206

⌈The Dem shall always support DTCExtendedDataRecordNumber 0xFE for the primary fault memory in case at least one extended data record in the range 0x90-0x9F is supported (which means regulatory emission OBD is used) and return all regulated emission OBD stored extended data records in the range of 0x90 to 0x9F.⌋

[SWS_Dem_01399] Support of DTCExtendedDataRecordNumber 0xFF in primary fault memory

Upstream requirements: RS_Diag_04206

⌈The Dem shall always support DTCExtendedDataRecordNumber 0xFF for the primary fault memory and return all stored extended data records in the range of 0x01 to 0xEF.⌋

[SWS_Dem_01400] Support of DTCExtendedDataRecordNumber 0xFF in user defined fault memory

Upstream requirements: RS_Diag_04206

⌈The Dem shall always support DTCExtendedDataRecordNumber 0xFF for an user defined fault memory and return all stored extended data records in the range of 0x01 to 0xFE.⌋

The Dcm uses the function Dem_DisableDTCRecordUpdate, if the freeze frame or extended data of a specific DTC is about to be accessed by subsequent API calls. This is done to ensure, that the requested data of this DTC are not changed while the Dcm accesses those by several APIcalls.

[SWS_Dem_00270] ⌈The function Dem_DisableDTCRecordUpdate shall protect the event related data of the selected DTC within the selected DTCOrigin from updating or deleting, to allow a consistent read for the following subsequent APIcalls:

- Dem_SelectFreezeFrameData
  Dem_GetSizeOfFreezeFrameSelection and
  Dem_GetNextFreezeFrameData
- Dem_SelectExtendedDataRecord
  Dem_GetSizeOfExtendedDataRecordSelection and
  Dem_GetNextExtendedDataRecord.

⌋

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.

Note: Event related data might still be updated in background (e.g. Dem-internal data elements).

Note: The function Dem_DisableDTCRecordUpdate does not affect the UDS status information update.

[SWS_Dem_00271] ⌈The function Dem_EnableDTCRecordUpdate shall release the currently disabled DTC which has been protected by the function Dem_DisableDTCRecordUpdate, so that the data can be updated again.⌋

The function Dem_EnableDTCRecordUpdate is the counterpart to the function Dem_DisableDTCRecordUpdate.

The Dcm calls the function Dem_EnableDTCRecordUpdate after the freeze frame and extended data of the specific DTC were protected by the function Dem_DisableDTCRecordUpdate, after the access by subsequent APIcalls is finished.

[SWS_Dem_00648] ⌈If development error detection is enabled and the function Dem_DisableDTCRecordUpdate is called while another DTC is locked by the same client, the Dem module shall report the Det DEM_E_WRONG_CONDITION.⌋

7.11.2.5 Clear diagnostic information

The Dem provides the functionality to clear DTCs to the Dcm and other diagnostic clients. The general DTC clearing process is described in chapter 7.7.2.2.

7.11.2.6 Control DTC setting

[SWS_Dem_01290] DTCs affected by Dem_DisableDTCSetting and Dem_EnableDTCSetting ⌈Calls of the APIs Dem_DisableDTCSetting and Dem_EnableDTCSetting have effect on all events and UDS status bytes assigned the DemEventMemorySet referenced by the clientId of the call. DTCs not referenced from this DemEventMemorySet are not affected.⌋

[SWS_Dem_00079] ⌈If the function Dem_DisableDTCSetting is called, the Dem shall disable the storage of corresponding events and UDS status byte updates in primary memory and all user defined memories that do have DemAffectedBy-ControlDTCSetting not configured or set to true.⌋

The function Dem_DisableDTCSetting is used in case of an induced failure situation in a system, e.g. during flashreprogramming of one ECU in a network. In that case all the ECUs are commanded via diagnostic request (forwarded from Dcm by using Dem_DisableDTCSetting / Dem_EnableDTCSetting) to ignore DTC reports, as the flashed ECU is not participating in the normal communication anymore.

Note: If one of the other networked ECUs needs one of the signals, which are now missing, for this case a failsafereaction of the ECU cannot be assigned to the UDS status byte updates, as these are also suppressed during disabled DTC setting.

[SWS_Dem_00080]

Upstream requirements: RS_Diag_04115

⌈If the function Dem_EnableDTCSetting is called, the Dem shall enable the storage of corresponding events and UDS status byte.⌋

[SWS_Dem_00626]

Upstream requirements: RS_Diag_04115

⌈When DTC setting is disabled, all status reports from the API Dem_SetEventStatus for the related events shall be ignored (no change of UDS status byte) by the Dem.⌋

Note: This is similar like the enable condition handling (refer to [SWS_Dem_00449]). It has no impact on Dem_ResetEventDebounceStatus, Dem_ResetEventStatus

and Dem_ClearDTC. In case of Dem-internal debouncing the related fault detection counter will be frozen or reset (refer to chapter Figure 7.30 and Figure 7.33).

7.11.2.7 Asynchronous Dcm operations

Most of the APIs of the Dem/Dcm interface depend on NVRAM data. Therefore, an asynchronous processing of these APIs is realized by using the additional "DEM_PENDING" return value.

7.11.3 Interaction with J1939 Diagnostic Manager (J1939)

[SWS_Dem_00971]

Upstream requirements: RS_Diag_04112

⌈In case the TestFailed bit of an event within an primary event memory changes and a J1939 DTC number is assigned to that event, the function J1939Dcm_DemTriggerOnDTCStatus shall be called for each DemClient referencing this DemEventMemorySet⌋

7.11.4 Interaction with Function Inhibition Manager (FIM)

The purpose of the Function Inhibition Manager is to control (enable/disable) function entities within software components based on inhibit conditions such as detected errors. The Dem contribution to this functionality is to provide the monitoring status change and the DemComponent information to the FiM.

[SWS_Dem_00029]

Upstream requirements: RS_Diag_04031

⌈If DemTriggerFiMReports is set to True, the Dem module shall notify the FiM module (refer to [8]) on each change of the monitoring status (refer also to [SWS_Dem_00016]), by calling the function FiM_DemTriggerOnMonitorStatus in the context of Dem_SetEventStatus.⌋

The Fim uses the function Dem_GetMonitorStatus for possible plausibility checks, rebuilding, etc. of inhibition relations.

[SWS_Dem_01325]

Upstream requirements: RS_Diag_04031

⌈The Dem shall call FiM_DemInit after the monitor and component states have been initialized.⌋

[SWS_Dem_01189]

Upstream requirements: RS_Diag_04031

⌈If DemTriggerFiMReports is set to True, the Dem shall call FiM_DemInit during Dem_Init to trigger the initialization of the permissions inside the Fim(independent of trigger or polling mode). The FiM_DemInit shall not be called at any other point in time.⌋

7.11.5 Interaction with NVRAM Manager (NvM)

Typically, the Dem module uses nonvolatile memory blocks (configurable in size by the NVRAM Manager [5]) to achieve permanent storage of UDS status information, event related data and required internal states (e.g. retrieve status at startup). Each nonvolatile memory block used from the Dem, needs also to be configured (refer to DemNvRamBlockId). The number, the type, and the content of the used nonvolatile memory blocks are not prescribed. These shall be handled implementation specific. The NvM usage can also be deactivated by configuration (refer to multiplicity of DemNvRamBlockId), so that the Dem will work based on RAM only.

[SWS_Dem_00339]

Upstream requirements: RS_Diag_04107

⌈The Dem module shall verify the validity (which relates to block states), integrity (which relates to CRC results), and for general NvM-reading errors of its nonvolatile blocks (before using the respective data).⌋

Usually this verification is done in the API Dem_Init by using NvM_GetErrorStatus for these blocks, which are read by the ECU State Manager (refer to API NvM_ReadAll).

Note: For the non-volatile data of the Dem module, it is recommended to configure a CRC in the NvM.

[SWS_Dem_00578]

Upstream requirements: SRS_Mem_08549

⌈If the NVM module was not able to read some nonvolatile data of the Dem module, the Dem module shall initialize all non-volatile data with their initial values.⌋

Note: To avoid inconsistencies between readable blocks and erroneous blocks, all nonvolatile data are initialized. The initialization is done to allow the fault detection mechanism of the NvM module, to report the respective reading error(s) to the Dem module (refer to Dem_SetEventStatus). These errors denote the defective NVRAM.

[SWS_Dem_00340]

Upstream requirements: SRS_BSW_00101

⌈After the API Dem_Init has finished, the Dem shall be fully operational.⌋

[SWS_Dem_01237] Immediate storage of event related data at first occurrence

Upstream requirements: RS_Diag_04211

⌈If a DTC is configured with DemNvStorageStrategy = IMMEDIATE_AT_FIRST_OCCURRENCE, the Dem shall trigger the storage of its event related data to NvM when the configured trigger DemEventMemoryEntryStorageTrigger to store the data occurs the first time.⌋

Note: For DTCs that are stored immediately at first occurrence, it is necessary to ensure data consistency (e.g. with the UDS status byte) during Dem_Init.

Note: Following the storage at first occurrence, any further changes in the event related data will be stored during shutdown (refer to [SWS_Dem_00341])

[SWS_Dem_01238] Storage of event related data at shutdown

Upstream requirements: RS_Diag_04211

⌈If a DTC is configured with DemNvStorageStrategy = DURING_SHUTDOWN or DemNvStorageStrategy is not configured, the Dem shall trigger the storage of the event related data to NvM only during shutdown. (refer to [SWS_Dem_00341])⌋

[SWS_Dem_00102]

Upstream requirements: SRS_BSW_00336

⌈The API Dem_Shutdown shall finalize all pending operations in the Dem module to prepare the internal states and data for transfer to the NVRAM. The event memory shall be locked and not modified until the API Dem_Init is recalled.⌋

[SWS_Dem_00341]

Upstream requirements: SRS_BSW_00336, RS_Diag_04211

⌈The Dem shall trigger the storage of its changed nonvolatile data to NVRAM during Dem_Shutdown.⌋

Based on the Dem configuration and implementation, the copying process to NVRAM of those Demrelated NvMblocks to be stored after Dem_Shutdown, is performed by the API NvM_WriteAll called by the ECU State Manager.

If the ECU power supply is disconnected before the NvM has finished copying all data to NVRAM, these data will be incomplete/inconsistent or not stored. At next startup, the events of the last operating cycle could not be found anymore. Therefore, the NVRAM Manager configuration provides mechanisms for data consistency, like redundant data blocks.

[SWS_Dem_00164]

Upstream requirements: RS_Diag_04077

⌈The Dem module shall use the APIs NvM_WriteBlock and NvM_ReadBlock of the NVRAM Manager, if there is the necessity to store and restore data between Dem_Init and Dem_Shutdown.⌋

Note: The NvM module realizes a retry mechanism for block reading and writing. Therefore, the Dem module does not implement any retry mechanism for its nonvolatile blocks.

[SWS_Dem_00579]

Upstream requirements: SRS_BSW_00171

⌈If the NVM module was not able to write (some) nonvolatile data of the Dem module, the Dem module shall ignore the reported negative return values by the NvM.⌋

Note: If writing of nonvolatile Dem data fails, the Dem module is not able to perform any adequate reaction.

7.11.5.1 Initialize NvM Data

Reprogramming of an ECU with an updated DEM configuration may lead to inconsistencies of the integrity and validity of Dem related data stored in NvM. (e.g. Ids stored in non-volatile Dem related data might not match to the reprogrammed DEM configuration)

`Dem` is providing the functions `Dem_InitAllNvmBlocks` and `Dem_InitAllNvmBlocksGetResult` to support a non-AUTOSAR-specified handler/manager to resolve those validity issues by initializing all Dem related NvM data blocks.

[SWS_Dem_01420] Acceptance of Requests During the Valid Initialization Sequence

Upstream requirements: RS_Diag_04211

⌈If the function `Dem_InitAllNvmBlocks` is called after `Dem_PreInit` and before `Dem_Init`, the Dem shall accept the request and return `E_OK`.⌋

[SWS_Dem_01421] Handling of Requests Outside the Valid Initialization Sequence

Upstream requirements: RS_Diag_04211

⌈If the function `Dem_InitAllNvmBlocks` is called before `Dem_PreInit` or after `Dem_Init`, the `Dem` shall not accept the request and shall return `E_NOT_OK`.⌋

[SWS_Dem_01422] Initialization and Storage Actions for Accepted Non-Volatile Memory Requests

Upstream requirements: RS_Diag_04211

⌈If a `Dem_InitAllNvmBlocks` request was accepted before `Dem_Init`, the `Dem` shall do the following actions during execution of `Dem_Init`:

- Reset the volatile data buffers of all the `Dem` related NvM data blocks to initial values.

- Trigger an immediate NvM storage request to store the updated volatile data buffers in NvM.

⌋

[SWS_Dem_01423] Successful Completion of Non-Volatile Memory Initialization and Storage

Upstream requirements: RS_Diag_04211

⌈The function `Dem_InitAllNvmBlocksGetResult` shall return `E_OK` if the function `Dem_InitAllNvmBlocks` was called successfully and all `Dem` related NvM blocks are initialized and stored to non-volatile memory.⌋

[SWS_Dem_01424] Indication of Ongoing Non-Volatile Memory Storage

Upstream requirements: RS_Diag_04211

⌈The function `Dem_InitAllNvmBlocksGetResult` shall return `E_PENDING` if the function `Dem_InitAllNvmBlocks` was called successfully, but storing to non-volatile memory is still ongoing and not yet finished.⌋

[SWS_Dem_01425] Error Indication for Unsuccessful or Invalid Non-Volatile Memory Operations

Upstream requirements: RS_Diag_04211

⌈The function `Dem_InitAllNvmBlocksGetResult` shall return `E_NOT_OK` in all other cases. (e.g. when `Dem_InitAllNvmBlocks` was not called successfully or `Dem` related NvM storage operations are returning an error)⌋

7.11.6 Interaction with Default Error Tracer (`Det`)

The interaction with the `Det` is described in chapter 7.12.

7.11.7 Required data by the Dem module

The Dem module requires different information for internal computation/processing. If this information is provided by SWCs (as provide-ports), they are described in the respective ServiceNeeds (diagnostic capabilities).

One kind of information required by the Dem module are event related data (represented by freeze frames and extended data records, refer to chapter 7.7.7).

Note: For OBDrelevant ECUs, the Dem module could access (via the data element interface) on the following data values:

- engine temperature engine speed
- vehicle speed (refer to [SWS_Dem_00346], PID $0D)
- distance information

- programming event
- ambient temperature
- ambient pressure
- accelerator pedal information

However, the list of data elements and their required size and resolution are implementationspecific and will be configured in OBD and handled during integration process. For example, the vehicle speed, accelerator pedal information, ambient temperature, etc. are necessary to evaluate the cycle conditions of the IUMPR General Denominator. Similar inputs are necessary for the PID computation (e.g. the engine temperature for the computation of the WarmUp cycle or the WarmUp cycle condition itself). These variables are typically accessed through the RTE.

7.11.8 Scaling information on Service Interfaces

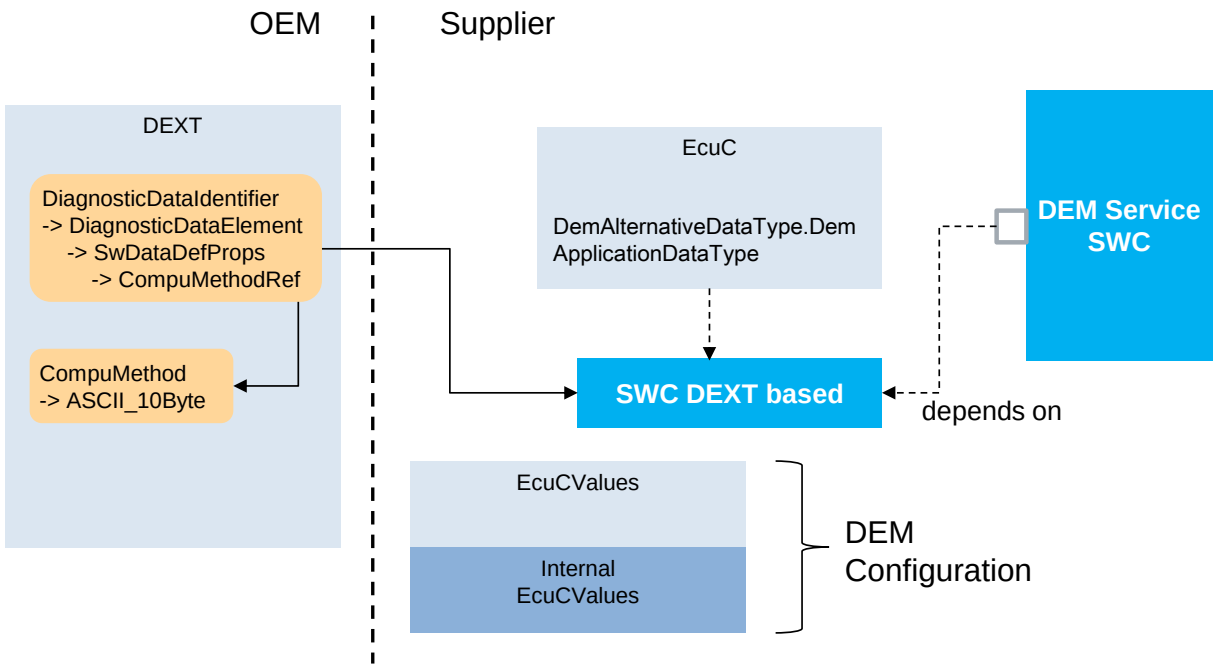
The SoftwareComponentTemplate [22] defines the meta-class SwDataDefProps which covers all properties of a particular data object under various aspects. This chapter describes two alternative workflows how the SwDataDefProps properties such as CompuMethod, DataContraints and Units can be used within the Dem Service SWC and make them there available per `DemExternalSRDataElementClass`.

[SWS_Dem_CONSTR_06117] ⌈The aggregation of `DemTextTableMapping` at `DemAlternativeDataType` is only valid if the category of the CompuMethod of the DataType referenced by `DemApplicationDataType` has category set to TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.⌋

Work flow 1

This work flow is the use of a EcucForeignReference inside the generated EcuC values. While importing the DEXT [23] information the `DemDiagnosticDataElementRef` is derived and holds a EcucForeignReference to a DiagnosticDataElement in the DEXT [23] file. This EcucForeignReference enables the access to all SwDataDefProps (BaseTyoe, CompuMethod, DatConstr, etc.) of the corresponding DiagnosticDataElement. The container `DemAlternativeDiagnosticDataElement` aggregates this EcucForeignReference. In the process step of generating the corresponding Service SWC all needed content will be copied directly based on the EcucForeignReference from DEXT [23] to the Service SWC. In this work flow the

existence of the DEXT file while the generation of the Service SWC is required.

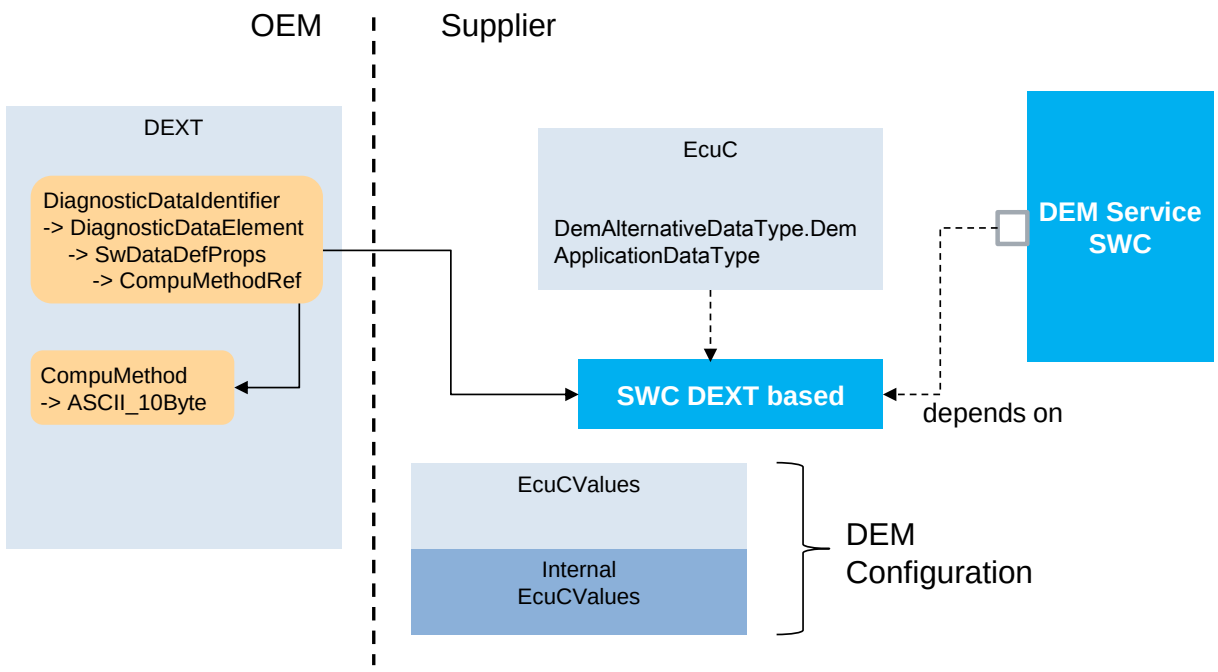


* The DEXT importer generates one SWC which contains all SwDataDefProps and the ECUC parameter (DemAlternativeDataType.DemApplicationDataType) refers to the SWC. Within the SWBuild processing this reference to SWC is taken and the required contents (CompuMethod / DataContraints / Units) are included into the Dem SWCD.

Figure 7.50: Workflow 1

Work flow 2

This work flow is that while importing the DEXT [23] information beside the EcuC values also a SWC fragment is generated. In this SWC fragment all needed SwDataDefProps are directly copied from the DEXT [23] file. Inside the generated EcuC values the EcuC Parameter `DemApplicationDataType` refers to the SWC fragment and enables the access to all SwDataDefProps (BaseTyoe, CompuMethod, DatConstr, etc.). In the process step of generating the corresponding Service SWC all needed content will be included based on the reference from `DemApplicationDataType` to the SWC fragment. In this work flow the existence of the DEXT [23] file while the generation of the Service SWC is not required.



* The DEXT importer generates one SWC which contains all SwDataDefProps and the ECUC parameter (DemAlternativeDataType.DemApplicationDataType) refers to the SWC. Within the SWBuild processing this reference to SWC is taken and the required contents (CompuMethod / DataContraints / Units) are included into the Dem SWCD.

Figure 7.51: Workflow 2

7.12 Error classification

Document [6] Chapter 7.2 *"Error Handling"* describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

Based on this foundation, the following section specifies particular errors arranged in the respective subsections below.

7.12.1 Development Errors

[SWS_Dem_00173] Definition of development errors in module Dem

Upstream requirements: SWS_BSW_00050, SWS_BSW_00212

⌈

Type of error	*Related error code*	*Error value*
API function called with a parameter value, which is not allowed by active configuration	DEM_E_WRONG_CONFIGURATION	0x10
API function called with a NULL pointer	DEM_E_PARAM_POINTER	0x11
API function called with wrong parameter value	DEM_E_PARAM_DATA	0x12
API function called with wrong length parameter value	DEM_E_PARAM_LENGTH	0x13
Dem initialisation failed (refer to [SWS_BSW_00151])	DEM_E_INIT_FAILED	0x14
API function called before the Dem module has been full initialized (refer to [SWS_Dem_00124],[SWS_Dem_00364]) or after the Dem module has been shut down (refer to [SWS_Dem_00368])	DEM_E_UNINIT	0x20
Required conditions for the respective API call are not fulfilled (e.g. an invalid status change was initiated, or a filter was not set correctly, etc. - refer to [SWS_Dem_00518]).	DEM_E_WRONG_CONDITION	0x40
Dem_DcmGetAvailableOBDMIDs called with invalid OBDMID.	DEM_E_INVALID_OBDMID	0x50

⌋

[SWS_Dem_00124]

Upstream requirements: SRS_BSW_00406

⌈If `DemDevErrorDetect` is set to TRUE and any Dem API excluding :

- `Dem_SetEventStatus`
- `Dem_ResetEventStatus`
- `Dem_SetEventAvailable`
- `Dem_ResetEventDebounceStatus`
- `Dem_MainFunction`
- `Dem_InitAllNvmBlocks`

is called before `Dem` has been fully initialized, the `Dem` module shall set the error code `DEM_E_UNINIT`.⌋

Note: If development error detection is disabled and the Dem is not fully initialized, the behavior of the APIs is undefined.

[SWS_Dem_01324] ⌈During `Dem_Init` the `Dem` initializes the monitor and component states. After this point in time, the `Dem` shall return `E_OK` on the APIs `Dem_GetMonitorStatus` and `Dem_GetComponentFailed`.⌋

[SWS_Dem_00364]

Upstream requirements: SRS_BSW_00406

⌈If development error detection is enabled and any instance calls `Dem_SetEventStatus` or `Dem_ResetEventDebounceStatus` before the Dem was preinitialized, the Dem module shall set the error code `DEM_E_UNINIT`.⌋

Note: If development error detection is disabled and the Dem is not preinitialized, the behavior of these APIs is undefined.

[SWS_Dem_00368]

Upstream requirements: SRS_BSW_00337

⌈If development error detection is enabled and any instance calls any Dem API, excluding `Dem_SetEventStatus`, `Dem_ResetEventDebounceStatus` and `Dem_MainFunction`, after `Dem_Shutdown` has been called, the Dem module shall set the error code `DEM_E_UNINIT` until `Dem_Init` is called again.⌋

Note: If development error detection is disabled and the Dem is shut down, the behavior of these APIs is undefined.

[SWS_Dem_00518]

Upstream requirements: RS_Diag_04057

⌈If development error detection is enabled and a Dem function is called with required preconditions not fulfilled, the Dem module shall set the error code `DEM_E_WRONG_CONDITION`.⌋

Note: For example, `Dem_GetNextFilteredDTCAndFDC` is called, after `Dem_SetDTCFilter` with `FilterForFaultDetectionCounter` = FALSE was called.

7.12.2 Runtime Errors

[SWS_Dem_91035] Definition of runtime errors in module Dem ⌈

Type of error	*Related error code*	*Error value*
The UDS status corresponding to a changed monitor status could not be processed.	DEM_E_UDS_STATUS_PROCESSING_FAILED	0x21
No valid data for data element available by SW-C or BSW call	DEM_E_NODATAAVAILABLE	0x30

⌋

7.12.3 Production Errors

There are no production errors.

8 API specification

The figures below show the interfaces between Dem and its surrounding SW-Cs and BSW modules. The description of the interface shall give a simple overview of these relations.

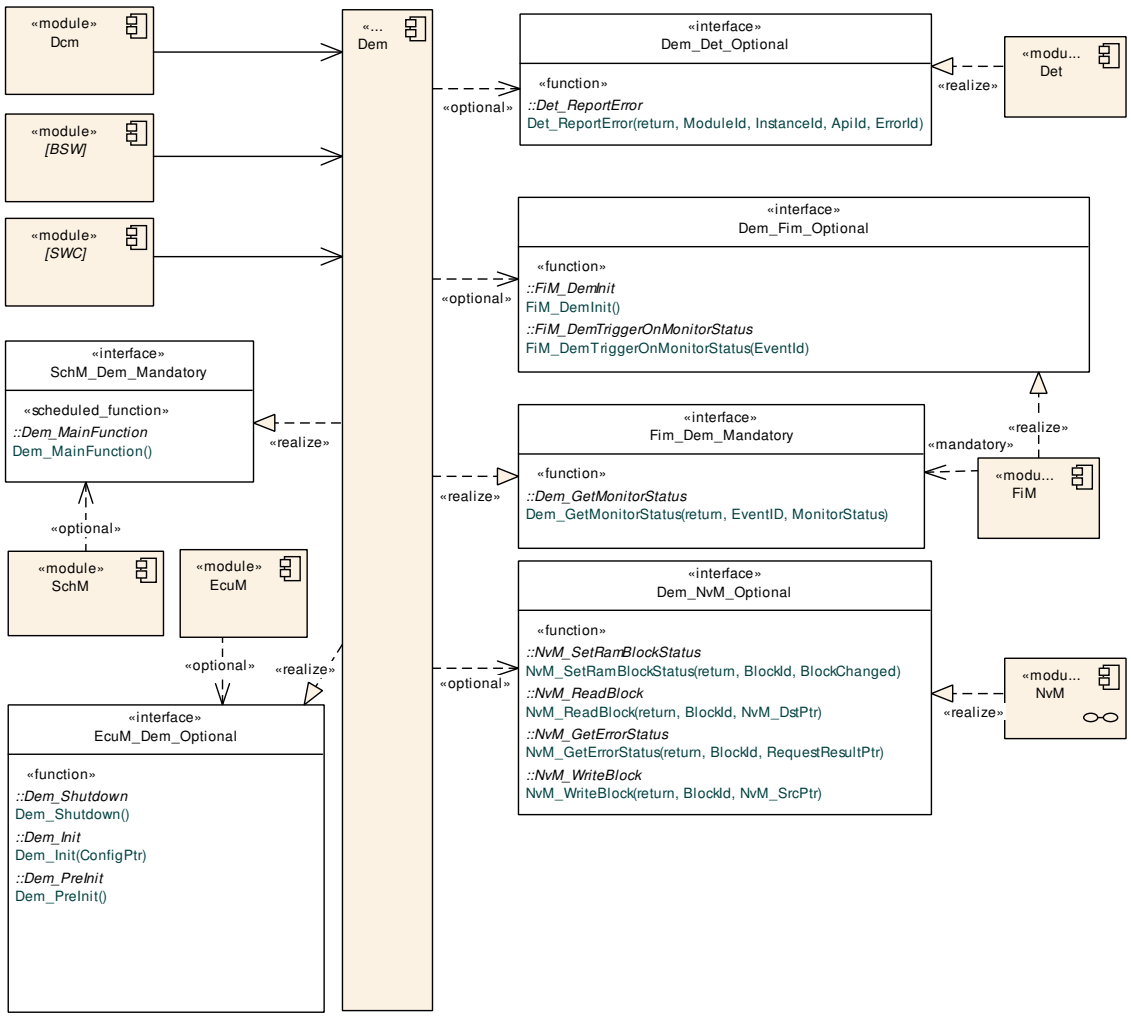


Figure 8.1: Overview of interfaces between the Dem and other BSW modules

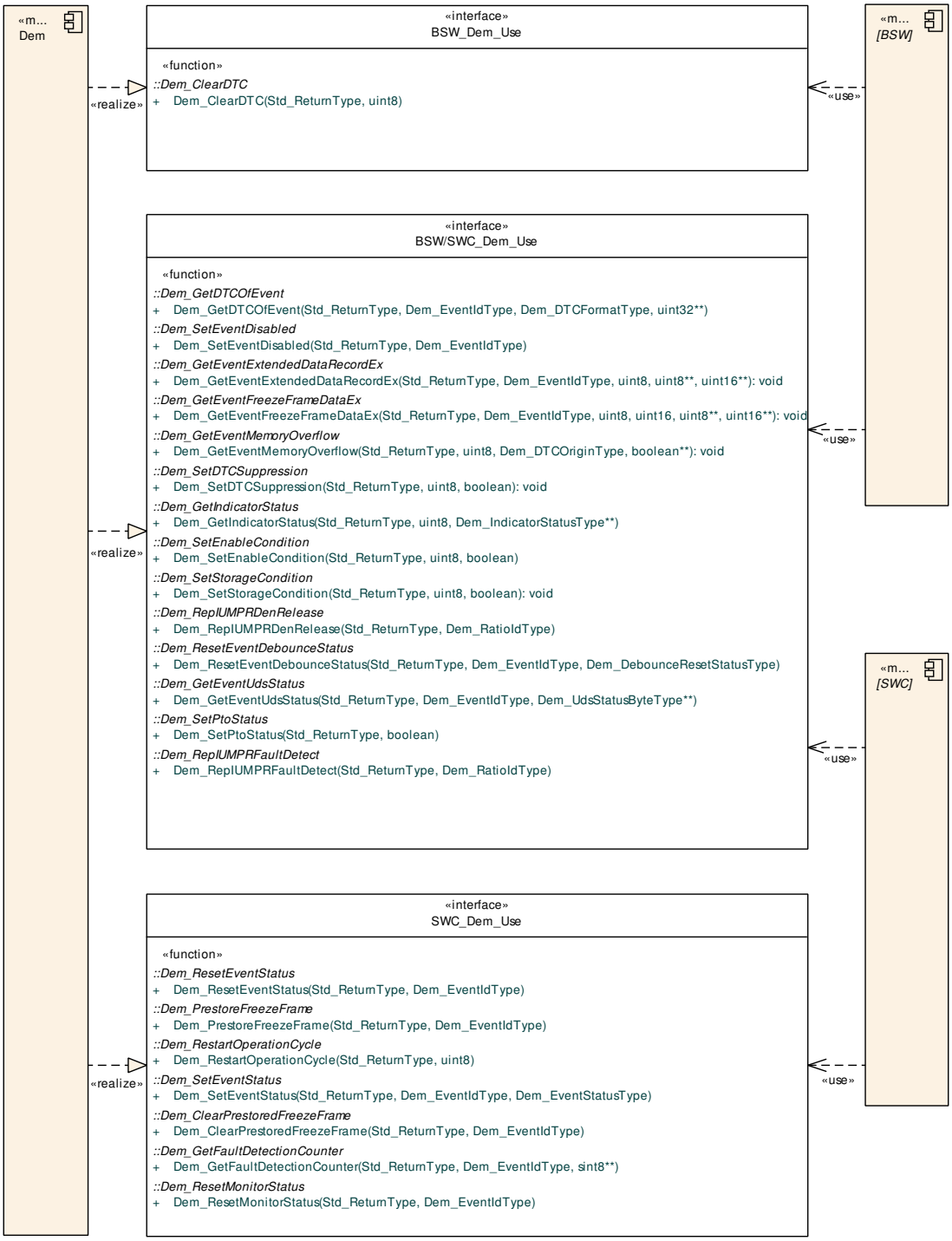


Figure 8.2: Overview of interfaces between the Dem and other BSW modules (in general)

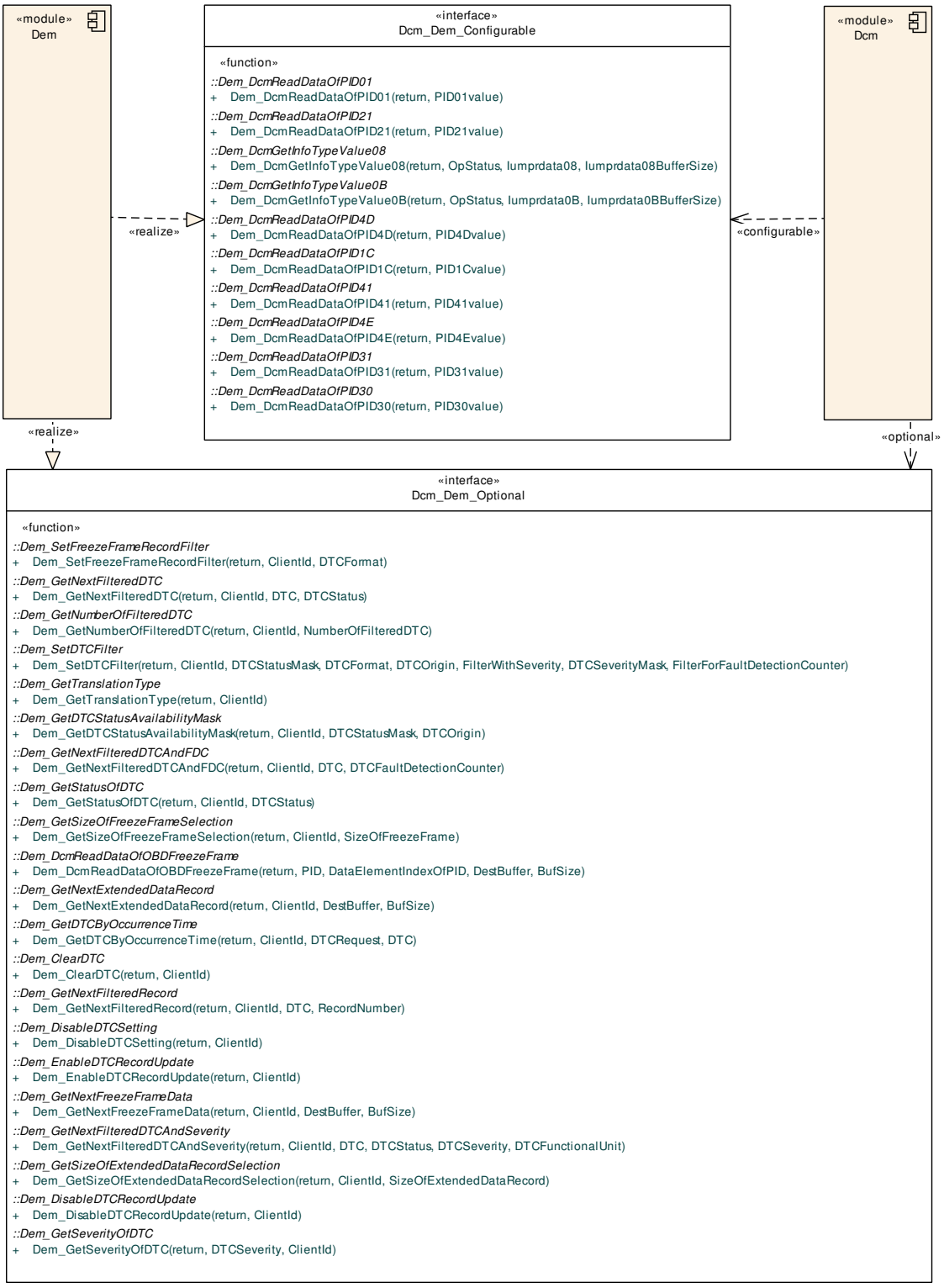


Figure 8.3: Overview of interfaces between the Dem and Dcm

8.1 Imported Types

This section lists all types included from other modules.

[SWS_Dem_00176] Definition of imported datatypes of module Dem

Upstream requirements: SRS_BSW_00301

⌈

Module	*Header File*	*Imported Type*
Dcm	Rte_Dcm_Type.h	Dcm_NegativeResponseCodeType
	Rte_Dcm_Type.h	Dcm_OpStatusType
NvM	Rte_NvM_Type.h	NvM_BlockIdType
	Rte_NvM_Type.h	NvM_RequestResultType
Std	Std_Types.h	Std_ReturnType

⌋

8.2 Type definitions

The following Data Types shall be used for the functions defined in this specification.

8.2.1 Dem data types

8.2.1.1 Dem_ComponentIdType

[SWS_Dem_01114] Definition of datatype Dem_ComponentIdType ⌈

Name	Dem_ComponentIdType		
Kind	Type		
Derived from	uint16		
Range	1..65535	–	Internal identifier of a monitored component. Remark: 0 is not a valid value
Description	Identification of a DemComponent by assigned ComponentId. The ComponentId is automatically assigned by the Dem.		
Available via	Dem.h		

⌋

8.2.1.2 Dem_ConfigType

[SWS_Dem_00924] Definition of datatype Dem_ConfigType ⌈

Name	Dem_ConfigType	
Kind	Structure	
Elements	implementation specific	
	Type	–
	Comment	–

▽

△

Description	This type of the external data structure shall contain the post build initialization data for the Dem.
Available via	Dem.h

⌋

8.2.1.3 Dem_EventIdType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.8.

8.2.1.4 Dem_EventStatusType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.9.

8.2.1.5 Dem_DebouncingStateType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.5.

8.2.1.6 Dem_DebounceResetStatusType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.6.

8.2.1.7 Dem_UdsStatusByteType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.21.

8.2.1.8 Dem_IndicatorStatusType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.17.

8.2.1.9 Dem_MonitorDataType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.15.

8.2.1.10 Dem_MonitorStatusType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.16.

8.2.1.11 Dem_DTCKindType

[SWS_Dem_00932] Definition of datatype Dem_DTCKindType

Upstream requirements: RS_Diag_04129

⌈

Name	Dem_DTCKindType		
Kind	Type		
Derived from	uint8		
Range	DEM_DTC_KIND_ALL_DTCS	0x01	Select all DTCs
	DEM_DTC_KIND_EMISSION_REL_DTCS	0x02	Select OBD-relevant DTCs
	DEM_DTC_KIND_WWHOBD_CLASS_A	0x03	Select all OBD-relevant DTCs with WWH-OBD class A
	DEM_DTC_KIND_WWHOBD_CLASS_B1	0x04	Select all OBD-relevant DTCs with WWH-OBD class B1
	DEM_DTC_KIND_WWHOBD_CLASS_B2	0x05	Select all OBD-relevant DTCs with WWH-OBD class B2
	DEM_DTC_KIND_WWHOBD_CLASS_C	0x06	Select all OBD-relevant DTCs with WWH-OBD class C
Description	This type is used to filter DTCs for their kind.		
Available via	Dem_J1939Dcm.h		

⌋

8.2.1.12 Dem_DTCFormatType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.10.

8.2.1.13 Dem_DTCOriginType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.4.

8.2.1.14 Dem_DTCRequestType

[SWS_Dem_00935] Definition of datatype Dem_DTCRequestType

Upstream requirements: RS_Diag_04195

⌈

Name	Dem_DTCRequestType		
Kind	Type		
Derived from	uint8		
Range	DEM_FIRST_FAILED_DTC	0x01	first failed DTC requested
	DEM_MOST_RECENT_FAILED_DTC	0x02	most recent failed DTC requested
	DEM_FIRST_DET_CONFIRMED_DTC	0x03	first detected confirmed DTC requested
	DEM_MOST_REC_DET_CONFIRMED_DTC	0x04	most recently detected confirmed DTC requested
Description	This type is used to request a DTC with specific attributes.		
Available via	Dem.h		

⌋

8.2.1.15 Dem_DTCTranslationFormatType

[SWS_Dem_00936] Definition of datatype Dem_DTCTranslationFormatType

Upstream requirements: RS_Diag_04067

⌈

Name	Dem_DTCTranslationFormatType		
Kind	Type		
Derived from	uint8		
Range	DEM_DTC_TRANSLATION_ISO15031_6	0x00	ISO15031-6 DTC format/SAE J2012-DA_DTCFormat_00 DTC format
	DEM_DTC_TRANSLATION_ISO14229_1	0x01	ISO14229-1 DTC format
	DEM_DTC_TRANSLATION_SAEJ1939_73	0x02	SAEJ1939-73 DTC format
	DEM_DTC_TRANSLATION_ISO11992_4	0x03	ISO11992-4 DTC format
	DEM_DTC_TRANSLATION_J2012DA_FORMAT_04	0x04	SAE_J2012-DA_DTCFormat_04 DTC format
Description	DTC translation format as defined in ISO14229-1.		
Available via	Dem.h		

⌋

8.2.1.16 Dem_DTCSeverityType

[SWS_Dem_00937] Definition of datatype Dem_DTCSeverityType ⌈

Name	Dem_DTCSeverityType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	DEM_SEVERITY_NO_ SEVERITY	0x00	No severity information available
	bit	DEM_SEVERITY_WWHOBD_ CLASS_NO_CLASS	0x01	No class information
	bit	DEM_SEVERITY_WWHOBD_ CLASS_A	0x02	WWH-OBD Class A
	bit	DEM_SEVERITY_WWHOBD_ CLASS_B1	0x04	WWH-OBD Class B1
	bit	DEM_SEVERITY_WWHOBD_ CLASS_B2	0x08	WWH-OBD Class B2
	bit	DEM_SEVERITY_WWHOBD_ CLASS_C	0x10	WWH-OBD Class C
	bit	DEM_SEVERITY_ MAINTENANCE_ONLY	0x20	maintenance required
	bit	DEM_SEVERITY_CHECK_AT_ NEXT_HALT	0x40	check at next halt
	bit	DEM_SEVERITY_CHECK_ IMMEDIATELY	0x80	Check immediately
Description	Type definition of DTCSeverityMask / DTCSeverity byte containing the DTC severity and DTC class information according to ISO 14229-1 Annex D.3. The upper 3 bits (bit 7-5) are used to represent the DTC severity information. The lower 5 bits (bit 4-0) are used to represent the DTC class information.			
Available via	Dem.h			

⌋

8.2.1.17 Dem_RatioIdType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.20.

8.2.1.18 Dem_DTRControlType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.7.

8.2.1.19 Dem_InitMonitorReasonType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.11.

8.2.1.20 Dem_IumprDenomCondIdType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.12.

8.2.1.21 Dem_IumprDenomCondStatusType

This data type is used in both, C-APIs and service interfaces. It's definition is available in chapter 8.6.1.13.

8.2.1.22 Dem_J1939DcmDTCStatusFilterType

[SWS_Dem_00945] Definition of datatype Dem_J1939DcmDTCStatusFilterType

Upstream requirements: RS_Diag_04112

⌈

Name	Dem_J1939DcmDTCStatusFilterType		
Kind	Type		
Derived from	uint8		
Range	DEM_J1939DTC_ACTIVE	0	active DTCs
	DEM_J1939DTC_ PREVIOUSLY_ACTIVE	1	previously active DTCs
	DEM_J1939DTC_ PENDING	2	pending DTCs
	DEM_J1939DTC_ PERMANENT	3	permanent DTCs
	DEM_J1939DTC_ CURRENTLY_ACTIVE	4	currently active DTC
Description	The type to distinguish which DTCs should be filtered.		
Available via	Dem.h		

⌋

8.2.1.23 Dem_J1939DcmDTCSeverityType

[SWS_Dem_91204] Definition of datatype Dem_J1939DcmDTCSeverityType ⌈

Name	Dem_J1939DcmDTCSeverityType			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	DEM_SEVERITY_NO_ SEVERITY	0x00	No severity information available
	bit	DEM_SEVERITY_CHECK_ IMMEDIATELY	0x80	Select all DTCs with severity "CheckImmediately"

▽

△

	bit	DEM_SEVERITY_CHECK_AT_NEXT_HALT	0x40	Select all DTCs with severity "CheckAtNextHalt"
	bit	DEM_SEVERITY_MAINTENANCE_ONLY	0x20	Select all DTCs with severity "MaintenanceOnly"
Description	Type definition of a DTC severity byte containing the DTC severity information for a J1939 DTC. The upper three bits (bits 7-5) represent the DTC severity information; no DTC class information is represented.			
Available via	Dem_J1939Dcm.h			

⌋

8.2.1.24 Dem_J1939DcmSetClearFilterType

[SWS_Dem_00946] Definition of datatype Dem_J1939DcmSetClearFilterType

Upstream requirements: RS_Diag_04112

⌈

Name	Dem_J1939DcmSetClearFilterType		
Kind	Type		
Derived from	uint8		
Range	DEM_J1939DTC_CLEAR_ACTIVE	0	active DTCs
	DEM_J1939DTC_CLEAR_PREVIOUSLY_ACTIVE	1	previously active DTCs
	DEM_J1939DTC_CLEAR_ACTIVE_AND_PREVIOUSLY_ACTIVE	2	active and previously active DTCs
Description	The type to distinguish which DTCs gets cleared		
Available via	Dem.h		

⌋

8.2.1.25 Dem_J1939DcmSetFreezeFrameFilterType

[SWS_Dem_00947] Definition of datatype Dem_J1939DcmSetFreezeFrameFilterType ⌈

Name	Dem_J1939DcmSetFreezeFrameFilterType		
Kind	Type		
Derived from	uint8		
Range	DEM_J1939DCM_FREEZEFRAME	0	FreezeFrame (DM04)
	DEM_J1939DCM_EXPANDED_FREEZEFRAME	1	ExpandedFreezeFrame (DM25)

▽

△

	DEM_J1939DCM_SPNS_IN_EXPANDED_FREEZEFRAME	2	SPNs in ExpandedFreezeFrame (DM24)
Description	The type to distinguish which DTCs gets cleared		
Available via	Dem.h		

⌋

8.2.1.26 Dem_J1939DcmLampStatusType

[SWS_Dem_00948] Definition of datatype Dem_J1939DcmLampStatusType

Upstream requirements: RS_Diag_04110

⌈

Name	Dem_J1939DcmLampStatusType	
Kind	Structure	
Elements	LampStatus	
	Type	uint8
	Comment	lamp status
	FlashLampStatus	
	Type	uint8
	Comment	flash lamp status
Description	For details refer SAE J1939-73	
Available via	Dem.h	

⌋

8.2.1.27 Dem_J1939DcmDiagnosticReadiness1Type

[SWS_Dem_00949] Definition of datatype Dem_J1939DcmDiagnosticReadiness1Type

Upstream requirements: RS_Diag_04113

⌈

Name	Dem_J1939DcmDiagnosticReadiness1Type	
Kind	Structure	
Elements	ActiveTroubleCodes	
	Type	uint8
	Comment	Number of active DTCs
	PreviouslyActiveDiagnosticTroubleCodes	
	Type	uint8

▽

△

	Comment	Number of previously active DTCs
	OBDCompliance	
	Type	uint8
	Comment	OBD Compliance
	ContinuouslyMonitoredSystemsSupport_Status	
	Type	uint8
	Comment	Identifies the continuously monitored system support and status
	NonContinuouslyMonitoredSystemsSupport5	
	Type	uint8
	Comment	Identifies the non-continuously monitored systems support (byte5)
	NonContinuouslyMonitoredSystemsSupport6	
	Type	uint8
	Comment	Identifies the non-continuously monitored systems support (byte6)
	NonContinuouslyMonitoredSystemsStatus7	
	Type	uint8
	Comment	Identifies the non-continuously monitored systems status (byte7)
	NonContinuouslyMonitoredSystemsStatus8	
	Type	uint8
	Comment	Identifies the non-continuously monitored systems status (byte8)
Description	This structure represents all data elemets of the DM05 message. The encoding shall be done acording SAE J1939-73	
Available via	Dem.h	

⌋

8.2.1.28 Dem_J1939DcmDiagnosticReadiness2Type

[SWS_Dem_00950] Definition of datatype Dem_J1939DcmDiagnosticReadiness2Type

Upstream requirements: RS_Diag_04113

⌈

Name	Dem_J1939DcmDiagnosticReadiness2Type	
Kind	Structure	
Elements	DistanceTraveledWhileMILisActivated	
	Type	uint16
	Comment	The kilometers accumulated while the MIL is activated
	DistanceSinceDTCsCleared	
	Type	uint16
	Comment	Distance accumulated since emission related DTCs were cleared
	MinutesRunbyEngineWhileMILisActivated	
	Type	uint16
	Comment	Accumulated count (in minutes) while the MIL is activated (on)

▽

△

	TimeSinceDiagnosticTroubleCodesCleared	
	Type	uint16
	Comment	Engine running time accumulated since emission related DTCs were cleared
Description	This structure represents all data elemets of the DM21 message. The encoding shall be done acording SAE J1939-73	
Available via	Dem.h	

⌋

8.2.1.29 Dem_J1939DcmDiagnosticReadiness3Type

[SWS_Dem_00951] Definition of datatype Dem_J1939DcmDiagnosticReadiness3Type

Upstream requirements: RS_Diag_04113

⌈

Name	Dem_J1939DcmDiagnosticReadiness3Type	
Kind	Structure	
Elements	TimeSinceEngineStart	
	Type	uint16
	Comment	Time since key-on that the engine has been running.
	NumberofWarmupsSinceDTCsCleared	
	Type	uint8
	Comment	Number of OBD warm-up cycles since all DTCs were cleared
	ContinuouslyMonitoredSystemsEnableCompletedStatus	
	Type	uint8
	Comment	Identifies the continuously monitored system enable/completed support and status.
	NonContinuouslyMonitoredSystemsEnableStatus5	
	Type	uint8
	Comment	Enable status of non-continuous monitors this monitoring cycle (byte5)
	NonContinuouslyMonitoredSystemsEnableStatus6	
	Type	uint8
	Comment	Enable status of non-continuous monitors this monitoring cycle (byte6)
	NonContinuouslyMonitoredSystems7	
	Type	uint8
	Comment	Completion status of non-continuous monitors this monitoring cycle (byte7)
	NonContinuouslyMonitoredSystems8	
	Type	uint8
	Comment	Completion status of non-continuous monitors this monitoring cycle (byte8)

▽

△

Description	This structure represents all data elemets of the DM26 message. The encoding shall be done acording SAE J1939-73
Available via	Dem.h

⌋

8.2.1.30 Dem_EventOBDReadinessGroupType

[SWS_Dem_91194] Definition of datatype Dem_EventOBDReadinessGroupType

⌈

Name	Dem_EventOBDReadinessGroupType		
Kind	Type		
Derived from	uint8		
Range	0..255	–	–
Description	This Data type corresponds to the readiness group defined by SAE J1979-DA		
Available via	Dem.h		

⌋

8.2.2 Extension to Std ReturnType

[SWS_Dem_91201] Definition of Std_ReturnType-extension for module Dem

Upstream requirements: SRS_BSW_00377

⌈

Range	DEM_PENDING	0x04	Operation is currently in progress
	DEM_CLEAR_BUSY	0x05	Another client is currently clearing DTCs
	DEM_CLEAR_MEMORY_ ERROR	0x06	An error occurred during erasing a memory location
	DEM_CLEAR_FAILED	0x07	DTC clearing failed
	DEM_WRONG_DTC	0x08	Selected DTC does not exist
	DEM_WRONG_ DTCORIGIN	0x09	Selected DTCOrigin does not exist
	DEM_E_NO_DTC_ AVAILABLE	0x0A	No DTC configured in the requested format
	DEM_WRONG_ DTCTYPEFILTER	0x0B	The state of the requested J1939 DTC does not match the given DTCTypeFilter.
	DEM_E_NO_FDC_ AVAILABLE	0x0E	No fault detection counter available for the requested event
	DEM_BUFFER_TOO_ SMALL	0x15	The provided buffer size is too small
	DEM_BUSY	0x16	A different operation is currently in progress

▽

△

	DEM_NO_SUCH_ELEMENT	0x30	The requested element is not available
Description	Overlayed return values of Std_ReturnType for DEM APIs.		
Available via	Dem.h		

⌋

8.3 Function definitions

This chapter contains all APIs provided by the module.

8.3.1 Interface ECU State Manager <=> Dem

8.3.1.1 Dem_PreInit

[SWS_Dem_00179] Definition of API function Dem_PreInit ⌈

Service Name	Dem_PreInit

Syntax	`void Dem_PreInit (`
	`  void`
	`)`

Service ID [hex]	0x01
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	Initializes the internal states necessary to process events reported by BSW-modules.
Available via	Dem.h

⌋

8.3.1.2 Dem_Init

[SWS_Dem_00181] Definition of API function Dem_Init

Upstream requirements: SRS_BSW_00101

⌈

Service Name	Dem_Init

Syntax	`void Dem_Init (`
	`  const Dem_ConfigType* ConfigPtr`
	`)`

▽

△

Service ID [hex]	0x02	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ConfigPtr	Pointer to the configuration set in VARIANT-POST-BUILD.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Initializes or reinitializes this module.	
Available via	Dem.h	

⌋

8.3.1.3 Dem_Shutdown

[SWS_Dem_00182] Definition of API function Dem_Shutdown

Upstream requirements: SRS_BSW_00336

⌈

Service Name	Dem_Shutdown

Syntax	`void Dem_Shutdown (`
	`  void`
	`)`

Service ID [hex]	0x03
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	Shuts down this module.
Available via	Dem.h

⌋

8.3.2 Interface BSW modules / SW-Components <=> Dem

8.3.2.1 Dem_ClearDTC

[SWS_Dem_00665] Definition of API function Dem_ClearDTC ⌈

Service Name	Dem_ClearDTC

Syntax	`Std_ReturnType Dem_ClearDTC (`
	`  uint8 ClientId`
	`)`

▽

△

Service ID [hex]	0x23	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	None	
Return value	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 restricted by configuration to group of all DTCs only. DEM_WRONG_DTCORIGIN: Selected DTCOrigin does not exist 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_NONVOLATILE_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. DEM_BUSY: A different Dem_SelectDTC dependent operation according to [SWS_Dem_01253] of this client is currently in progress.
Description	Clears single DTCs, as well as groups of DTCs.	
Available via	Dem.h	

⌋

8.3.2.2 Dem_ClearPrestoredFreezeFrame

[SWS_Dem_00193] Definition of API function Dem_ClearPrestoredFreezeFrame ⌈

Service Name	Dem_ClearPrestoredFreezeFrame	
Syntax	`Std_ReturnType Dem_ClearPrestoredFreezeFrame ( Dem_EventIdType EventId )`	
Service ID [hex]	0x07	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Clear prestored freeze frame was successful E_NOT_OK: Clear prestored freeze frame failed
Description	Clears a prestored freeze frame of a specific event. This API can only be used through the RTE and therefore no declaration is exported via Dem.h. API Availability: This API will be available only if ({ecuc(Dem/DemConfigSet/DemEventParameter.DemFFPrestorageSupported)} == true)	

▽

△

Available via	Dem.h

⌋

8.3.2.3 Dem_GetComponentFailed

[SWS_Dem_01115] Definition of API function Dem_GetComponentFailed ⌈

Service Name	Dem_GetComponentFailed	

Syntax	`Std_ReturnType Dem_GetComponentFailed (`
	`  Dem_ComponentIdType ComponentId,`
	`  boolean* ComponentFailed`
	`)`	

Service ID [hex]	0x2a	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	ComponentId	Identification of a DemComponent
Parameters (inout)	None	
Parameters (out)	ComponentFailed	TRUE: failed FALSE: not failed
Return value	Std_ReturnType	`E_OK`: getting "ComponentFailed" was successful `E_NOT_OK`: getting "ComponentFailed" was not successful
Description	Gets the failed status of a DemComponent.	
Available via	Dem.h	

⌋

8.3.2.4 Dem_GetDTCSelectionResult

[SWS_Dem_91023] Definition of API function Dem_GetDTCSelectionResult ⌈

Service Name	Dem_GetDTCSelectionResult	

Syntax	`Std_ReturnType Dem_GetDTCSelectionResult (`
	`  uint8 ClientId`
	`)`	

Service ID [hex]	0xb8	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	None	

▽

△

Return value	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 DEM_PENDING: Checking the SelectDTC parameters is currently in progress. The caller shall call this function again later. DEM_BUSY: A different Dem_SelectDTC dependent operation according to [SWS_Dem_01253] of this client is currently in progress.
Description	Provides information if the last call to Dem_SelectDTC has selected a valid DTC or group of DTCs.	
Available via	Dem.h	

⌋

8.3.2.5 Dem_GetDTCSelectionResultForClearDTC

[SWS_Dem_91020] Definition of API function Dem_GetDTCSelectionResultForClearDTC ⌈

Service Name	Dem_GetDTCSelectionResultForClearDTC	
Syntax	`Std_ReturnType Dem_GetDTCSelectionResultForClearDTC ( uint8 ClientId )`	
Service ID [hex]	0xbb	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	None	
Return value	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. DEM_WRONG_DTC: Selected DTC value in selected format does not exist or a single DTC was selected and Dem only supports to Clear all DTCs. DEM_WRONG_DTCORIGIN: Selected DTCOrigin does not exist. DEM_PENDING: Checking the SelectDTC parameters is currently in progress. The caller shall call this function again later. DEM_BUSY: A different Dem_SelectDTC dependent operation according to [SWS_Dem_01253] of this client is currently in progress. E_NOT_OK: No DTC selected.
Description	Provides information if the last call to Dem_SelectDTC has selected a valid DTC or group of DTCs, respecting the settings if the Dem shall clear only all DTCs.	
Available via	Dem.h	

⌋

8.3.2.6 Dem_GetEventUdsStatus

[SWS_Dem_91008] Definition of API function Dem_GetEventUdsStatus

Upstream requirements: RS_Diag_04067

⌈

Service Name	Dem_GetEventUdsStatus	

Syntax	`Std_ReturnType Dem_GetEventUdsStatus (`
	`  Dem_EventIdType EventId,`
	`  Dem_UdsStatusByteType* UDSStatusByte`
	`)`	

Service ID [hex]	0xb6	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	UDSStatusByte	UDS DTC status byte of the requested event (refer to chapter "Status bit support"). If the return value of the function call is E_NOT_OK, this parameter does not contain valid data.
Return value	Std_ReturnType	`E_OK`: get of event status was successful `E_NOT_OK`: get of event status failed
Description	Gets the current UDS status byte assigned to the DTC for the event	
Available via	Dem.h	

⌋

8.3.2.7 Dem_GetMonitorStatus

[SWS_Dem_91007] Definition of API function Dem_GetMonitorStatus ⌈

Service Name	Dem_GetMonitorStatus	

Syntax	`Std_ReturnType Dem_GetMonitorStatus (`
	`  Dem_EventIdType EventID,`
	`  Dem_MonitorStatusType* MonitorStatus`
	`)`	

Service ID [hex]	0xb5	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventID	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	MonitorStatus	Monitor status byte of the requested event. If the return value of the function call is E_NOT_OK, this parameter does not contain valid data.
Return value	Std_ReturnType	`E_OK`: Get monitor status was successful, `E_NOT_OK`: getting the monitor status failed.
Description	Gets the current monitor status for an event.	
Available via	Dem.h	

⌋

8.3.2.8 Dem_GetDebouncingOfEvent

[SWS_Dem_00730] Definition of API function Dem_GetDebouncingOfEvent

Upstream requirements: RS_Diag_04068

⌈

Service Name	Dem_GetDebouncingOfEvent	
Syntax	`Std_ReturnType Dem_GetDebouncingOfEvent ( Dem_EventIdType EventId, Dem_DebouncingStateType* DebouncingState )`	
Service ID [hex]	0x9f	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
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	Std_ReturnType	`E_OK`: get of debouncing status per event state successful `E_NOT_OK`: get of debouncing per event state failed
Description	Gets the debouncing status of an event. This function shall not be used for EventId with native debouncing within their functions. It is rather for EventIds using debouncing within the Dem.	
Available via	Dem.h	

⌋

8.3.2.9 Dem_GetDTCOfEvent

[SWS_Dem_00198] Definition of API function Dem_GetDTCOfEvent

Upstream requirements: RS_Diag_04201

⌈

Service Name	Dem_GetDTCOfEvent	
Syntax	`Std_ReturnType Dem_GetDTCOfEvent ( Dem_EventIdType EventId, Dem_DTCFormatType DTCFormat, uint32* DTCOfEvent )`	
Service ID [hex]	0x0d	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	DTCFormat	Defines the output-format of the requested DTC value.
Parameters (inout)	None	

▽

△

Parameters (out)	DTCOfEvent	Receives the DTC value in respective 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	Std_ReturnType	E_OK: get of DTC was successful E_NOT_OK: the call was not successful DEM_E_NO_DTC_AVAILABLE: there is no DTC configured in the requested format
Description	Gets the DTC of an event.	
Available via	Dem.h	

⌋

8.3.2.10 Dem_GetDTCSuppression

[SWS_Dem_91025] Definition of API function Dem_GetDTCSuppression

Upstream requirements: RS_Diag_04126

⌈

Service Name	Dem_GetDTCSuppression	
Syntax	`Std_ReturnType Dem_GetDTCSuppression ( uint8 ClientId, boolean* SuppressionStatus )`	
Service ID [hex]	0xbc	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module
Parameters (inout)	None	
Parameters (out)	SuppressionStatus	Defines whether the respective DTC is suppressed (TRUE) or enabled (FALSE).
Return value	Std_ReturnType	E_OK: Operation was successful. E_NOT_OK: Dem_SelectDTC was not called. DEM_WRONG_DTC: No valid DTC or DTC group selected. DEM_WRONG_DTCORIGIN: Wrong DTC origin selected. DEM_PENDING: The requested value is calculated asynchronously and currently not available. The caller can retry later. DEM_BUSY: A different Dem_SelectDTC dependent operation according to [SWS_Dem_01253] of this client is currently in progress.
Description	Returns the suppression status of a specific DTC. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemSuppressionSupport)} == DEM_DTC_SUPPRESSION)	
Available via	Dem.h	

⌋

8.3.2.11 Dem_GetEventAvailable

[SWS_Dem_91040] Definition of API function Dem_GetEventAvailable ⌈

Service Name	Dem_GetEventAvailable	
Syntax	`Std_ReturnType Dem_GetEventAvailable ( Dem_EventIdType EventId, boolean* AvailableStatus )`	
Service ID [hex]	0xbe	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	AvailableStatus	TRUE if the event is available. FALSE if the event is not available.
Return value	Std_ReturnType	`E_OK` : Event availability has been obtained. `E_NOT_OK` : Event availability cannot be obtained.
Description	Get the Event's availability.	
Available via	Dem.h	

⌋

8.3.2.12 Dem_GetFaultDetectionCounter

[SWS_Dem_00203] Definition of API function Dem_GetFaultDetectionCounter ⌈

Service Name	Dem_GetFaultDetectionCounter	
Syntax	`Std_ReturnType Dem_GetFaultDetectionCounter ( Dem_EventIdType EventId, sint8* FaultDetectionCounter )`	
Service ID [hex]	0x3e	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
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 E_OK this parameter does not contain valid data. -128dec...127dec PASSED...FAILED according to ISO 14229-1
Return value	Std_ReturnType	`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	Gets the fault detection counter of an event. This API can only be used through the RTE, and therefore no declaration is exported via Dem.h.	
Available via	Dem.h	

⌋

8.3.2.13 Dem_GetIndicatorStatus

[SWS_Dem_00205] Definition of API function Dem_GetIndicatorStatus ⌈

Service Name	Dem_GetIndicatorStatus	
Syntax	`Std_ReturnType Dem_GetIndicatorStatus ( uint8 IndicatorId, Dem_IndicatorStatusType* IndicatorStatus )`	
Service ID [hex]	0x29	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	IndicatorId	Number of indicator
Parameters (inout)	None	
Parameters (out)	IndicatorStatus	Status of the indicator, like off, on, or blinking.
Return value	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation failed
Description	Gets the indicator status derived from the UDS status. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral/DemEventMemorySet/DemIndicator)} != NULL)	
Available via	Dem.h	

⌋

8.3.2.14 Dem_GetEventFreezeFrameDataEx

[SWS_Dem_01191] Definition of API function Dem_GetEventFreezeFrameDataEx ⌈

Service Name	Dem_GetEventFreezeFrameDataEx	
Syntax	`Std_ReturnType Dem_GetEventFreezeFrameDataEx ( Dem_EventIdType EventId, uint8 RecordNumber, uint16 DataId, uint8* DestBuffer, uint16* BufSize )`	
Service ID [hex]	0x6e	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	RecordNumber	This parameter is a unique identifier for a freeze frame record as defined in ISO14229-1. **0xFF means most recent freeze frame record is returned.** 0x00 is only supported if the Dem module supports WWH-OBD (refer to DemOBDSupport)
	DataId	This parameter specifies the DID (ISO14229-1) that shall be copied to the destination buffer.
Parameters (inout)	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 record shall be written to. The format is raw hexadecimal values and contains no header-information.
Return value	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation could not be performed DEM_NO_SUCH_ELEMENT: The requested event data is not currently stored (but the request was valid) OR The requested record number is not supported by the event OR The requested DID is not supported by the freeze frame. DEM_BUFFER_TOO_SMALL: The provided buffer size is too small.
Description	Gets the data of a freeze frame by event.	
Available via	Dem.h	

⌋

8.3.2.15 Dem_GetEventExtendedDataRecordEx

[SWS_Dem_01190] Definition of API function Dem_GetEventExtendedDataRecordEx

Upstream requirements: RS_Diag_04205

⌈

Service Name	Dem_GetEventExtendedDataRecordEx	

Syntax	`Std_ReturnType Dem_GetEventExtendedDataRecordEx (`
	`  Dem_EventIdType EventId,`
	`  uint8 RecordNumber,`
	`  uint8* DestBuffer,`
	`  uint16* BufSize`
	`)`	

Service ID [hex]	0x6d	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	RecordNumber	Identification of requested Extended data record. Valid values are between 0x01 and 0xEF as defined in ISO14229-1.
Parameters (inout)	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 shall be written to. The format is raw hexadecimal values and contains no header-information.
Return value	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation could not be performed DEM_NO_SUCH_ELEMENT: The requested event data is not currently stored (but the request was valid) OR the requested record number is not supported by the event. DEM_BUFFER_TOO_SMALL: The provided buffer size is too small.
Description	Gets the data of an extended data record by event.	
Available via	Dem.h	

⌋

8.3.2.16 Dem_GetEventMemoryOverflow

[SWS_Dem_00559] Definition of API function Dem_GetEventMemoryOverflow

Upstream requirements: RS_Diag_04093

⌈

Service Name	Dem_GetEventMemoryOverflow	
Syntax	`Std_ReturnType Dem_GetEventMemoryOverflow ( uint8 ClientId, Dem_DTCOriginType DTCOrigin, boolean* OverflowIndication )`	
Service ID [hex]	0x32	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	DemClientID identifying the DemEventMemorySet indicating in which event memory the overflow has occurred.
	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 (inout)	None	
Parameters (out)	OverflowIndication	This parameter returns TRUE if the according event memory was overflowed, otherwise it returns FALSE.
Return value	Std_ReturnType	`E_OK`: Operation was successful `E_NOT_OK`: Operation failed or is not supported
Description	Gets the event memory overflow indication status.	
Available via	Dem.h	

⌋

8.3.2.17 Dem_GetNumberOfEventMemoryEntries

[SWS_Dem_00652] Definition of API function Dem_GetNumberOfEventMemory Entries

Upstream requirements: RS_Diag_04109

⌈

Service Name	Dem_GetNumberOfEventMemoryEntries	
Syntax	`Std_ReturnType Dem_GetNumberOfEventMemoryEntries ( uint8 ClientId, Dem_DTCOriginType DTCOrigin, uint8* NumberOfEventMemoryEntries )`	
Service ID [hex]	0x35	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	DemClientID identifying the DemEventMemorySet to which the requested event memory belongs to.

▽

△

	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 (inout)	None	
Parameters (out)	NumberOfEventMemoryEntries	Number of entries currently stored in the requested event memory.
Return value	Std_ReturnType	`E_OK`: Operation was successful `E_NOT_OK`: Operation failed
Description	Returns the number of entries currently stored in the requested event memory.	
Available via	Dem.h	

⌋

8.3.2.18 Dem_ResetEventDebounceStatus

[SWS_Dem_00683] Definition of API function Dem_ResetEventDebounceStatus

Upstream requirements: RS_Diag_04105

⌈

Service Name	Dem_ResetEventDebounceStatus	
Syntax	`Std_ReturnType Dem_ResetEventDebounceStatus ( Dem_EventIdType EventId, Dem_DebounceResetStatusType DebounceResetStatus )`	
Service ID [hex]	0x09	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	DebounceResetStatus	Freeze or reset the internal debounce counter/timer of the specified event.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Operation was successful `E_NOT_OK`: Only on development error
Description	Control the internal debounce counter/timer by BSW modules and SW-Cs. The event qualification will not be affected by these debounce state changes. This API is available for BSW modules as soon as Dem_PreInit has been completed (refer to [SWS_Dem_00438] and [SWS_Dem_00167]).	
Available via	Dem.h	

⌋

8.3.2.19 Dem_ResetEventStatus

[SWS_Dem_00185] Definition of API function Dem_ResetEventStatus ⌈

Service Name	Dem_ResetEventStatus	
Syntax	`Std_ReturnType Dem_ResetEventStatus ( Dem_EventIdType EventId )`	
Service ID [hex]	0x05	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Request to reset the event status was successful accepted. `E_NOT_OK`: Request to reset the event status failed or is not allowed, because the event is already tested in this operation cycle.
Description	Resets the event failed status. This API can only be used through the RTE and therefore no declaration is exported via Dem.h.	
Available via	Dem.h	

⌋

8.3.2.20 Dem_ResetMonitorStatus

[SWS_Dem_91196] Definition of API function Dem_ResetMonitorStatus

Status: DRAFT

⌈

Service Name	Dem_ResetMonitorStatus (draft)	
Syntax	`Std_ReturnType Dem_ResetMonitorStatus ( Dem_EventIdType EventId )`	
Service ID [hex]	0xc5	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Request to reset the monitor status was successful and accepted. `E_NOT_OK`: Request to reset the monitor status failed or is not allowed.
Description	Resets the monitor failed status. **Tags:** atp.Status=draft	
Available via	Dem.h	

⌋

8.3.2.21 Dem_RestartOperationCycle

[SWS_Dem_00194] Definition of API function Dem_RestartOperationCycle ⌈

Service Name	Dem_RestartOperationCycle	

Syntax	`Std_ReturnType Dem_RestartOperationCycle (`
	`  uint8 OperationCycleId`
	`)`	

Service ID [hex]	0x08	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	OperationCycleId	Identification of operation cycle, like power cycle, driving cycle.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: set of operation cycle was accepted and will be handled asynchronously `E_NOT_OK`: set of operation cycle was rejected
Description	Sets an operation cycle state. This API can only be used through the RTE and therefore no declaration is exported via Dem.h. The interface has an asynchronous behavior to avoid exceeding of typical timing requirements on APIs if a large number of events has to be processed and during the re-initializations of the related monitors. The asynchronous acknowledgements are the related InitMonitorForEvent callbacks.	
Available via	Dem.h	

⌋

8.3.2.22 Dem_PrestoreFreezeFrame

[SWS_Dem_00188] Definition of API function Dem_PrestoreFreezeFrame ⌈

Service Name	Dem_PrestoreFreezeFrame	

Syntax	`Std_ReturnType Dem_PrestoreFreezeFrame (`
	`  Dem_EventIdType EventId`
	`)`	

Service ID [hex]	0x06	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK` Freeze frame prestorage was successful `E_NOT_OK` Freeze frame prestorage failed
Description	Captures the freeze frame data for a specific event. This API can only be used through the RTE and therefore no declaration is exported via Dem.h. API Availability: This API will be available only if ({ecuc(Dem/DemConfigSet/DemEvent Parameter.DemFFPrestorageSupported)} == true)	
Available via	Dem.h	

⌋

8.3.2.23 Dem_SelectDTC

[SWS_Dem_91016] Definition of API function Dem_SelectDTC ⌈

Service Name	Dem_SelectDTC	
Syntax	`Std_ReturnType Dem_SelectDTC ( uint8 ClientId, uint32 DTC, Dem_DTCFormatType DTCFormat, Dem_DTCOriginType DTCOrigin )`	
Service ID [hex]	0xb7	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
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.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: DTC successfully selected. `DEM_BUSY`: Another Dem_SelectDTC or Dem_SelectDTC dependent operation of this client is currently in progress.
Description	Selects a DTC or DTC group as target for further operations.	
Available via	Dem.h	

⌋

8.3.2.24 Dem_SetComponentAvailable

[SWS_Dem_01117] Definition of API function Dem_SetComponentAvailable ⌈

Service Name	Dem_SetComponentAvailable	
Syntax	`Std_ReturnType Dem_SetComponentAvailable ( Dem_ComponentIdType ComponentId, boolean AvailableStatus )`	
Service ID [hex]	0x2b	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ComponentId	Identification of a DemComponent.
	AvailableStatus	This parameter specifies whether the respective Component shall be available (TRUE) or not (FALSE).
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Operation was successful `E_NOT_OK`: Operation failed
Description	Set the availability of a specific DemComponent.	
Available via	Dem.h	

⌋

8.3.2.25 Dem_SetDTCSuppression

[SWS_Dem_01047] Definition of API function Dem_SetDTCSuppression

Upstream requirements: RS_Diag_04126

⌈

Service Name	Dem_SetDTCSuppression	
Syntax	`Std_ReturnType Dem_SetDTCSuppression ( uint8 ClientId, boolean SuppressionStatus )`	
Service ID [hex]	0x33	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
	SuppressionStatus	This parameter specifies whether the respective DTC shall be disabled (TRUE) or enabled (FALSE).
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: The 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. `DEM_PENDING`: The requested value is calculated asynchronously and currently not available. The caller can retry later. `DEM_BUSY`: A different Dem_SelectDTC dependent operation according to [SWS_Dem_01253] of this client is currently in progress.
Description	Set the suppression status of a specific DTC. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemSuppression Support)} == DEM_DTC_SUPPRESSION)	
Available via	Dem.h	

⌋

8.3.2.26 Dem_SetEnableCondition

[SWS_Dem_00201] Definition of API function Dem_SetEnableCondition

Upstream requirements: RS_Diag_04192

⌈

Service Name	Dem_SetEnableCondition
Syntax	`Std_ReturnType Dem_SetEnableCondition ( uint8 EnableConditionID, boolean ConditionFulfilled )`
Service ID [hex]	0x39
Sync/Async	Asynchronous

▽

△

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).
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	In case the enable condition could be set successfully the API call returns `E_OK`. If the setting of the enable condition failed the return value of the function is `E_NOT_OK`.
Description	Sets an enable condition. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral/DemEnable Condition)} != NULL)	
Available via	Dem.h	

⌋

8.3.2.27 Dem_SetEventAvailable

[SWS_Dem_01080] Definition of API function Dem_SetEventAvailable

Upstream requirements: RS_Diag_04126

⌈

Service Name	Dem_SetEventAvailable	
Syntax	`Std_ReturnType Dem_SetEventAvailable ( Dem_EventIdType EventId, boolean AvailableStatus )`	
Service ID [hex]	0x37	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	AvailableStatus	This parameter specifies whether the respective Event shall be available (TRUE) or not (FALSE).
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Request to set the availability status was successful. `E_NOT_OK`: Request to set the availability status not accepted.
Description	Set the available status of a specific Event.	
Available via	Dem.h	

⌋

8.3.2.28 Dem_SetEventConfirmationThresholdCounter

[SWS_Dem_91004] Definition of API function Dem_SetEventConfirmation ThresholdCounter ⌈

Service Name	Dem_SetEventConfirmationThresholdCounter	
Syntax	`Std_ReturnType Dem_SetEventConfirmationThresholdCounter ( Dem_EventIdType EventId, uint8 FailureCycleCounterThreshold )`	
Service ID [hex]	0x57	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	FailureCycleCounter Threshold	Failure cycle counter threshold of event to be set.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Change of threshold was successful. `E_NOT_OK`: Threshold cannot be changed as `DemEventConfirmationThresholdCounterAdaptable` is set to FALSE for this event.
Description	Set the failure confirmation threshold of an event.	
Available via	Dem.h	

⌋

8.3.2.29 Dem_SetEventStatus

[SWS_Dem_00183] Definition of API function Dem_SetEventStatus ⌈

Service Name	Dem_SetEventStatus	
Syntax	`Std_ReturnType Dem_SetEventStatus ( Dem_EventIdType EventId, Dem_EventStatusType EventStatus )`	
Service ID [hex]	0x04	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	EventStatus	Monitor test result
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: set of event status was successful `E_NOT_OK`: Event status setting or processing failed or could not be accepted.
Description	Called by SW-Cs or BSW modules to report monitor status information to the Dem. BSW modules calling Dem_SetEventStatus can safely ignore the return value. This API will be available only if ({Dem/DemConfigSet/DemEventParameter/DemEventReportingType} == STANDARD_REPORTING)	
Available via	Dem.h	

⌋

8.3.2.30 Dem_SetEventStatusWithMonitorData

[SWS_Dem_91037] Definition of API function Dem_SetEventStatusWithMonitorData ⌈

Service Name	Dem_SetEventStatusWithMonitorData	

Syntax	`Std_ReturnType Dem_SetEventStatusWithMonitorData (`
	`  Dem_EventIdType EventId,`
	`  Dem_EventStatusType EventStatus,`
	`  Dem_MonitorDataType monitorData0,`
	`  Dem_MonitorDataType monitorData1`
	`)`	

Service ID [hex]	0xbd	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	EventStatus	Monitor test result
	monitorData0	–
	monitorData1	–
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: set of event status was successful `E_NOT_OK`: Event status setting or processing failed or could not be accepted.
Description	This API will be available only if ({Dem/DemConfigSet/DemEventParameter/DemEventReportingType} == STANDARD_REPORTING_WITH_MONITOR_DATA)	
Available via	Dem.h	

⌋

8.3.2.31 Dem_SetStorageCondition

[SWS_Dem_00556] Definition of API function Dem_SetStorageCondition

Upstream requirements: RS_Diag_04219

⌈

Service Name	Dem_SetStorageCondition	

Syntax	`Std_ReturnType Dem_SetStorageCondition (`
	`  uint8 StorageConditionID,`
	`  boolean ConditionFulfilled`
	`)`	

Service ID [hex]	0x38	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	StorageConditionID	This parameter identifies the storage condition.
	ConditionFulfilled	This parameter specifies whether the storage condition assigned to the StorageConditionID is fulfilled (TRUE) or not fulfilled (FALSE).
Parameters (inout)	None	

▽

△

Parameters (out)	None	
Return value	Std_ReturnType	In case the storage condition could be set successfully the API call returns `E_OK`. If the setting of the storage condition failed the return value of the function is `E_NOT_OK`.
Description	Sets a storage condition. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral/DemStorage Condition)} != NULL)	
Available via	Dem.h	

⌋

8.3.2.32 Dem_SetWIRStatus

[SWS_Dem_00839] Definition of API function Dem_SetWIRStatus ⌈

Service Name	Dem_SetWIRStatus	
Syntax	`Std_ReturnType Dem_SetWIRStatus ( Dem_EventIdType EventId, boolean WIRStatus )`	
Service ID [hex]	0x7a	
Sync/Async	Asynchronous	
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"
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Request to set the WIR status was successful. `E_NOT_OK`: Request to set the WIR status was not accepted (e.g. disabled controlDTCSetting) and should be repeated.
Description	Sets the WIR status bit via failsafe SW-Cs. This API can only be used through the RTE and therefore no declaration is exported via Dem.h.	
Available via	Dem.h	

⌋

8.3.2.33 Dem_InitAllNvmBlocks

[SWS_Dem_91199] Definition of API function Dem_InitAllNvmBlocks ⌈

Service Name	Dem_InitAllNvmBlocks	
Syntax	`Std_ReturnType Dem_InitAllNvmBlocks ( void )`	
Service ID [hex]	0xc6	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Request to initialize all Dem related NvM data accepted `E_NOT_OK`: Request to initialize all Dem related NvM data not accepted.
Description	Request to initialize all Dem related NvM data blocks.	
Available via		

⌋

8.3.2.34 Dem_InitAllNvmBlocksGetResult

[SWS_Dem_91200] Definition of API function Dem_InitAllNvmBlocksGetResult ⌈

Service Name	Dem_InitAllNvmBlocksGetResult	
Syntax	`Std_ReturnType Dem_InitAllNvmBlocksGetResult ( void )`	
Service ID [hex]	0xc7	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: All Dem related NvM blocks are initialized successfully. `DEM_PENDING`: InitAllNvmBlocks is currently in progress. The caller shall call this function again at a later moment to get the result. `E_NOT_OK`: Request to initialize Dem related NvM data failed (e.g., in case Dem_InitAllNvmBlocks was not called successfully or NvM storage operations are returning an error).
Description	Queries the result of a of a corresponding Dem_InitAllNvmBlocks request.	
Available via		

⌋

8.3.3 Interface Dcm <=> Dem

8.3.3.1 Access DTCs and Status Information

8.3.3.1.1 Dem_GetTranslationType

[SWS_Dem_00230] Definition of API function Dem_GetTranslationType ⌈

Service Name	Dem_GetTranslationType	
Syntax	`Dem_DTCTranslationFormatType Dem_GetTranslationType ( uint8 ClientId )`	
Service ID [hex]	0x3c	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Dem_DTCTranslation FormatType	Returns the configured DTC translation format. A combination of different DTC formats is not possible.
Description	Gets the supported DTC formats of the ECU.The supported formats are configured via Dem TypeOfDTCSupported.	
Available via	Dem.h	

⌋

8.3.3.1.2 Dem_GetDTCStatusAvailabilityMask

[SWS_Dem_00213] Definition of API function Dem_GetDTCStatusAvailability Mask

Upstream requirements: RS_Diag_04067

⌈

Service Name	Dem_GetDTCStatusAvailabilityMask	
Syntax	`Std_ReturnType Dem_GetDTCStatusAvailabilityMask ( uint8 ClientId, Dem_UdsStatusByteType* DTCStatusMask, Dem_DTCOriginType DTCOrigin )`	
Service ID [hex]	0x16	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
	DTCOrigin	This parameter selects the event memory the DTCStatus AvailabilityMask is requested for. Only the values DEM_DTC_ ORIGIN_PRIMARY_ MEMORY and DEM_DTC_ORIGIN_ USERDEFINED_MEMORY_<Name> are valid.
Parameters (inout)	None	

▽

△

Parameters (out)	DTCStatusMask	DTCStatusMask The value DTCStatusMask indicates the supported DTC status bits from the Dem. All supported information is indicated by setting the corresponding status bit to 1. See ISO14229-1.
Return value	Std_ReturnType	`E_OK`: get of DTC status mask was successful `E_NOT_OK`: get of DTC status mask failed
Description	Gets the DTC Status availability mask of the selected fault memory.	
Available via	Dem.h	

⌋

8.3.3.1.3 Dem_GetStatusOfDTC

[SWS_Dem_00212] Definition of API function Dem_GetStatusOfDTC

Upstream requirements: RS_Diag_04067

⌈

Service Name	Dem_GetStatusOfDTC	

Syntax	`Std_ReturnType Dem_GetStatusOfDTC (`
	`  uint8 ClientId,`
	`  uint8* DTCStatus`
	`)`	

Service ID [hex]	0x15	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
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 is other than E_OK the out parameters of this function do not contain valid data.
Return value	Std_ReturnType	`E_OK`: The 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 `DEM_PENDING`: Retrieving the DTC status is currently in progress. The caller shall call this function again at a later moment. `DEM_NO_SUCH_ELEMENT` - Selected DTC does not have an assigned DTC status. `DEM_BUSY`: A different Dem_SelectDTC dependent operation according to [SWS_Dem_01253] of this client is currently in progress.
Description	Gets the status of a DTC. For large configurations and DTC-calibration, the interface behavior can be asynchronous (splitting the DTC-search into segments). The DTCs of OBD Events Suppression shall be reported as Dem_WRONG_DTC.	
Available via	Dem.h	

⌋

8.3.3.1.4 Dem_GetSeverityOfDTC

[SWS_Dem_00232] Definition of API function Dem_GetSeverityOfDTC

Upstream requirements: RS_Diag_04071

⌈

Service Name	Dem_GetSeverityOfDTC	

Syntax	`Std_ReturnType Dem_GetSeverityOfDTC (`
	`  Dem_DTCSeverityType* DTCSeverity,`
	`  uint8 ClientId`
	`)`	

Service ID [hex]	0x0e	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	DTCSeverity	This parameter contains the DTCSeverity according to ISO 14229-1.
Return value	Std_ReturnType	E_OK: The DTC severity is correctly provided in the DTCSeverity 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 DEM_PENDING: Retrieving the DTC is currently in progress. The caller shall call this function again at a later moment. DEM_BUSY: A different Dem_SelectDTC dependent operation according to [SWS_Dem_01253] of this client is currently in progress.
Description	Gets the severity of the requested DTC. For large configurations and DTC-calibration, the interface behavior can be asynchronous (splitting the DTC-search into segments).	
Available via	Dem.h	

⌋

8.3.3.1.5 Dem_GetFunctionalUnitOfDTC

[SWS_Dem_00594] Definition of API function Dem_GetFunctionalUnitOfDTC

Upstream requirements: RS_Diag_04156

⌈

Service Name	Dem_GetFunctionalUnitOfDTC	

Syntax	`Std_ReturnType Dem_GetFunctionalUnitOfDTC (`
	`  uint8 ClientId,`
	`  uint8* DTCFunctionalUnit`
	`)`	

Service ID [hex]	0x34	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.

▽

△

Parameters (inout)	None	
Parameters (out)	DTCFunctionalUnit	Functional unit value of this DTC
Return value	Std_ReturnType	E_OK: The DTC functional unit is provided in the parameter DTCFunctionalUnit. 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 DEM_PENDING: Retrieving the DTC functional unit is currently in progress. The caller shall call this function again at a later moment. DEM_BUSY: A different Dem_SelectDTC dependent operation according to [SWS_Dem_01253] of this client is currently in progress.
Description	Gets the functional unit of the requested DTC.	
Available via	Dem.h	

⌋

8.3.3.1.6 Dem_SetDTCFilter

[SWS_Dem_00208] Definition of API function Dem_SetDTCFilter

Upstream requirements: RS_Diag_04205

⌈

Service Name	Dem_SetDTCFilter	

Syntax	`Std_ReturnType Dem_SetDTCFilter (`
	`  uint8 ClientId,`
	`  uint8 DTCStatusMask,`
	`  Dem_DTCFormatType DTCFormat,`
	`  Dem_DTCOriginType DTCOrigin,`
	`  boolean FilterWithSeverity,`
	`  Dem_DTCSeverityType DTCSeverityMask,`
	`  boolean FilterForFaultDetectionCounter`
	`)`	

Service ID [hex]	0x13	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
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. If passed value does not fit to Configuration, the DET error DEM_E_WRONG_CONFIGURATION shall be reported, e.g. if DTCFormat "DEM_DTC_FORMAT_OBD" is passed, but OBD is not supported per configuration.

▽

△

	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. If passed value does not fit to Configuration, the DET error DEM_E_WRONG_CONFIGURATION shall be reported.
	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 used for filtering. FALSE: severity information is not used for filtering.
	DTCSeverityMask	Contains the DTCSeverityMask according to ISO14229-1.
	FilterForFaultDetection Counter	This flag defines whether the fault detection counter information shall be used for filtering. This is to allow for coexistence of DTCs with and without fault detection counter information. If fault detection counter information is filter criteria, only those DTCs with a fault detection counter value between 1 and 0x7E shall be reported. Remark: If the event does not use the debouncing inside Dem, then the Dem must request this information via Get FaultDetectionCounter. TRUE: fault detection counter is used for filtering. FALSE: fault detection counter is not used for filtering.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Filter was successfully set. `E_NOT_OK`: Indicates a wrong DTCOrigin or DTCFormat
Description	Sets the DTC Filter. The server shall perform a bit-wise logical AND-ing operation between the parameter DTCStatusMask and the current UDS status in the server. In addition to the DTCStatus AvailabilityMask, the server shall return all DTCs for which the result of the AND-ing operation is non-zero [i.e. (statusOfDTC & DTCStatusMask) != 0]. The server shall process only the DTC Status bits that it is supporting. OBD Events Suppression shall be ignored for this computation. If no DTCs within the server match the masking criteria specified in the client's request, no DTC or status information shall be provided following the DTCStatusAvailabilityMask byte in the positive response message (((statusOfDTC & DTCStatusMask) != 0) && ((severity & DTCSeverityMask) != 0)) == TRUE	
Available via	Dem.h	

⌋

8.3.3.1.7 Dem_GetNumberOfFilteredDTC

[SWS_Dem_00214] Definition of API function Dem_GetNumberOfFilteredDTC ⌈

Service Name	Dem_GetNumberOfFilteredDTC

Syntax	`Std_ReturnType Dem_GetNumberOfFilteredDTC (`
	`  uint8 ClientId,`
	`  uint16* NumberOfFilteredDTC`
	`)`

▽

△

Service ID [hex]	0x17	
Sync/Async	Asynchronous	
Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	NumberOfFilteredDTC	The number of DTCs matching the defined status mask.
Return value	Std_ReturnType	`E_OK`: Getting number of filtered DTCs was successful `E_NOT_OK`: No DTC filter set `DEM_PENDING`: The requested operation is currently in progress. The caller shall call this function again at a later moment.
Description	Gets the number of a filtered DTC.	
Available via	Dem.h	

⌋

8.3.3.1.8 Dem_GetNextFilteredDTC

[SWS_Dem_00215] Definition of API function Dem_GetNextFilteredDTC ⌈

Service Name	Dem_GetNextFilteredDTC	
Syntax	`Std_ReturnType Dem_GetNextFilteredDTC ( uint8 ClientId, uint32* DTC, uint8* DTCStatus )`	
Service ID [hex]	0x18	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
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 the out parameters of this function do not contain valid data.
	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 is other than E_OK the out parameters of this function do not contain valid data.
Return value	Std_ReturnType	`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 `DEM_PENDING`: The requested operation is currently in progress. The caller shall call this function again at a later moment. Note that according to [SWS_Dem_00653] this return value is not always allowed.
Description	Gets the next filtered DTC matching the filter criteria. For UDS services, the interface has an asynchronous behavior, because a large number of DTCs has to be processed.	
Available via	Dem.h	

⌋

8.3.3.1.9 Dem_GetNextFilteredDTCAndFDC

[SWS_Dem_00227] Definition of API function Dem_GetNextFilteredDTCAndFDC ⌈

Service Name	Dem_GetNextFilteredDTCAndFDC	
Syntax	`Std_ReturnType Dem_GetNextFilteredDTCAndFDC ( uint8 ClientId, uint32* DTC, sint8* DTCFaultDetectionCounter )`	
Service ID [hex]	0x3b	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
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 the out parameters of this function do not contain valid data.
	DTCFaultDetection Counter	This parameter receives the Fault Detection Counter information of the requested DTC. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data. -128dec...127dec PASSED...FAILED according to ISO 14229-1
Return value	Std_ReturnType	`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 `DEM_PENDING`: The requested operation is asynchronously processed is currently in progress. The caller shall call this function again at a later moment.
Description	Gets the next filtered DTC and its associated Fault Detection Counter (FDC) matching the filter criteria. The interface has an asynchronous behavior, because a large number of DTCs has to be processed and the FDC might be received asynchronously from a SW-C, too.	
Available via	Dem.h	

⌋

8.3.3.1.10 Dem_GetNextFilteredDTCAndSeverity

[SWS_Dem_00281] Definition of API function Dem_GetNextFilteredDTCAndSeverity ⌈

Service Name	Dem_GetNextFilteredDTCAndSeverity
Syntax	`Std_ReturnType Dem_GetNextFilteredDTCAndSeverity ( uint8 ClientId, uint32* DTC, uint8* DTCStatus, Dem_DTCSeverityType* DTCSeverity, uint8* DTCFunctionalUnit )`
Service ID [hex]	0x3d
Sync/Async	Asynchronous

▽

△

Reentrancy	Non Reentrant Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
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 the out parameters of this function do not contain valid data.
	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 is other than E_OK the out parameters of this function do not contain valid data.
	DTCSeverity	Receives the severity value returned by the function. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.
	DTCFunctionalUnit	Receives the functional unit value returned by the function. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.
Return value	Std_ReturnType	`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 `DEM_PENDING`: The requested operation is currently in progress. The caller shall call this function again at a later moment.
Description	Gets the next filtered DTC and its associated Severity matching the filter criteria. The interface has an asynchronous behavior, because a large number of DTCs has to be processed.	
Available via	Dem.h	

⌋

8.3.3.1.11 Dem_SetFreezeFrameRecordFilter

[SWS_Dem_00209] Definition of API function Dem_SetFreezeFrameRecordFilter

⌈

Service Name	Dem_SetFreezeFrameRecordFilter	
Syntax	`Std_ReturnType Dem_SetFreezeFrameRecordFilter ( uint8 ClientId, Dem_DTCFormatType DTCFormat )`	
Service ID [hex]	0x3f	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
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 (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	Status of the operation to (re-)set a freeze frame record filter. `E_OK`: Filter is accepted `E_NOT_OK`: Wrong filter selected
Description	Sets a freeze frame record filter.	
Available via	Dem.h	

⌋

8.3.3.1.12 Dem_GetNextFilteredRecord

[SWS_Dem_00224] Definition of API function Dem_GetNextFilteredRecord ⌈

Service Name	Dem_GetNextFilteredRecord	
Syntax	`Std_ReturnType Dem_GetNextFilteredRecord ( uint8 ClientId, uint32* DTC, uint8* RecordNumber )`	
Service ID [hex]	0x3a	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	DTC	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	Std_ReturnType	Status of the operation to retrieve a DTC and its associated snapshot record number from the Dem. `E_OK`: Returned next filtered element `DEM_NO_SUCH_ELEMENT`: No further element (matching the filter criteria) found `DEM_PENDING`: The requested value is calculated asynchronously and currently not available. The caller can retry later. Only used by asynchronous interfaces.
Description	Gets the next freeze frame record number and its associated DTC stored in the event memory. The interface has an asynchronous behavior, because NvRAM access might be required.	
Available via	Dem.h	

⌋

8.3.3.1.13 Dem_GetDTCByOccurrenceTime

[SWS_Dem_00218] Definition of API function Dem_GetDTCByOccurrenceTime ⌈

Service Name	Dem_GetDTCByOccurrenceTime	
Syntax	`Std_ReturnType Dem_GetDTCByOccurrenceTime ( uint8 ClientId, Dem_DTCRequestType DTCRequest, uint32* DTC )`	
Service ID [hex]	0x19	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
	DTCRequest	This parameter defines the request type of the DTC.
Parameters (inout)	None	

▽

△

Parameters (out)	DTC	Receives the DTC value in UDS format returned by the function. If the return value of the function is other than DEM_OCCURR_OK this parameter does not contain valid data.
Return value	Std_ReturnType	`E_OK`: get of DTC was successful `E_NOT_OK`: the call was not successful `DEM_NO_SUCH_ELEMENT`: The requested element is not stored
Description	Gets the DTC by occurrence time. There is no explicit parameter for the DTC-origin as the origin always is DEM_DTC_ORIGIN_PRIMARY_MEMORY.	
Available via	Dem.h	

⌋

8.3.3.1.14 Dem_SetDTCFilterByExtendedDataRecordNumber

[SWS_Dem_91192] Definition of API function Dem_SetDTCFilterByExtendedDataRecordNumber

Upstream requirements: RS_Diag_04250

⌈

Service Name	Dem_SetDTCFilterByExtendedDataRecordNumber	
Syntax	`Std_ReturnType Dem_SetDTCFilterByExtendedDataRecordNumber ( uint8 ClientId, Dem_DTCFormatType DTCFormat, uint8 ExtendedDataRecordNumber )`	
Service ID [hex]	0xc1	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module
	DTCFormat	Defines the DTC format of the requested data. Valid selections are DEM_DTC_FORMAT_UDS and DEM_DTC_FORMAT_OBD_3BYTE.
	ExtendedDataRecord Number	he extended data record number the filter is set for. Valid values are within the range of 0x01 and 0xFD.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: DTC filter for requested extended data record successfully set. `E_NOT_OK`: Indicates an invalid extended data record number was selected.
Description	Sets the DTC Filter based on a given extended data record on the primary fault memory. The server selects all DTCs that have a matching extended data record.	
Available via	Dem_Dcm.h	

⌋

8.3.3.1.15 Dem_SetDTCFilterByReadinessGroup

[SWS_Dem_91193] Definition of API function Dem_SetDTCFilterByReadiness Group ⌈

Service Name	Dem_SetDTCFilterByReadinessGroup	
Syntax	`Std_ReturnType Dem_SetDTCFilterByReadinessGroup ( uint8 ClientID, Dem_DTCFormatType DTCFormat, Dem_EventOBDReadinessGroupType ReadinessGroupNumber )`	
Service ID [hex]	0xc2	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientID	Unique client id, assigned to the instance of the calling module.
	DTCFormat	Defines the DTC format of the requested data. Valid selections are DEM_DTC_FORMAT_UDS and DEM_DTC_FORMAT_OBD_3BYTE.
	ReadinessGroupNumber	Specifies the DTC readiness group number defined by SAE J1979-DA.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: DTC filter for requested DTC readiness group successfully set. `E_NOT_OK`: Indicates an invalid DTC readiness group was selected.
Description	Sets the DTC Filter based on a given DTC readiness group on the primary fault memory. The server selects all DTCs that have this DTC readiness group configured.	
Available via	Dem_Dcm.h	

⌋

8.3.3.2 Access extended data records and FreezeFrame data

8.3.3.2.1 Dem_DisableDTCRecordUpdate

[SWS_Dem_00233] Definition of API function Dem_DisableDTCRecordUpdate

Upstream requirements: RS_Diag_04192

⌈

Service Name	Dem_DisableDTCRecordUpdate	
Syntax	`Std_ReturnType Dem_DisableDTCRecordUpdate ( uint8 ClientId )`	
Service ID [hex]	0x1a	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	None	

▽

△

Return value	Std_ReturnType	E_OK: Event memory update successfully disabled E_NOT_OK: No DTC selected 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 DEM_PENDING: Disabling the DTC record update is currently in progress. The caller shall call this function again at a later moment. DEM_BUSY: A different Dem_SelectDTC dependent operation according to [SWS_Dem_01253] of this client is currently in progress.
Description	Disables the event memory update of a specific DTC (only one at one time).	
Available via	Dem.h	

⌋

8.3.3.2.2 Dem_EnableDTCRecordUpdate

[SWS_Dem_00234] Definition of API function Dem_EnableDTCRecordUpdate ⌈

Service Name	Dem_EnableDTCRecordUpdate	
Syntax	`Std_ReturnType Dem_EnableDTCRecordUpdate ( uint8 ClientId )`	
Service ID [hex]	0x1b	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: DTC record successfully updated. E_NOT_OK: No DTC selected.
Description	Enables the event memory update of the DTC disabled by Dem_DisableDTCRecordUpdate() before. Note: As the call to Dem_EnableDTCRecordUpdate is the last action in the sequence of disabling/enabling a DTC, the caller is not interested in the return value. Therefore E_OK should be returned even if the DTC is enabled asynchronously.	
Available via	Dem.h	

⌋

8.3.3.2.3 Dem_GetSizeOfExtendedDataRecordSelection

[SWS_Dem_00240] Definition of API function Dem_GetSizeOfExtendedDataRecordSelection ⌈

Service Name	Dem_GetSizeOfExtendedDataRecordSelection	
Syntax	`Std_ReturnType Dem_GetSizeOfExtendedDataRecordSelection ( uint8 ClientId, uint32* SizeOfExtendedDataRecord )`	
Service ID [hex]	0x21	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	SizeOfExtendedDataRecord	Size of the requested extended data record(s) including record number. The format for a single ExtendedDataRecord is: {Record Number, data[1], ..., data[N]}
Return value	Std_ReturnType	E_OK: Size returned successfully E_NOT_OK : selection function is not called. DEM_PENDING: The requested value is calculated asynchronously and currently not available. The caller can retry later. DEM_WRONG_DTC: DTC value not existing DEM_WRONG_DTCORIGIN: Wrong DTC origin DEM_NO_SUCH_ELEMENT: Record number is not supported by configuration and therefore invalid
Description	Gets the size of Extended Data Record by DTC selected by the call of Dem_SelectExtendedDataRecord.	
Available via	Dem.h	

⌋

8.3.3.2.4 Dem_GetSizeOfFreezeFrameSelection

[SWS_Dem_00238] Definition of API function Dem_GetSizeOfFreezeFrameSelection ⌈

Service Name	Dem_GetSizeOfFreezeFrameSelection	
Syntax	`Std_ReturnType Dem_GetSizeOfFreezeFrameSelection ( uint8 ClientId, uint32* SizeOfFreezeFrame )`	
Service ID [hex]	0x1f	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	SizeOfFreezeFrame	Number of bytes in the requested freeze frame record.

▽

△

Return value	Std_ReturnType	E_OK: Size returned successfully E_NOT_OK : selection function is not called. DEM_PENDING: The requested value is calculated asynchronously and currently not available. The caller can retry later. DEM_WRONG_DTC: DTC value not existing DEM_WRONG_DTCORIGIN: Wrong DTC origin DEM_NO_SUCH_ELEMENT: Record number is not supported by configuration and therefore invalid
Description	Gets the size of freeze frame data by DTC selected by the call of Dem_SelectFreezeFrame Data.	
Available via	Dem.h	

⌋

8.3.3.2.5 Dem_GetNextExtendedDataRecord

[SWS_Dem_00239] Definition of API function Dem_GetNextExtendedDataRecord

⌈

Service Name	Dem_GetNextExtendedDataRecord	
Syntax	`Std_ReturnType Dem_GetNextExtendedDataRecord ( uint8 ClientId, uint8* DestBuffer, uint16* BufSize )`	
Service ID [hex]	0x20	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	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 to. The format is: {ExtendedDataRecordNumber, data[0], data[1], ..., data[n]}
Return value	Std_ReturnType	E_OK: Size and buffer successfully returned. E_NOT_OK: selection function is not called. DEM_BUFFER_TOO_SMALL: provided buffer size too small. DEM_PENDING: The requested value is calculated asynchronously and currently not available. The caller can retry later. DEM_WRONG_DTC: DTC value not existing DEM_WRONG_DTCORIGIN: Wrong DTC origin DEM_NO_SUCH_ELEMENT: Found no (further) element matching the filter criteria
Description	Gets extended data record for the DTC selected by Dem_SelectExtendedDataRecord. The function stores the data in the provided DestBuffer.	
Available via	Dem.h	

⌋

8.3.3.2.6 Dem_GetNextFreezeFrameData

[SWS_Dem_00236] Definition of API function Dem_GetNextFreezeFrameData ⌈

Service Name	Dem_GetNextFreezeFrameData	
Syntax	`Std_ReturnType Dem_GetNextFreezeFrameData ( uint8 ClientId, uint8* DestBuffer, uint16* BufSize )`	
Service ID [hex]	0x1d	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	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 record shall be written to. The format is: {RecordNumber, NumOfDIDs, DID[1], data[1], ..., DID[N], data[N]}
Return value	Std_ReturnType	`E_OK`: Size and buffer successfully returned. `DEM_BUFFER_TOO_SMALL`: provided buffer size too small `DEM_PENDING`: The requested value is calculated asynchronously and currently not available. The caller can retry later. `DEM_WRONG_DTC`: DTC value not existing `E_NOT_OK` : selection function is not called. `DEM_WRONG_DTCORIGIN`: Wrong DTC origin `DEM_NO_SUCH_ELEMENT`: Found no (further) element matching the filter criteria
Description	Gets freeze frame data by the DTC selected by Dem_SelectFreezeFrameData. The function stores the data in the provided DestBuffer.	
Available via	Dem.h	

⌋

8.3.3.2.7 Dem_SelectExtendedDataRecord

[SWS_Dem_91017] Definition of API function Dem_SelectExtendedDataRecord ⌈

Service Name	Dem_SelectExtendedDataRecord	
Syntax	`Std_ReturnType Dem_SelectExtendedDataRecord ( uint8 ClientId, uint8 ExtendedDataNumber )`	
Service ID [hex]	0xba	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.

▽

△

	ExtendedDataNumber	Identification/Number of requested extended data record. For primary fault memory the value of 0xFE is explictely allowed to request all regulated emissions OBD DTC extended data records. The value of 0xFF is explicitely allowed in all fault memories to retrieve the data of all extended datal records.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Extended data record successfully selected. DEM_WRONG_DTC: Selected DTC value in selected format does not exist. DEM_WRONG_DTCORIGIN: Selected DTCOrigin does not exist. DEM_PENDING: Selecting the extended data record is currently in progress. The caller shall call this function again at a later moment. DEM_BUSY: A different Dem_SelectDTC dependent operation according to [SWS_Dem_01253] of this client is currently in progress.
Description	Sets the filter to be used by Dem_GetNextExtendedDataRecord and Dem_GetSizeOfExtended DataRecordSelection.	
Available via	Dem.h	

⌋

8.3.3.2.8 Dem_SelectFreezeFrameData

[SWS_Dem_91015] Definition of API function Dem_SelectFreezeFrameData ⌈

Service Name	Dem_SelectFreezeFrameData	

Syntax	`Std_ReturnType Dem_SelectFreezeFrameData (`
	`  uint8 ClientId,`
	`  uint8 RecordNumber`
	`)`	

Service ID [hex]	0xb9	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
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 WWH-OBD snapshot record.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Freeze frame data successfully selected. DEM_WRONG_DTC: Selected DTC value in selected format does not exist. DEM_WRONG_DTCORIGIN: Selected DTCOrigin does not exist. DEM_PENDING: Selecting the freeze frame is currently in progress. The caller shall call this function again at a later moment. DEM_BUSY: A different Dem_SelectDTC dependent operation according to [SWS_Dem_01253] of this client is currently in progress.
Description	Sets the filter to be used by Dem_GetNextFreezeFrameData and Dem_GetSizeOfFreezeFrame Selection.	

▽

△

Available via	Dem.h

⌋

8.3.3.2.9 Dem_GetNumberOfFreezeFrameRecords

[SWS_Dem_91191] Definition of API function Dem_GetNumberOfFreezeFrame Records ⌈

Service Name	Dem_GetNumberOfFreezeFrameRecords	
Syntax	`Std_ReturnType Dem_GetNumberOfFreezeFrameRecords ( uint8 ClientId, uint16* NumberOfFilteredRecords )`	
Service ID [hex]	0x5a	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	NumberOfFiltered Records	Number of all freeze frame records currently stored in the primary event memory.
Return value	Std_ReturnType	Status of the operation E_OK: Returned correctly the number of freeze frame records DEM_PENDING: The requested value is calculated asynchronously and currently not available. The caller can retry later
Description	This function returns the number of all freeze frame records currently stored in the primary event memory	
Available via	Dem.h	

⌋

8.3.3.3 DTC storage

8.3.3.3.1 Dem_DisableDTCSetting

[SWS_Dem_00242] Definition of API function Dem_DisableDTCSetting

Upstream requirements: RS_Diag_04150

⌈

Service Name	Dem_DisableDTCSetting
Syntax	`Std_ReturnType Dem_DisableDTCSetting ( uint8 ClientId )`
Service ID [hex]	0x24
Sync/Async	Asynchronous

▽

△

Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Returned next filtered element DEM_PENDING: The requested operation is currently in progress. The caller shall call this function again at a later moment.
Description	Disables the DTC setting for all DTCs assigned to the DemEventMemorySet of the addressed client.	
Available via	Dem.h	

⌋

8.3.3.3.2 Dem_EnableDTCSetting

[SWS_Dem_00243] Definition of API function Dem_EnableDTCSetting

Upstream requirements: RS_Diag_04192, RS_Diag_04159, RS_Diag_04150

⌈

Service Name	Dem_EnableDTCSetting	

Syntax	`Std_ReturnType Dem_EnableDTCSetting (`
	`  uint8 ClientId`
	`)`	

Service ID [hex]	0x25	
Sync/Async	Asynchronous	
Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: The operation was successful; DEM_PENDING: The started operation is currently in progress. The caller shall call this function again at a later moment.
Description	(Re)-Enables the DTC setting for all DTCs assigned to the DemEventMemorySet of the addressed client.	
Available via	Dem.h	

⌋

8.3.4 OBD-specific Dcm <=> Dem Interfaces

8.3.4.1 Dem_DcmGetInfoTypeValue08

[SWS_Dem_00316] Definition of API function Dem_DcmGetInfoTypeValue08 ⌈

Service Name	Dem_DcmGetInfoTypeValue08	
Syntax	`Std_ReturnType Dem_DcmGetInfoTypeValue08 ( Dcm_OpStatusType OpStatus, uint8* Iumprdata08, uint8* Iumprdata08BufferSize )`	
Service ID [hex]	0x6b	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Only DCM_INITIAL will appear, because this API behaves synchronous.
Parameters (inout)	Iumprdata08BufferSize	The maximum number of data bytes that can be written to the Iumprdata08 Buffer. When the function returns, the value is updated with the actual number of data bytes that are written to the Iumprdata08BufferSize.
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	Std_ReturnType	Always `E_OK` is returned.
Description	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. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.4.2 Dem_DcmGetInfoTypeValue0B

[SWS_Dem_00317] Definition of API function Dem_DcmGetInfoTypeValue0B ⌈

Service Name	Dem_DcmGetInfoTypeValue0B	
Syntax	`Std_ReturnType Dem_DcmGetInfoTypeValue0B ( Dcm_OpStatusType OpStatus, uint8* Iumprdata0B, uint8* Iumprdata0BBufferSize )`	
Service ID [hex]	0x6c	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Only DCM_INITIAL will appear, because this API behaves synchronous.

▽

△

Parameters (inout)	lumprdata0BBufferSize	The maximum number of data bytes that can be written to the lumprdata0B Buffer. When the function returns, the value is updated with the actual number of data bytes that are written to the lumprdata0BBufferSize.
Parameters (out)	lumprdata0B	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	Std_ReturnType	Always `E_OK` is returned.
Description	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. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.4.3 Dem_DcmReadDataOfPID01

[SWS_Dem_00318] Definition of API function Dem_DcmReadDataOfPID01 ⌈

Service Name	Dem_DcmReadDataOfPID01	
Syntax	`Std_ReturnType Dem_DcmReadDataOfPID01 ( uint8* PID01value )`	
Service ID [hex]	0x61	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	PID01value	Buffer containing the contents of PID $01 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 PIDBuffer.
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to report the value of PID $01 computed by the Dem. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.4.4 Dem_DcmReadDataOfPID1C

[SWS_Dem_00325] Definition of API function Dem_DcmReadDataOfPID1C ⌈

Service Name	Dem_DcmReadDataOfPID1C	
Syntax	`Std_ReturnType Dem_DcmReadDataOfPID1C ( uint8* PID1Cvalue )`	
Service ID [hex]	0x63	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	PID1Cvalue	Buffer containing the contents of PID $1C computed by the Dem. The value of PID$1C is configuration within DemOBDCompliancy. 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 PIDBuffer.
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to report the value of PID $1C computed by the Dem. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.4.5 Dem_DcmReadDataOfPID1E

[SWS_Dem_91203] Definition of API function Dem_DcmReadDataOfPID1E ⌈

Service Name	Dem_DcmReadDataOfPID1E	
Syntax	`Std_ReturnType Dem_DcmReadDataOfPID1E ( uint8* PID1Evalue )`	
Service ID [hex]	0xc9	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	PID1Evalue	Buffer containing the 1 byte contents of PID $1E computed by the Dem. The buffer is provided by the Dcm in the appropriate size of one byte for PID $1E. The value provided by the Dem is a bitfield, where Bit 1-7 are reserved and always reported as '0'. Bit 0 has the following values: 0 = PTO not active 1 = PTO active
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to report the value of PID $1E computed by the Dem. The API is needed in OBD-relevant ECUs with PTO support only. API Availability: This API will be available only if ({ecuc(DemDemGeneral.DemPTOSupport)} == true)"	

▽

△

Available via	Dem_Dcm.h

⌋

8.3.4.6 Dem_DcmReadDataOfPID21

[SWS_Dem_00319] Definition of API function Dem_DcmReadDataOfPID21 ⌈

Service Name	Dem_DcmReadDataOfPID21	
Syntax	`Std_ReturnType Dem_DcmReadDataOfPID21 ( uint8* PID21value )`	
Service ID [hex]	0x64	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	PID21value	Buffer containing the contents of PID $21 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 PIDBuffer.
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to report the value of PID $21 computed by the Dem. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.4.7 Dem_DcmReadDataOfPID30

[SWS_Dem_00320] Definition of API function Dem_DcmReadDataOfPID30 ⌈

Service Name	Dem_DcmReadDataOfPID30	
Syntax	`Std_ReturnType Dem_DcmReadDataOfPID30 ( uint8* PID30value )`	
Service ID [hex]	0x65	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	PID30value	Buffer containing the contents of PID $30 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 PIDBuffer.

▽

△

Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to report the value of PID $30 computed by the Dem. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.4.8 Dem_DcmReadDataOfPID31

[SWS_Dem_00321] Definition of API function Dem_DcmReadDataOfPID31 ⌈

Service Name	Dem_DcmReadDataOfPID31	
Syntax	`Std_ReturnType Dem_DcmReadDataOfPID31 ( uint8* PID31value )`	
Service ID [hex]	0x66	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	PID31value	Buffer containing the contents of PID $31 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 PIDBuffer.
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to report the value of PID $31 computed by the Dem. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} == DEM_OBD_MASTER_ECU)	
Available via	Dem_Dcm.h	

⌋

8.3.4.9 Dem_DcmReadDataOfPID41

[SWS_Dem_00322] Definition of API function Dem_DcmReadDataOfPID41 ⌈

Service Name	Dem_DcmReadDataOfPID41
Syntax	`Std_ReturnType Dem_DcmReadDataOfPID41 ( uint8* PID41value )`
Service ID [hex]	0x67
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None

▽

△

Parameters (inout)	None	
Parameters (out)	PID41value	Buffer containing the contents of PID $41 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 PIDBuffer.
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to report the value of PID $41 computed by the Dem. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.4.10 Dem_DcmReadDataOfPID4D

[SWS_Dem_00323] Definition of API function Dem_DcmReadDataOfPID4D ⌈

Service Name	Dem_DcmReadDataOfPID4D	
Syntax	`Std_ReturnType Dem_DcmReadDataOfPID4D ( uint8* PID4Dvalue )`	
Service ID [hex]	0x68	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	PID4Dvalue	Buffer containing the contents of PID $4D 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 PIDBuffer.
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to report the value of PID $4D computed by the Dem. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} == DEM_OBD_MASTER_ECU)	
Available via	Dem_Dcm.h	

⌋

8.3.4.11 Dem_DcmReadDataOfPID4E

[SWS_Dem_00324] Definition of API function Dem_DcmReadDataOfPID4E ⌈

Service Name	Dem_DcmReadDataOfPID4E	
Syntax	`Std_ReturnType Dem_DcmReadDataOfPID4E ( uint8* PID4Evalue )`	
Service ID [hex]	0x69	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	PID4Evalue	Buffer containing the contents of PID $4E 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 PIDBuffer.
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to report the value of PID $4E computed by the Dem. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} == DEM_OBD_MASTER_ECU)	
Available via	Dem_Dcm.h	

⌋

8.3.4.12 Dem_DcmReadDataOfPID91

[SWS_Dem_01187] Definition of API function Dem_DcmReadDataOfPID91 ⌈

Service Name	Dem_DcmReadDataOfPID91	
Syntax	`Std_ReturnType Dem_DcmReadDataOfPID91 ( uint8* PID91value )`	
Service ID [hex]	0x6a	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
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 PIDBuffer.
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to report the value of PID $91 computed by the Dem. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.4.13 Dem_DcmReadDataOfOBDFreezeFrame

[SWS_Dem_00327] Definition of API function Dem_DcmReadDataOfOBDFreeze Frame ⌈

Service Name	Dem_DcmReadDataOfOBDFreezeFrame	
Syntax	`Std_ReturnType Dem_DcmReadDataOfOBDFreezeFrame ( uint8 PID, uint8 DataElementIndexOfPID, uint8* DestBuffer, uint16* BufSize )`	
Service ID [hex]	0x52	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	PID	This parameter is an identifier for a PID as defined in ISO15031-5.
	DataElementIndexOfPID	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 (configured in Dcm, refer to [SWS_Dem_00597]).
Parameters (inout)	DestBuffer	This parameter contains a byte pointer that points to the buffer, to which the data element of the PID 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.
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK` Freeze frame data was successfully reported `E_NOT_OK` Freeze frame data was not successfully reported
Description	Gets data element per PID and index of the most important freeze frame being selected for the output of service $02. The function stores the data in the provided DestBuffer. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.4.14 Dem_DcmGetDTCOfOBDFreezeFrame

[SWS_Dem_00624] Definition of API function Dem_DcmGetDTCOfOBDFreeze Frame ⌈

Service Name	Dem_DcmGetDTCOfOBDFreezeFrame
Syntax	`Std_ReturnType Dem_DcmGetDTCOfOBDFreezeFrame ( uint8 FrameNumber, uint32* DTC, Dem_DTCFormatType DTCFormat )`
Service ID [hex]	0x53
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.
	DTCFormat	Output format of the DTC value.
Parameters (inout)	None	
Parameters (out)	DTC	Diagnostic Trouble Code in ODB format. If the return value of the function is other than E_OK this parameter does not contain valid data.
Return value	Std_ReturnType	`E_OK`: operation was successful `E_NOT_OK`: no DTC available
Description	Gets DTC by freeze frame record number. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.4.15 Dem_DcmGetAvailableOBDMIDs

[SWS_Dem_00766] Definition of API function Dem_DcmGetAvailableOBDMIDs ⌈

Service Name	Dem_DcmGetAvailableOBDMIDs	
Syntax	`Std_ReturnType Dem_DcmGetAvailableOBDMIDs ( uint8 Obdmid, uint32* Obdmidvalue )`	
Service ID [hex]	0xa3	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	Obdmid	Availablity OBDMID ($00,$20, $40...)
Parameters (inout)	None	
Parameters (out)	Obdmidvalue	Bit coded information on the support of OBDMIDs.
Return value	Std_ReturnType	`E_OK`: Report of DTR result successful
Description	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. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.4.16 Dem_DcmGetNumTIDsOfOBDMID

[SWS_Dem_00767] Definition of API function Dem_DcmGetNumTIDsOfOBDMID ⌈

Service Name	Dem_DcmGetNumTIDsOfOBDMID	
Syntax	`Std_ReturnType Dem_DcmGetNumTIDsOfOBDMID ( uint8 Obdmid, uint8* numberOfTIDs )`	
Service ID [hex]	0xa4	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	Obdmid	OBDMID subject of the request to identify the number of assigned TIDs
Parameters (inout)	None	
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	Std_ReturnType	`E_OK`: get number of TIDs successful `E_NOT_OK`: get number of TIDs failed
Description	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. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.4.17 Dem_DcmGetDTRData

[SWS_Dem_00768] Definition of API function Dem_DcmGetDTRData ⌈

Service Name	Dem_DcmGetDTRData	
Syntax	`Std_ReturnType Dem_DcmGetDTRData ( uint8 Obdmid, uint8 TIDindex, uint8* TIDvalue, uint8* UaSID, uint16* Testvalue, uint16* Lowlimvalue, uint16* Upplimvalue )`	
Service ID [hex]	0xa5	
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 (inout)	None	
Parameters (out)	TIDvalue	TID to be put on the tester reponse

▽

△

	UaSID	UaSID to be put on the tester reponse
	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	Std_ReturnType	`E_OK`: Report of DTR result successful `E_NOT_OK`: Report of DTR result failed
Description	Reports a DTR data along with TID-value, UaSID, test result with lower and upper limit. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.4.18 Dem_DcmGetInfoTypeValue79

[SWS_Dem_91050] Definition of API function Dem_DcmGetInfoTypeValue79 ⌈

Service Name	Dem_DcmGetInfoTypeValue79	

Syntax	`Std_ReturnType Dem_DcmGetInfoTypeValue79 (`
	`  Dcm_OpStatusType OpStatus,`
	`  uint8* DataValueBuffer,`
	`  uint8* DataValueBufferSize`
	`)`	

Service ID [hex]	0xc3	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	OpStatus “Only DCM_INITIAL will appear, because this API behaves synchronous.”
Parameters (inout)	DataValueBufferSize	The maximum number of data bytes that can be written to the DataValueBuffer. When the function returns, the value is updated with the actual number of data bytes that are written to the Data ValueBuffer.
Parameters (out)	DataValueBuffer	Buffer containing the contents of the monitor activity denominator. The buffer is provided by the caller with the appropriate size.
Return value	Std_ReturnType	Always `E_OK` is returned
Description	Service to report the value of monitor activity denominator PID computed by the Dem. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if: ({ecucDem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.4.19 Dem_DcmReadDataOfPIDF501

[SWS_Dem_91051] Definition of API function Dem_DcmReadDataOfPIDF501 ⌈

Service Name	Dem_DcmReadDataOfPIDF501	
Syntax	`Std_ReturnType Dem_DcmReadDataOfPIDF501 ( uint8* PIDF501value )`	
Service ID [hex]	0xc4	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	PIDF501value	Buffer containing the contents of the PID 0xF501. The buffer is provided by the Dcm with the appropriate size.
Return value	Std_ReturnType	Always `E_OK` is returned
Description	Function to report the value of PID 0xF501 computed by the Dem. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if: ({ecucDem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem_Dcm.h	

⌋

8.3.5 Interface J1939Dcm <=> Dem

8.3.5.1 Access DTCs and Status Information

8.3.5.1.1 Dem_J1939DcmSetDTCFilter

[SWS_Dem_00970] Definition of API function Dem_J1939DcmSetDTCFilter

Upstream requirements: RS_Diag_04112

⌈

Service Name	Dem_J1939DcmSetDTCFilter	
Syntax	`Std_ReturnType Dem_J1939DcmSetDTCFilter ( Dem_J1939DcmDTCStatusFilterType DTCStatusFilter, Dem_DTCKindType DTCKind, Dem_DTCOriginType DTCOrigin, uint8 ClientId, Dem_J1939DcmLampStatusType* LampStatus )`	
Service ID [hex]	0x90	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	DTCStatusFilter	The following types are available: DEM_J1939DTC_ACTIVE DEM_J1939DTC_PREVIOUSLY_ACTIVE DEM_J1939DTC_PENDING DEM_J1939DTC_PERMANENT DEM_J1939DTC_CURRENTLY_ACTIVE

▽

△

	DTCKind	Defines the functional group of DTCs to be reported (e.g. all DTC, OBD-relevant DTC)
	DTCOrigin	This parameter is used to select the source memory the DTCs shall be read/cleared from.
	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	LampStatus	E_OK: Operation successful DEM_NO_SUCH_ELEMENT: The requested element is not available DEM_PENDING: Operation successful and result pending. DEM_BUFFER_TOO_SMALL: The provided buffer is too small
Return value	Std_ReturnType	`E_OK`: Operation successful `E_NOT_OK`: Filter could not be set
Description	The function sets the DTC filter for a specific node and returns the composite lamp status of the filtered DTCs.	
Available via	Dem_J1939Dcm.h	

⌋

8.3.5.1.2 Dem_J1939DcmGetNumberOfFilteredDTC

[SWS_Dem_00972] Definition of API function Dem_J1939DcmGetNumberOfFilteredDTC ⌈

Service Name	Dem_J1939DcmGetNumberOfFilteredDTC	
Syntax	`Std_ReturnType Dem_J1939DcmGetNumberOfFilteredDTC ( uint16* NumberOfFilteredDTC, uint8 ClientId )`	
Service ID [hex]	0x91	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	NumberOfFilteredDTC	The number of DTCs matching the defined status mask.
Return value	Std_ReturnType	`E_OK`: Operation successful `DEM_PENDING`: Operation successful and result pending.
Description	Gets the number of currently filtered DTCs set by the function Dem_J1939DcmSetDTCFilter.	
Available via	Dem_J1939Dcm.h	

⌋

8.3.5.1.3 Dem_J1939DcmGetNextFilteredDTC

[SWS_Dem_00973] Definition of API function Dem_J1939DcmGetNextFiltered DTC ⌈

Service Name	Dem_J1939DcmGetNextFilteredDTC	
Syntax	`Std_ReturnType Dem_J1939DcmGetNextFilteredDTC ( uint32* J1939DTC, uint8* OccurenceCounter, uint8 ClientId )`	
Service ID [hex]	0x92	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	J1939DTC	Receives the J1939DTC value. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.
	OccurenceCounter	This parameter receives the corresponding occurrence counter. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.
Return value	Std_ReturnType	`E_OK`: Operation successful `DEM_NO_SUCH_ELEMENT`: The requested element is not available `DEM_PENDING`: Operation successful and result pending. `DEM_BUFFER_TOO_SMALL`: The provided buffer is too small
Description	Gets the next filtered J1939 DTC.	
Available via	Dem_J1939Dcm.h	

⌋

8.3.5.1.4 Dem_J1939DcmGetNextFilteredDTCWithSeverity

[SWS_Dem_91205] Definition of API function Dem_J1939DcmGetNextFiltered DTCWithSeverity ⌈

Service Name	Dem_J1939DcmGetNextFilteredDTCWithSeverity	
Syntax	`Std_ReturnType Dem_J1939DcmGetNextFilteredDTCWithSeverity ( uint8 ClientId, uint32* J1939DTC, uint8* OccurenceCounter, Dem_J1939DcmDTCSeverityType* DTCSeverity )`	
Service ID [hex]	0xca	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	J1939DTC	Receives the J1939DTC value. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.

▽

△

	OccurenceCounter	This parameter receives the corresponding occurrence counter. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.
	DTCSeverity	Receives the severity value returned by the function. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.
Return value	Std_ReturnType	`E_OK`: Operation successful DEM_NO_SUCH_ELEMENT: The requested element is not available DEM_PENDING: Operation successful and result pending. DEM_BUFFER_TOO_SMALL: The provided buffer is too small
Description	Gets the next filtered J1939 DTC and its severity.	
Available via	Dem_J1939Dcm.h	

⌋

8.3.5.1.5 Dem_J1939DcmFirstDTCwithLampStatus

[SWS_Dem_00974] Definition of API function Dem_J1939DcmFirstDTCwithLampStatus

Upstream requirements: RS_Diag_04110

⌈

Service Name	Dem_J1939DcmFirstDTCwithLampStatus	
Syntax	`void Dem_J1939DcmFirstDTCwithLampStatus ( uint8 ClientId )`	
Service ID [hex]	0x93	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	The function sets the filter to the first applicable DTC for the DM31 response for a specific node.	
Available via	Dem_J1939Dcm.h	

⌋

8.3.5.1.6 Dem_J1939DcmGetNextDTCwithLampStatus

[SWS_Dem_00975] Definition of API function Dem_J1939DcmGetNextDTCwithLampStatus ⌈

Service Name	Dem_J1939DcmGetNextDTCwithLampStatus	
Syntax	`Std_ReturnType Dem_J1939DcmGetNextDTCwithLampStatus ( Dem_J1939DcmLampStatusType* LampStatus, uint32* J1939DTC, uint8* OccurenceCounter, uint8 ClientId )`	
Service ID [hex]	0x94	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	LampStatus	Receives the lamp status returned by this function. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.
	J1939DTC	Receives the J1939DTC value. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.
	OccurenceCounter	This parameter receives the corresponding occurrence counter. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.
Return value	Std_ReturnType	`E_OK`: Operation successful `DEM_NO_SUCH_ELEMENT`: The requested element is not available `DEM_PENDING`: Operation successful and result pending. `DEM_BUFFER_TOO_SMALL`: The provided buffer is too small
Description	Gets the next filtered J1939 DTC for DM31 including current LampStatus.	
Available via	Dem_J1939Dcm.h	

⌋

8.3.5.2 DTC storage

8.3.5.2.1 Dem_J1939DcmClearDTC

[SWS_Dem_00976] Definition of API function Dem_J1939DcmClearDTC

Upstream requirements: RS_Diag_04112

⌈

Service Name	Dem_J1939DcmClearDTC
Syntax	`Std_ReturnType Dem_J1939DcmClearDTC ( Dem_J1939DcmSetClearFilterType DTCTypeFilter, Dem_DTCOriginType DTCOrigin, uint8 ClientId )`
Service ID [hex]	0x95
Sync/Async	Asynchronous

▽

△

Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	DTCTypeFilter	Defines the type of DTCs to be cleared.
	DTCOrigin	This parameter is used to select the source memory the DTCs shall be read/cleared from.
	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: DTC successfully cleared DEM_WRONG_DTC: Selected DTC value in selected format does not exist. DEM_WRONG_DTCORIGIN: Selected DTCOrigin does not exist 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_NONVOLATILE_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	Clears the status of all event(s) related to the specified DTC(s), as well as all associated event memory entries for these event(s).	
Available via	Dem_J1939Dcm.h	

⌋

8.3.5.2.2 Dem_J1939DcmClearSingleDTC

[SWS_Dem_91202] Definition of API function Dem_J1939DcmClearSingleDTC ⌈

Service Name	Dem_J1939DcmClearSingleDTC	

Syntax	`Std_ReturnType Dem_J1939DcmClearSingleDTC (`
	`  Dem_J1939DcmSetClearFilterType DTCTypeFilter,`
	`  uint32 J1939DTC,`
	`  Dem_DTCOriginType DTCOrigin,`
	`  uint8 ClientId`
	`)`	

Service ID [hex]	0xc8	
Sync/Async	Asynchronous	
Reentrancy	Reentrant for different ClientIds, non reentrant for the same ClientId.	
Parameters (in)	DTCTypeFilter	Defines the type of DTCs to be cleared
	J1939DTC	J1939 Diagnostic Trouble code
	DTCOrigin	This parameter is used to select the source memory the DTCs shall be read/cleared from.
	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	None	

▽

△

Return value	Std_ReturnType	E_OK: DTC successfully cleared DEM_WRONG_DTC: Selected DTC value in selected format does not exist or is an OBD DTC. DEM_PENDING: Clearing the DTCs is currently in progress. The caller shall call this function again at a later moment. DEM_WRONG_DTCORIGIN: Selected DTCOrigin does not exist DEM_CLEAR_FAILED: DTC clearing failed. DEM_WRONG_DTCTYPEFILTER: The state of the requested J1939 DTC does not match the given DTCTypeFilter. DEM_CLEAR_MEMORY_ERROR: An error occurred during erasing a memory location. 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.
Description	Clears the status of all event(s) related to the specified DTC, as well as all associated event memory entries for these event(s).	
Available via	Dem_J1939Dcm.h	

⌋

8.3.5.2.3 Dem_J1939DcmSetFreezeFrameFilter

[SWS_Dem_00977] Definition of API function Dem_J1939DcmSetFreezeFrame Filter

Upstream requirements: RS_Diag_04112

⌈

Service Name	Dem_J1939DcmSetFreezeFrameFilter	
Syntax	`Std_ReturnType Dem_J1939DcmSetFreezeFrameFilter ( Dem_J1939DcmSetFreezeFrameFilterType FreezeFrameKind, uint8 ClientId )`	
Service ID [hex]	0x96	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	FreezeFrameKind	The following types are available: DEM_J1939DCM_FREEZEFRAME DEM_J1939DCM_EXPANDED_FREEZEFRAME DEM_J1939DCM_SPNS_IN_EXPANDED_FREEZEFRAME
	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Operation successful E_NOT_OK: Filter could not be set
Description	The function sets the FreezeFrame filter for a specific node.	
Available via	Dem_J1939Dcm.h	

⌋

8.3.5.2.4 Dem_J1939DcmGetNextFreezeFrame

[SWS_Dem_00978] Definition of API function Dem_J1939DcmGetNextFreeze Frame

Upstream requirements: RS_Diag_04112

⌈

Service Name	Dem_J1939DcmGetNextFreezeFrame	
Syntax	`Std_ReturnType Dem_J1939DcmGetNextFreezeFrame ( uint32* J1939DTC, uint8* OccurenceCounter, uint8* DestBuffer, uint16* BufSize, uint8 ClientId )`	
Service ID [hex]	0x97	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	DestBuffer	This parameter contains a byte pointer that points to the buffer, to which the freeze frame data record shall be written to.
	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 Dest Buffer
Parameters (out)	J1939DTC	Receives the J1939DTC value. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.
	OccurenceCounter	This parameter receives the corresponding occurrence counter. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.
Return value	Std_ReturnType	`E_OK`: Operation successful `DEM_NO_SUCH_ELEMENT`: The requested element is not available `DEM_PENDING`: Operation successful and result pending. `DEM_BUFFER_TOO_SMALL`: The provided buffer is too small
Description	Gets next freeze frame data. The function stores the data in the provided DestBuffer.	
Available via	Dem_J1939Dcm.h	

⌋

8.3.5.2.5 Dem_J1939DcmGetNextSPNInFreezeFrame

[SWS_Dem_00979] Definition of API function Dem_J1939DcmGetNextSPNIn FreezeFrame

Upstream requirements: RS_Diag_04112

⌈

Service Name	Dem_J1939DcmGetNextSPNInFreezeFrame	
Syntax	`Std_ReturnType Dem_J1939DcmGetNextSPNInFreezeFrame ( uint32* SPNSupported, uint8* SPNDataLength, uint8 ClientId )`	
Service ID [hex]	0x98	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	SPNSupported	This parameter contains the next SPN in the ExpandedFreeze Frame
	SPNDataLength	This parameter contains the corresponding data length of the SPN
Return value	Std_ReturnType	`E_OK`: Operation successful `DEM_NO_SUCH_ELEMENT`: The requested element is not available `DEM_PENDING`: Operation successful and result pending.
Description	Gets next SPN.	
Available via	Dem_J1939Dcm.h	

⌋

8.3.5.3 Reporting

8.3.5.3.1 Dem_J1939DcmSetRatioFilter

[SWS_Dem_00980] Definition of API function Dem_J1939DcmSetRatioFilter ⌈

Service Name	Dem_J1939DcmSetRatioFilter	
Syntax	`Std_ReturnType Dem_J1939DcmSetRatioFilter ( uint16* IgnitionCycleCounter, uint16* OBDMonitoringConditionsEncountered, uint8 ClientId )`	
Service ID [hex]	0x99	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	IgnitionCycleCounter	Ignition Cycle Counter

▽

△

	OBDMonitoringConditions Encountered	OBD Monitoring Conditions Encountered
Return value	Std_ReturnType	E_OK: Operation successful E_NOT_OK: Filter could not be set
Description	The function sets the Ratio filter for a specific node and returns the corresponding Ignition Cycle Counter and General Denominator.	
Available via	Dem_J1939Dcm.h	

⌋

8.3.5.3.2 Dem_J1939DcmGetNextFilteredRatio

[SWS_Dem_00981] Definition of API function Dem_J1939DcmGetNextFilteredRatio

Upstream requirements: RS_Diag_04112

⌈

Service Name	Dem_J1939DcmGetNextFilteredRatio	
Syntax	`Std_ReturnType Dem_J1939DcmGetNextFilteredRatio ( uint32* SPN, uint16* Numerator, uint16* Denominator, uint8 ClientId )`	
Service ID [hex]	0x9a	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	SPN	Receives the SPN of the applicaple system monitor. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.
	Numerator	Receives the Numerator of the applicable system monitor. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.
	Denominator	Receives the Denominator of the applicable system monitor. If the return value of the function is other than E_OK the out parameters of this function do not contain valid data.
Return value	Std_ReturnType	E_OK: Operation successful DEM_NO_SUCH_ELEMENT: The requested element is not available DEM_PENDING: Operation successful and result pending.
Description	Gets the next filtered Ratio.	
Available via	Dem_J1939Dcm.h	

⌋

8.3.5.3.3 Dem_J1939DcmReadDiagnosticReadiness1

[SWS_Dem_00982] Definition of API function Dem_J1939DcmReadDiagnosticReadiness1

Upstream requirements: RS_Diag_04113

⌈

Service Name	Dem_J1939DcmReadDiagnosticReadiness1	
Syntax	`Std_ReturnType Dem_J1939DcmReadDiagnosticReadiness1 ( Dem_J1939DcmDiagnosticReadiness1Type* DataValue, uint8 ClientId )`	
Service ID [hex]	0x9b	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	DataValue	Buffer of 8 bytes containing the contents of Diagnostic Readiness 1 (DM05) computed by the Dem.
Return value	Std_ReturnType	`E_OK`: Operation was successful `E_NOT_OK`: Operation failed
Description	Service to report the value of Diagnostic Readiness 1 (DM05) computed by the Dem.	
Available via	Dem_J1939Dcm.h	

⌋

8.3.5.3.4 Dem_J1939DcmReadDiagnosticReadiness2

[SWS_Dem_00983] Definition of API function Dem_J1939DcmReadDiagnosticReadiness2

Upstream requirements: RS_Diag_04113

⌈

Service Name	Dem_J1939DcmReadDiagnosticReadiness2	
Syntax	`Std_ReturnType Dem_J1939DcmReadDiagnosticReadiness2 ( Dem_J1939DcmDiagnosticReadiness2Type* DataValue, uint8 ClientId )`	
Service ID [hex]	0x9c	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	DataValue	Buffer of 8 bytes containing the contents of Diagnostic Readiness 2 (DM21) computed by the Dem.
Return value	Std_ReturnType	`E_OK`: Operation was successful `E_NOT_OK`: Operation failed
Description	Service to report the value of Diagnostic Readiness 2 (DM21) computed by the Dem.	

▽

△

Available via	Dem_J1939Dcm.h

⌋

8.3.5.3.5 Dem_J1939DcmReadDiagnosticReadiness3

[SWS_Dem_00770] Definition of API function Dem_J1939DcmReadDiagnosticReadiness3

Upstream requirements: RS_Diag_04113

⌈

Service Name	Dem_J1939DcmReadDiagnosticReadiness3	
Syntax	`Std_ReturnType Dem_J1939DcmReadDiagnosticReadiness3 ( Dem_J1939DcmDiagnosticReadiness3Type* DataValue, uint8 ClientId )`	
Service ID [hex]	0x9d	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	ClientId to address the J1939 event memory
Parameters (inout)	None	
Parameters (out)	DataValue	Buffer of 8 bytes containing the contents of Diagnostic Readiness 3 (DM26) computed by the Dem.
Return value	Std_ReturnType	`E_OK`: Operation was successful `E_NOT_OK`: Operation failed
Description	Service to report the value of Diagnostic Readiness 3 (DM26) computed by the Dem.	
Available via	Dem_J1939Dcm.h	

⌋

8.3.6 OBD-specific Interfaces

8.3.6.1 Dem_SetEventDisabled

[SWS_Dem_00312] Definition of API function Dem_SetEventDisabled ⌈

Service Name	Dem_SetEventDisabled	
Syntax	`Std_ReturnType Dem_SetEventDisabled ( Dem_EventIdType EventId )`	
Service ID [hex]	0x51	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different EventIds. Non reentrant for the same EventId.	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	

▽

△

Parameters (out)	None	
Return value	Std_ReturnType	`E_OK` set of event to disabled was successfull. `E_NOT_OK` set of event disabled failed
Description	Service for reporting the event as disabled to the Dem for the PID $41 computation. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem.h	

⌋

8.3.6.2 Dem_ReplUMPRFaultDetect

[SWS_Dem_00313] Definition of API function Dem_ReplUMPRFaultDetect ⌈

Service Name	Dem_ReplUMPRFaultDetect	
Syntax	`Std_ReturnType Dem_RepIUMPRFaultDetect ( Dem_RatioIdType RatioID )`	
Service ID [hex]	0x73	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different RatioIDs. Non reentrant for the same RatioID.	
Parameters (in)	RatioID	Ratio Identifier reporting that a respective monitor could have found a fault - only used when interface option "API" is selected
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK` report of IUMPR result was successfully reported
Description	Service for reporting that faults are possibly found because all conditions are fullfilled. API is needed in OBD-relevant ECUs only API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem.h	

⌋

8.3.6.3 Dem_SetIUMPRDenCondition

[SWS_Dem_00733] Definition of API function Dem_SetIUMPRDenCondition

Upstream requirements: RS_Diag_04192

⌈

Service Name	Dem_SetIUMPRDenCondition
Syntax	`Std_ReturnType Dem_SetIUMPRDenCondition ( Dem_IumprDenomCondIdType ConditionId, Dem_IumprDenomCondStatusType ConditionStatus )`

▽

△

Service ID [hex]	0xae	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	ConditionId	Identification of a IUMPR denominator condition ID (General Denominator, Cold start, EVAP, 500mi).
	ConditionStatus	Status of the IUMPR denominator condition (Not-reached, reached, not reachable / inhibited)
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: set of IUMPR denominator condition was successful `E_NOT_OK`: set of IUMPR denominator condition failed or could not be accepted.
Description	In order to communicate the status of the (additional) denominator conditions among the OBD relevant ECUs, the API is used to forward the condition status to a Dem of a particular ECU. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem.h	

⌋

8.3.6.4 Dem_GetIUMPRDenCondition

[SWS_Dem_00734] Definition of API function Dem_GetIUMPRDenCondition ⌈

Service Name	Dem_GetIUMPRDenCondition	

Syntax	`Std_ReturnType Dem_GetIUMPRDenCondition (`
	`  Dem_IumprDenomCondIdType ConditionId,`
	`  Dem_IumprDenomCondStatusType* ConditionStatus`
	`)`	

Service ID [hex]	0xaf	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	ConditionId	Identification of a IUMPR denominator condition ID (General Denominator, Cold start, EVAP, 500mi).
Parameters (inout)	None	
Parameters (out)	ConditionStatus	Status of the IUMPR denominator condition (Not-reached, reached, not reachable / inhibited)
Return value	Std_ReturnType	`E_OK`: get of IUMPR denominator condition status was successful `E_NOT_OK`: get of condition status failed
Description	In order to communicate the status of the (additional) denominator conditions among the OBD relevant ECUs, the API is used to retrieve the condition status from the Dem of the ECU where the conditions are computed. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem.h	

⌋

8.3.6.5 Dem_RepIUMPRDenRelease

[SWS_Dem_00315] Definition of API function Dem_RepIUMPRDenRelease ⌈

Service Name	Dem_RepIUMPRDenRelease	
Syntax	`Std_ReturnType Dem_RepIUMPRDenRelease ( Dem_RatioIdType RatioID )`	
Service ID [hex]	0x72	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	RatioID	Ratio Identifier reporting that specific denominator is released (for physical reasons - e.g. temperature conditions or minimum activity)
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK` report of IUMPR denominator status was successfully reported `E_NOT_OK` report of IUMPR denominator status was not successfully reported
Description	Service is used to release a denominator of a specific monitor. API is needed in OBD-relevant ECUs only API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem.h	

⌋

8.3.6.6 Dem_SetPtoStatus

[SWS_Dem_00627] Definition of API function Dem_SetPtoStatus ⌈

Service Name	Dem_SetPtoStatus	
Syntax	`Std_ReturnType Dem_SetPtoStatus ( boolean PtoStatus )`	
Service ID [hex]	0x79	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	PtoStatus	sets the status of the PTO (TRUE==active; FALSE==inactive)
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	Returns `E_OK` when the new PTO-status has been adopted by the Dem; returns `E_NOT_OK` in all other cases.
Description	API is needed in OBD-relevant ECUs only API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem.h	

⌋

8.3.6.7 Dem_ReadDataOfPID01

[SWS_Dem_01167] Definition of API function Dem_ReadDataOfPID01 ⌈

Service Name	Dem_ReadDataOfPID01	
Syntax	`Std_ReturnType Dem_ReadDataOfPID01 ( uint8* PID01value )`	
Service ID [hex]	0xb3	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	PID01value	Buffer containing the contents of PID $01 computed by the Dem. The buffer is provided by the application with the size of 4 bytes.
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to report the value of PID $01 computed by the Dem. API is needed in OBD relevant ECUs only	
Available via	Dem.h	

⌋

8.3.6.8 Dem_GetDataOfPID21

[SWS_Dem_01093] Definition of API function Dem_GetDataOfPID21 ⌈

Service Name	Dem_GetDataOfPID21	
Syntax	`Std_ReturnType Dem_GetDataOfPID21 ( uint8* PID21value )`	
Service ID [hex]	0xb1	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	PID21value	Content of PID $21 as raw hex value.
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to get the value of PID $21 from the Dem by a software component. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral/DemGeneral OBD.DemOBDCentralizedPID21Handling)} == true) && ({ecuc(Dem/DemGeneral.Dem OBDSupport)} == DEM_OBD_MASTER_ECU)	
Available via	Dem.h	

⌋

8.3.6.9 Dem_SetDataOfPID21

[SWS_Dem_00735] Definition of API function Dem_SetDataOfPID21 ⌈

Service Name	Dem_SetDataOfPID21	
Syntax	`Std_ReturnType Dem_SetDataOfPID21 ( const uint8* PID21value )`	
Service ID [hex]	0xa6	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	PID21value	Buffer containing the contents of PID $21. 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 PIDBuffer.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to set the value of PID $21 in the Dem by a software component. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem.h	

⌋

8.3.6.10 Dem_SetDataOfPID31

[SWS_Dem_00736] Definition of API function Dem_SetDataOfPID31 ⌈

Service Name	Dem_SetDataOfPID31	
Syntax	`Std_ReturnType Dem_SetDataOfPID31 ( const uint8* PID31value )`	
Service ID [hex]	0xa7	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	PID31value	Buffer containing the contents of PID $31. 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 PIDBuffer.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to set the value of PID $31 in the Dem by a software component. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem.h	

⌋

8.3.6.11 Dem_SetDataOfPID4D

[SWS_Dem_00737] Definition of API function Dem_SetDataOfPID4D ⌈

Service Name	Dem_SetDataOfPID4D	
Syntax	`Std_ReturnType Dem_SetDataOfPID4D ( const uint8* PID4Dvalue )`	
Service ID [hex]	0xa8	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	PID4Dvalue	Buffer containing the contents of PID $4D. 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 PIDBuffer.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to set the value of PID $4D in the Dem by a software component. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem.h	

⌋

8.3.6.12 Dem_SetDataOfPID4E

[SWS_Dem_00738] Definition of API function Dem_SetDataOfPID4E ⌈

Service Name	Dem_SetDataOfPID4E	
Syntax	`Std_ReturnType Dem_SetDataOfPID4E ( const uint8* PID4Evalue )`	
Service ID [hex]	0xa9	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	PID4Evalue	Buffer containing the contents of PID $4E. 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 PIDBuffer.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to set the value of PID $4E in the Dem by a software component. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem.h	

⌋

8.3.6.13 Dem_GetCycleQualified

[SWS_Dem_00740] Definition of API function Dem_GetCycleQualified ⌈

Service Name	Dem_GetCycleQualified	
Syntax	`Std_ReturnType Dem_GetCycleQualified ( uint8 OperationCycleId, boolean* isQualified )`	
Service ID [hex]	0xab	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OperationCycleId	Identification of a configured DemOperationCycle.
Parameters (inout)	None	
Parameters (out)	isQualified	TRUE: The dependent operation cylcle is qualified. FALSE: The qualification conditions of the dependent operation cylcle have not been met.
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Returns the qualification state of the dependent operation cycle. API Availability: This API will be available only if any of the ({ecuc(Dem/DemGeneral/Dem OperationCycle.DemLeadingCycleRef)} != NULL)	
Available via	Dem.h	

⌋

8.3.6.14 Dem_SetCycleQualified

[SWS_Dem_91001] Definition of API function Dem_SetCycleQualified ⌈

Service Name	Dem_SetCycleQualified	
Syntax	`Std_ReturnType Dem_SetCycleQualified ( uint8 OperationCycleId )`	
Service ID [hex]	0x56	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	OperationCycleId	Identification of a configured DemOperationCycle
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	Always `E_OK` is returned.
Description	Sets a dependent operation cycle as qualified, so it may be processed along with its leading cycle.	
Available via	Dem.h	

⌋

8.3.6.15 Dem_GetDTCSeverityAvailabilityMask

[SWS_Dem_01168] Definition of API function Dem_GetDTCSeverityAvailability Mask ⌈

Service Name	Dem_GetDTCSeverityAvailabilityMask	
Syntax	`Std_ReturnType Dem_GetDTCSeverityAvailabilityMask ( uint8 ClientId, Dem_DTCSeverityType* DTCSeverityMask )`	
Service ID [hex]	0xb2	
Sync/Async	Synchronous	
Reentrancy	Re-entrant for different ClientIDs, Non re-entrant for same ClientId.	
Parameters (in)	ClientId	Unique client id, assigned to the instance of the calling module.
Parameters (inout)	None	
Parameters (out)	DTCSeverityMask	DTCSeverityMask The value DTCSeverityMask indicates the supported DTC severity bits from the Dem. All supported information is indicated by setting the corresponding status bit to 1. See ISO14229-1.
Return value	Std_ReturnType	`E_OK`: get of DTC severity mask was successful `E_NOT_OK`: get of DTC severity mask failed
Description	Gets the DTC Severity availability mask.	
Available via	Dem.h	

⌋

8.3.6.16 Dem_GetB1Counter

[SWS_Dem_01169] Definition of API function Dem_GetB1Counter ⌈

Service Name	Dem_GetB1Counter	
Syntax	`Std_ReturnType Dem_GetB1Counter ( uint16* B1Counter )`	
Service ID [hex]	0xb4	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	B1Counter	Buffer containing the B1 counter. The buffer is provided by the application with the size of 2 bytes.
Return value	Std_ReturnType	Always `E_OK` is returned, as `E_NOT_OK` will never appear.
Description	Service to report the value of the B1 counter computed by the Dem. API is needed in WWH-OBD relevant ECUs only	
Available via	Dem.h	

⌋

8.3.6.17 Dem_SetDTR

[SWS_Dem_00765] Definition of API function Dem_SetDTR ⌈

Service Name	Dem_SetDTR	
Syntax	`Std_ReturnType Dem_SetDTR ( uint16 DTRId, sint32 TestResult, sint32 LowerLimit, sint32 UpperLimit, Dem_DTRControlType Ctrlval )`	
Service ID [hex]	0xa2	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different DTRIds. Non reentrant for the same DTRId.	
Parameters (in)	DTRId	Identification of a DTR element by assigned DTRId.
	TestResult	Test result of DTR
	LowerLimit	Lower limit of DTR
	UpperLimit	Upper limit of DTR
	Ctrlval	Control value of the DTR to support its interpretation Dem-internally.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Report of DTR result successful `E_NOT_OK`: Report of DTR result failed
Description	Reports a DTR result with lower and upper limit. The internal eventstatus serves as master whether the DTR values are forwarded or ignored, also taking the DTRUpdateKind into account. The EventId that is related to the DTR is assigned per configuration (and derived from ServiceNeeds). Processing takes enable/storage conditions into account. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	
Available via	Dem.h	

⌋

8.4 Scheduled functions

These functions are directly called by Basic Software Scheduler. The following functions shall have no return value and no parameter. All functions shall be non reentrant.

8.4.1 Dem_MainFunction

[SWS_Dem_00266] Definition of scheduled function Dem_MainFunction ⌈

Service Name	Dem_MainFunction
Syntax	`void Dem_MainFunction ( void )`

▽

△

Service ID [hex]	0x55
Description	Processes all not event based Dem internal functions.
Available via	SchM_Dem.h

⌋

[SWS_Dem_00125]

Upstream requirements: SRS_BSW_00373

⌈The function Dem_MainFunction shall process all not event based Dem module internal functions.⌋

For details refer to [6] Chapter 8.4 *"Scheduled functions"*.

8.4.2 Runnable Entity MainFunction

```
RunnableEntity MainFunction
        symbol "Dem_MainFunction"
        canbeInvokedConcurrently = FALSE
        SSCP = port CBEventUdsStatusChanged_*,
            CallbackEventUdsStatusChanged
        SSCP = port GeneralCBStatusEvt,
            GeneralCallbackEventUdsStatusChanged
        SSCP = port CBStatusDTC_*, DTCStatusChanged
        SSCP = port CBDataEvt_*, EventDataChanged
        SSCP = port GeneralCBDataEvt, EventDataChanged
        SSCP = port CBReadData_*, ReadData
```

8.5 Expected interfaces

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

8.5.1 Mandatory interfaces

This chapter defines all interfaces, which are required to fulfill the core functionality of the Dem module.

API function Description

[SWS_Dem_91195] Definition of mandatory interfaces required by module Dem
⌈

API Function	***Header File***	***Description***
There are no mandatory interfaces.		

⌋

8.5.2 Optional interfaces

This chapter defines all interfaces, which are required to fulfill an optional functionality of the Dem module.

[SWS_Dem_00255] Definition of optional interfaces requested by module Dem

Upstream requirements: SRS_BSW_00171

⌈

API Function	*Header File*	*Description*
Dcm_DemTriggerOnDTCStatus	Dcm_Dem.h	Triggers on changes of the UDS status byte. Allows to trigger on ROE Event for subservice On DTCStatusChanged.
Det_ReportError	Det.h	Service to report development errors.
FiM_DemInit	FiM_Dem.h	This service re-initializes the FIM.
FiM_DemTriggerOnComponentStatus	FiM_Dem.h	Triggers on changes of the component failed status.
FiM_DemTriggerOnMonitorStatus	FiM_Dem.h	This service is provided to be called by the Dem in order to inform the Fim about monitor status changes.
J1939Dcm_DemTriggerOnDTCStatus	J1939Dcm_Dem.h	Trigger for DM01 message that a UDS status change has happened.
NvM_GetErrorStatus	NvM.h	Service to read the block dependent error/status information.
NvM_ReadBlock	NvM.h	Service to copy the data of the NV block to its corresponding RAM block.
NvM_SetRamBlockStatus	NvM.h	Service for setting the RAM block status of a permanent RAM block or the status of the explicit synchronization of a NVRAM block.
NvM_WriteBlock	NvM.h	Service to copy the data of the RAM block to its corresponding NV block.

⌋

Note: Based on implementation strategy FiM_DemInit can be used (refer also to chapter chapter 7.11.3).

Note: Based on implementation strategy either NvM_[Read|Write]Block or NvM_SetRamBlockStatus can be omitted, or the NvM usage is deactivated by configuration completely (refer also to chapter chapter 7.11.5).

8.5.3 Callout interfaces

No content.

8.5.4 Configurable interfaces

In this chapter, all interfaces are listed where the target function could be configured. The target function is usually a callback function. The names of this kind of interfaces are not fixed because they are configurable.

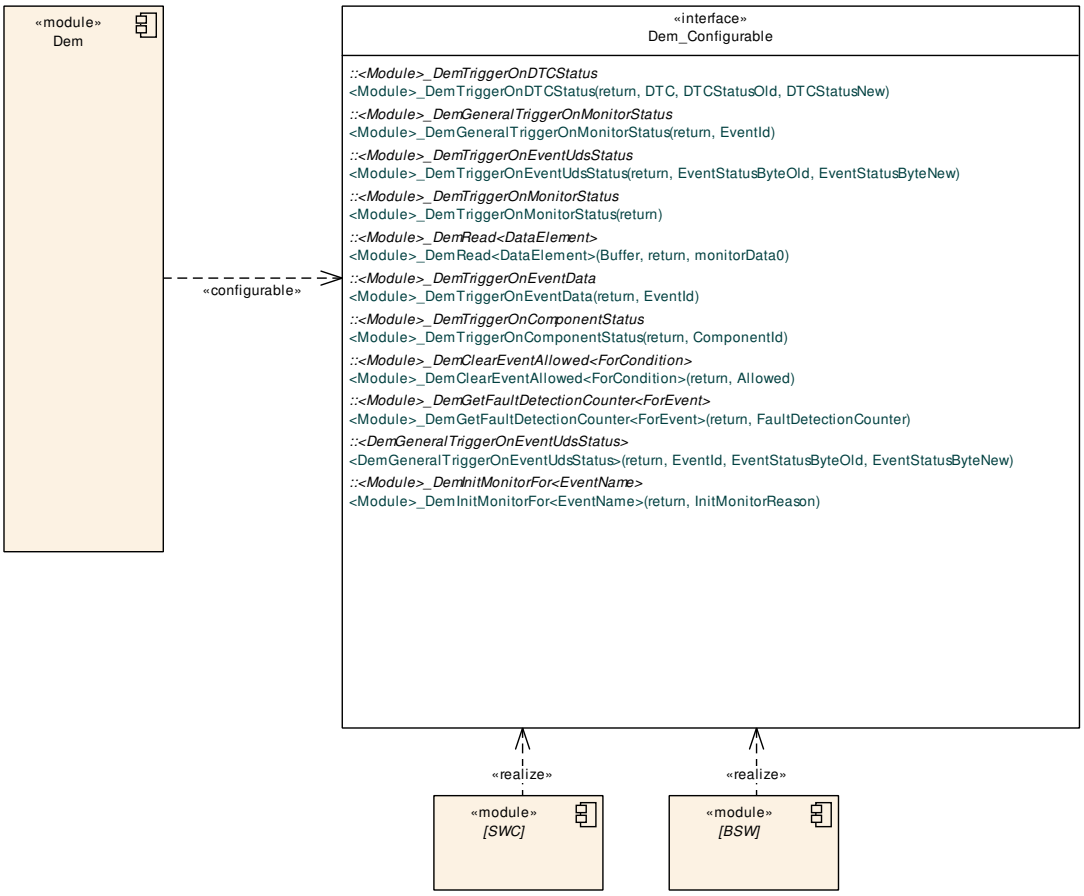


Figure 8.4: Configuration interfaces of the Dem module

8.5.4.1 Interface BSW modules / SW-Components <=> Dem

The callback interface from Dem to SW-Components is realized via RTE port interfaces. The following callback descriptions address the c-callbacks of other BSW modules.

8.5.4.1.1 InitMonitorForEvent

[SWS_Dem_00256] Definition of configurable interface <Module>_DemInitMonitorFor<EventName>

Upstream requirements: SRS_BSW_00310, SRS_BSW_00101, RS_Diag_04270

⌈

Service Name	<Module>_DemInitMonitorFor<EventName>	
Syntax	`Std_ReturnType <Module>_DemInitMonitorFor<EventName> ( Dem_InitMonitorReasonType InitMonitorReason )`	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	InitMonitorReason	Specific (re-)initialization reason evaluated from the monitor to identify the initialization kind to be performed.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	Return value unused - only for compatibility with according RTE operation.
Description	Inits the diagnostic monitor of a specific event. There is one separate callback per event (if configured), if no port interface is provided by the Dem.	
Available via	Dem_Externals.h	

⌋

8.5.4.1.2 DemTriggerOnComponentStatus

[SWS_Dem_01116] Definition of configurable interface <Module>_DemTriggerOnComponentStatus ⌈

Service Name	<Module>_DemTriggerOnComponentStatus	
Syntax	`Std_ReturnType <Module>_DemTriggerOnComponentStatus ( Dem_ComponentIdType ComponentId )`	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	ComponentId	Identification of a DemComponent.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	Return value unused - only for compatibility with according RTE operation.
Description	Triggers on changes of the DemComponent failed status.	
Available via	Dem_Externals.h	

⌋

8.5.4.2 ClearDtcNotification

[SWS_Dem_91002] Definition of configurable interface <Module>_ClearDtcNotification ⌈

Service Name	<Module>_ClearDtcNotification	
Syntax	`Std_ReturnType <Module>_ClearDtcNotification ( uint32 DTC, Dem_DTCFormatType DTCFormat, Dem_DTCOriginType DTCOrigin )`	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	DTC	DTC or group of DTC that is cleared.
	DTCFormat	Format of the DTC value.
	DTCOrigin	Event memory which is selected by the current clear operation.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK` is always returned
Description	Called by the Dem when performing a clear DTC operation.	
Available via	Dem_Externals.h	

⌋

8.5.4.3 DemGeneralTriggerOnMonitorStatus

[SWS_Dem_91009] Definition of configurable interface <Module>_DemGeneralTriggerOnMonitorStatus ⌈

Service Name	<Module>_DemGeneralTriggerOnMonitorStatus	
Syntax	`Std_ReturnType <Module>_DemGeneralTriggerOnMonitorStatus ( Dem_EventIdType EventId )`	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: this function always returns `E_OK` for compatibility reasons with RTE
Description	Triggers on changes of the monitor status. Called synchronously in context of event status reporting.	
Available via	Dem_Externals.h	

⌋

8.5.4.4 DemGeneralTriggerOnEventUdsStatus

[SWS_Dem_00259] Definition of configurable interface <DemGeneralTriggerOnEventUdsStatus>

Upstream requirements: RS_Diag_04148

⌈

Service Name	<DemGeneralTriggerOnEventUdsStatus>	
Syntax	`Std_ReturnType <DemGeneralTriggerOnEventUdsStatus> ( Dem_EventIdType EventId, Dem_UdsStatusByteType EventStatusByteOld, Dem_UdsStatusByteType EventStatusByteNew )`	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
	EventStatusByteOld	UDS DTC status byte of event before change (refer to chapter "Status bit support").
	EventStatusByteNew	UDS DTC status byte of event after change (refer to chapter "Status bit support").
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: this function always returns `E_OK` for compatibility reasons with RTE.
Description	Triggers on changes of the UDS DTC status byte.	
Available via	Dem_Externals.h	

⌋

8.5.4.5 DemTriggerOnEventUdsStatus

[SWS_Dem_91006] Definition of configurable interface <Module>_DemTriggerOnEventUdsStatus ⌈

Service Name	<Module>_DemTriggerOnEventUdsStatus	
Syntax	`Std_ReturnType <Module>_DemTriggerOnEventUdsStatus ( Dem_UdsStatusByteType EventStatusByteOld, Dem_UdsStatusByteType EventStatusByteNew )`	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventStatusByteOld	UDS DTC status byte of event before change (refer to chapter "Status bit support").
	EventStatusByteNew	UDS DTC status byte of event after change (refer to chapter "Status bit support").
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: this function always returns `E_OK` for compatibility reasons with RTE

▽

△

Description	Triggers on changes of the UDS DTC status byte.
Available via	Dem_Externals.h

⌋

8.5.4.6 DemTriggerOnDTCStatus

[SWS_Dem_00260] Definition of configurable interface <Module>_DemTriggerOnDTCStatus ⌈

Service Name	<Module>_DemTriggerOnDTCStatus	

Syntax	`Std_ReturnType <Module>_DemTriggerOnDTCStatus (`
	`  uint32 DTC,`
	`  Dem_UdsStatusByteType DTCStatusOld,`
	`  Dem_UdsStatusByteType DTCStatusNew`
	`)`	

Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	DTC	Diagnostic Trouble Code in UDS format.
	DTCStatusOld	UDS status before change
	DTCStatusNew	UDS status after change
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: this function always returns `E_OK` for compatibility reasons with RTE.
Description	Triggers on changes of the UDS status byte.	
Available via	Dem_Externals.h	

⌋

8.5.4.7 DemTriggerOnMonitorStatus

[SWS_Dem_91010] Definition of configurable interface <Module>_DemTriggerOnMonitorStatus ⌈

Service Name	<Module>_DemTriggerOnMonitorStatus

Syntax	`Std_ReturnType <Module>_DemTriggerOnMonitorStatus (`
	`  void`
	`)`

Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None

▽

△

Return value	Std_ReturnType	`E_OK`: this function always returns `E_OK` for compatibility reasons with RTE
Description	Triggers on changes of the monitor status. Called synchronously in context of event status reporting.	
Available via	Dem_Externals.h	

⌋

8.5.4.8 EventDataChanged

[SWS_Dem_00562] Definition of configurable interface <Module>_DemTriggerOnEventData

Upstream requirements: RS_Diag_04160

⌈

Service Name	<Module>_DemTriggerOnEventData	
Syntax	`Std_ReturnType <Module>_DemTriggerOnEventData ( Dem_EventIdType EventId )`	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	EventId	Identification of an event by assigned EventId.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: this function always returns `E_OK` for compatibility reasons with RTE.
Description	Triggers on changes of the event related data in the event memory.	
Available via	Dem_Externals.h	

⌋

8.5.4.9 ClearEventAllowed

[SWS_Dem_00563] Definition of configurable interface <Module>_DemClearEventAllowed<ForCondition> ⌈

Service Name	<Module>_DemClearEventAllowed<ForCondition>	
Syntax	`Std_ReturnType <Module>_DemClearEventAllowed<ForCondition> ( boolean* Allowed )`	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	Allowed	True - clearance of event is allowed False - clearance of event is not allowed

▽

△

Return value	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation failed
Description	Triggers on DTC-deletion, which is not allowed if the out-parameter returns False. There is one separate callback per condition, which can be assigned to one or several events, if no port interface is provided by the Dem. Parameter "Allowed" will be unchanged in case E_NOT_OK is returned.	
Available via	Dem_Externals.h	

⌋

8.5.4.10 ReadDataElement

If DemDataElementFncSignature is set to DATA_FNC_NORMAL, the following definition is used:

[SWS_Dem_00564] Definition of configurable interface <Module>_DemRead<DataElement> ⌈

Service Name	<Module>_DemRead<DataElement>	

Syntax	`Std_ReturnType <Module>_DemRead<DataElement> (`
	`  uint8* Buffer,`
	`  [Dem_MonitorDataType monitorData0]`
	`)`	

Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	monitorData0	MonitorData that was provided by Dem_SetEventStatusWith MonitorData
Parameters (inout)	None	
Parameters (out)	Buffer	Buffer containing the value of the data element
Return value	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation failed
Description	Requests the current value of the data element. There is one separate callback per data element, if no port interface is provided by the Dem.	
Available via	Dem_Externals.h	

⌋

Note: Square brackets [] indicate that an argument is optional.

If DemDataElementFncSignature is set to DATA_FNC_PROXY, the following definition is used:

[SWS_Dem_91060] Definition of configurable interface <Module>_DemRead<DataElement>

Upstream requirements: RS_Diag_04254

⌈

Service Name	<Module>_DemRead<DataElement>	
Syntax	`Std_ReturnType <Module>_DemRead<DataElement> ( uint8* Data, uint16 DataLength, Dcm_NegativeResponseCodeType* ErrorCode )`	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	DataLength	Length in byte of the data to be read
Parameters (inout)	None	
Parameters (out)	Data	Buffer where the requested data shall be copied to
	ErrorCode	If the operation Xxx_ReadData returns value E_NOT_OK, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value. This parameter exists to ensure 1:1 interface compatibility for Dcm and Dem SwCluC proxy interfaces enabling a common data access. The returned error code has no meaning for the Dem and shall be discarded.
Return value	Std_ReturnType	`E_OK`: Request was successful. `E_NOT_OK`: Request was not successful.
Description	Requests the current value of the data element. There is one separate callback per data element, if no port interface is provided by the Dem.	
Available via	Dem_Externals.h	

⌋

8.5.4.11 GetFaultDetectionCounter

[SWS_Dem_00263] Definition of configurable interface <Module>_DemGetFaultDetectionCounter<ForEvent> ⌈

Service Name	<Module>_DemGetFaultDetectionCounter<ForEvent>	
Syntax	`Std_ReturnType <Module>_DemGetFaultDetectionCounter<ForEvent> ( sint8* FaultDetectionCounter )`	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
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 E_OK this parameter does not contain valid data. -128dec...127dec PASSED...FAILED according to ISO 14229-1
Return value	Std_ReturnType	`E_OK`: request was successful `E_NOT_OK`: request failed

▽

△

Description	Gets the current fault detection counter value. There is one c-callback per event using monitor-internal debouncing, if no port interface is provided by the Dem.
Available via	Dem_Externals.h

⌋

8.6 Service Interfaces

8.6.1 Implementation Data Types

8.6.1.1 Dem_DataType

[SWS_Dem_91026] Definition of ImplementationDataType Dem_DataType_{Data}
⌈

Name	Dem_DataType_{Data}	
Kind	Type	
Derived from	***Basetype***	***Variation***
	Dem_DataArrayType_{Data}	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == ([S|U]INT[8|16|32|64]|FLOAT|DOUBLE)_N) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == ([S|U]INT[8|16|32|64]|FLOAT|DOUBLE)_N))}
	Dem_DataPrimitiveType_{Data}	{(ecu((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == (BOOLEAN|[S|U]INT[8|16|32|64]|FLOAT|DOUBLE) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == (BOOLEAN|[S|U]INT[8|16|32|64]|FLOAT|DOUBLE)))))}
Description	Data type definition for external data elements	
Variation	((({ecuc(Dem/DemGeneral/DemDataElementClass)} instanceof {ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass)}) && ({ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementUsePort)} == true)) || ({ecuc(Dem/DemGeneral/DemDataElementClass)} instanceof {ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass)})) Data = {ecuc(Dem/DemGeneral/DemDataElementClass.SHORT-NAME)}	
Available via	Rte_Dem_Type.h	

⌋

8.6.1.2 Dem_DataPrimitiveType

[SWS_Dem_91033] Definition of ImplementationDataType Dem_DataPrimitiveType_{Data} ⌈

Name	Dem_DataPrimitiveType_{Data}	
Kind	Type	
Derived from	***Basetype***	***Variation***
	boolean	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == BOOLEAN) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == BOOLEAN)}
	float32	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == FLOAT) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == FLOAT)}
	float64	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == DOUBLE) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == DOUBLE)}
	sint16	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == SINT16) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == SINT16)}
	sint32	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == SINT32) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == SINT32)}
	sint64	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == SINT64) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == SINT64)}
	sint8	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == SINT8) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == SINT8)}
	uint16	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == UINT16) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == UINT16)}
	uint32	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == UINT32) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == UINT32)}
	uint64	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == UINT64) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == UINT64)}
	uint8	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == UINT8) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == UINT8)}
Description	–	

▽

△

Variation	{(ecu((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == (BOOLEAN|[S|U]INT[8|16|32|64]|FLOAT|DOUBLE) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == (BOOLEAN|[S|U]INT[8|16|32|64]|FLOAT|DOUBLE))))} Data = {ecuc(Dem/DemGeneral/DemDataElementClass.SHORT-NAME)}
Available via	Rte_Dem_Type.h

⌋

8.6.1.3 Dem_DataArrayType

[SWS_Dem_91034] Definition of ImplementationDataType Dem_DataArrayType_{Data} ⌈

Name	Dem_DataArrayType_{Data}	
Kind	Array	
Element type	***Type***	***Variation***
	float32	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == FLOAT_N) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == FLOAT_N))}
	float64	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == DOUBLE_N) || ((Dem/Dem General/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == DOUBLE_N))}
	sint16	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == SINT16_N) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == SINT16_N))}
	sint32	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == SINT32_N) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == SINT32_N))}
	sint64	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == SINT64_N) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == SINT64_N))}
	sint8	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == SINT8_N) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == SINT8_N))}
	uint16	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == UINT16_N) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == UINT16_N))}
	uint32	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == UINT32_N) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == UINT32_N))}
	uint64	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == UINT64_N) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == UINT64_N))}

▽

△

	uint8	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == UINT8_N) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == UINT8_N))}
Size	–	
Description	Array data type definition for external data elements	
Variation	{(ecuc((Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataType) == ([S|U]INT[8|16|32|64]|FLOAT|DOUBLE)_N) || ((Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType) == ([S|U]INT[8|16|32|64]|FLOAT|DOUBLE)_N))} Data = {ecuc(Dem/DemGeneral/DemDataElementClass.SHORT-NAME)}	
Available via	Rte_Dem_Type.h	

⌋

8.6.1.4 Dem_DTCOriginType

[SWS_Dem_00934] Definition of ImplementationDataType Dem_DTCOriginType
⌈

Name	Dem_DTCOriginType		
Kind	Type		
Derived from	uint16		
Range	DEM_DTC_ORIGIN_PRIMARY_MEMORY	0x0001	Event information located in the primary memory
	DEM_DTC_ORIGIN_PERMANENT_MEMORY	0x0003	The Event information is located in the permanent memory
	DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY	0x0004	OBD event information located in the primary memory.
	DEM_DTC_ORIGIN_USERDEFINED_MEMORY_<Name>	0x01XX	Event information located in the user defined memory, where XX is the configured DemUserDefinedMemoryIdentifier in hexadecimal and <Name> is the Short-Name of the DemUserDefinedMemory.
Description	This enum is used to define the location of the events. The definition and use of the different memory types is OEM-specific.		
Variation	–		
Available via	Rte_Dem_Type.h		

⌋

8.6.1.5 Dem_DebouncingStateType

[SWS_Dem_01000] Definition of ImplementationDataType Dem_DebouncingStateType

Upstream requirements: RS_Diag_04105

⌈

Name	Dem_DebouncingStateType			
Kind	Bitfield			
Derived from	uint8			
Elements	**Kind**	**Name**	**Mask**	**Description**
	bit	DEM_TEMPORARILY_DEFECTIVE	0x01	Bit 0: Temporarily Defective (corresponds to 0 < FDC < 127)
	bit	DEM_FINALLY_DEFECTIVE	0x02	Bit 1: finally Defective (corresponds to FDC = 127)
	bit	DEM_TEMPORARILY_HEALED	0x04	Bit 2: temporarily healed (corresponds to -128 < FDC < 0)
	bit	DEM_TEST_COMPLETE	0x08	Bit 3: Test complete (corresponds to FDC = -128 or FDC = 127)
	bit	DEM_DTR_UPDATE	0x10	Bit 4: DTR Update (= Test complete && Debouncing complete && enable conditions / storage conditions fulfilled)
Description	–			
Variation	–			
Available via	Rte_Dem_Type.h			

⌋

8.6.1.6 Dem_DebounceResetStatusType

[SWS_Dem_00927] Definition of ImplementationDataType Dem_DebounceResetStatusType ⌈

Name	Dem_DebounceResetStatusType		
Kind	Type		
Derived from	uint8		
Range	DEM_DEBOUNCE_STATUS_FREEZE	0x00	Freeze the internal debounce timer.
	DEM_DEBOUNCE_STATUS_RESET	0x01	Reset the internal debounce counter/timer.
		0x02 - 0xFF	reserved
Description	This type contains all definitions to control an internal debounce counter/timer via the function Dem_ResetEventDebounceStatus().		
Variation	–		
Available via	Rte_Dem_Type.h		

⌋

8.6.1.7 Dem_DTRControlType

[SWS_Dem_00941] Definition of ImplementationDataType Dem_DTRControlType ⌈

Name	Dem_DTRControlType		
Kind	Type		
Derived from	uint8		
Range	DEM_DTR_CTL_NORMAL	0x00	Values are reported and regarded as valid test result
	DEM_DTR_CTL_NO_MAX	0x01	Values are reported, but maximum limit is not available (not valid); upper limit value is ignored.
	DEM_DTR_CTL_NO_MIN	0x02	Values are reported, but minimum limit is not available (not valid); lower limit value is ignored.
	DEM_DTR_CTL_RESET	0x03	Values are all ignored. External representation will be all zeros as initialized (e.g. after fault clear)
	DEM_DTR_CTL_INVISIBLE	0x04	Values are all ignored. This DTR is treated for the external view (tester) as if not integrated.
Description	Control parameter for the interpretation of the reported test results.		
Variation	–		
Available via	Rte_Dem_Type.h		

⌋

8.6.1.8 Dem_EventIdType

[SWS_Dem_00925] Definition of ImplementationDataType Dem_EventIdType ⌈

Name	Dem_EventIdType		
Kind	Type		
Derived from	uint16		
Range	1..65535	–	Internal identifier of a diagnostic event Remark: 0 is not a valid value
Description	Identification of an event by assigned EventId. The EventId is assigned by the Dem. Example: 1 refers to monitor x, 2 refers to monitor y, etc.		
Variation	–		
Available via	Rte_Dem_Type.h		

⌋

8.6.1.9 Dem_EventStatusType

[SWS_Dem_00926] Definition of ImplementationDataType Dem_EventStatusType

⌈

Name	Dem_EventStatusType		
Kind	Type		
Derived from	uint8		
Range	DEM_EVENT_STATUS_PASSED	0x00	Monitor reports qualified test result passed.
	DEM_EVENT_STATUS_FAILED	0x01	Monitor reports qualified test result failed.
	DEM_EVENT_STATUS_PREPASSED	0x02	Monitor reports non-qualified test result pre-passed (debounced Dem-internally).
	DEM_EVENT_STATUS_PREFAILED	0x03	Monitor reports non-qualified test result pre-failed (debounced Dem-internally).
	DEM_EVENT_STATUS_FDC_THRESHOLD_REACHED	0x04	Monitor triggers the storage of ExtendedDataRecords and FreezeFrames ON_FDC_THRESHOLD.
		0x05 - 0xFF	reserved
Description	This type contains all monitor test result values, which can be reported via Dem_SetEventStatus().		
Variation	–		
Available via	Rte_Dem_Type.h		

⌋

8.6.1.10 Dem_DTCFormatType

[SWS_Dem_00933] Definition of ImplementationDataType Dem_DTCFormatType

Upstream requirements: RS_Diag_04112

⌈

Name	Dem_DTCFormatType		
Kind	Type		
Derived from	uint8		
Range	DEM_DTC_FORMAT_OBD	0	selects the 2-byte OBD DTC format (refer to configuration parameter DemObdDTC)
	DEM_DTC_FORMAT_UDS	1	selects the 3-byte UDS DTC format (refer to configuration parameter DemUdsDTC)
	DEM_DTC_FORMAT_J1939	2	selects the merged SPN + FMI to 3-byte J1939 DTC format (refer to DemJ1939DTC)
	DEM_DTC_FORMAT_OBD_3BYTE	3	Selects the 3-byte OBD DTC defined by DemDtcValue3Byte in case that SAE J2012 and UDS DTC format separation is used.

▽

△

Description	This type is used to select the format of the DTC value.
Variation	–
Available via	Rte_Dem_Type.h

⌋

8.6.1.11 Dem_InitMonitorReasonType

[SWS_Dem_00942] Definition of ImplementationDataType Dem_InitMonitorReasonType

Upstream requirements: RS_Diag_04151

⌈

Name	Dem_InitMonitorReasonType		
Kind	Type		
Derived from	uint8		
Range	DEM_INIT_MONITOR_CLEAR	0x01	Event was cleared and all internal values and states are reset.
	DEM_INIT_MONITOR_RESTART	0x02	Operation cycle of the event was restarted.
	DEM_INIT_MONITOR_REENABLED	0x03	Enable conditions or DTC settings re-enabled.
	DEM_INIT_MONITOR_STORAGE_REENABLED	0x04	Storage condition reenabled.
Description	(Re-)Initialization reason returned by the callback <Module>_DemInitMonitorFor<EventName>().		
Variation	–		
Available via	Rte_Dem_Type.h		

⌋

8.6.1.12 Dem_IumprDenomCondIdType

[SWS_Dem_00943] Definition of ImplementationDataType Dem_IumprDenomCondIdType ⌈

Name	Dem_IumprDenomCondIdType
Kind	Type
Derived from	uint8

▽

△

Range	DEM_IUMPR_GENERAL_INDIVIDUAL_DENOMINATOR	0x05	Individual denominators to support different conditions than the general denominator. It acts on individual denominators and allows a different condition to be set than for the general denominator. If the standard individual denominator conditions differ from the general denominator conditions, they typically differ by a "fueled engine" criterion.
	DEM_IUMPR_GENERAL_OBDCOND	0x06	IUMPR denominator condition "General Denominator" for output with Infotype $08/$0B
	DEM_IUMPR_DEN_COND_COLDSTART_CMPL	0x07	Complete IUMPR denominator condition "Cold Start"
	DEM_IUMPR_DEN_COND_EVAP_CMPL	0x08	Complete IUMPR denominator condition "EVAP"
	DEM_IUMPR_DEN_COND_500MI_CMPL	0x09	Complete IUMPR denominator condition "500 miles"
	DEM_IUMPR_DEN_COND_CSERS	0x10	Complete IUMPR denominator condition for "Cold start emission reduction strategy" monitors
	DEM_IUMPR_DEN_COND_EVAPPURGEFLOW_CMPL	0x11	Complete IUMPR denominator condition "EVAP purge flow"
Description	This type contains all possible additional IUMPR denominator conditions to be broadcasted among OBD-relevant ECUs.		
Variation	–		
Available via	Rte_Dem_Type.h		

⌋

8.6.1.13 Dem_IumprDenomCondStatusType

[SWS_Dem_00944] Definition of ImplementationDataType Dem_IumprDenomCondStatusType ⌈

Name	Dem_IumprDenomCondStatusType		
Kind	Type		
Derived from	uint8		
Range	DEM_IUMPR_DEN_STATUS_NOT_REACHED	0x00	Condition of IUMPR-Denominator given by IUMPRDenCondId is not met (yet).
	DEM_IUMPR_DEN_STATUS_REACHED	0x01	Condition of IUMPR-Denominator given by IUMPRDenCondId is met
	DEM_IUMPR_DEN_STATUS_INHIBITED	0x02	Condition of IUMPR-Denominator given by IUMPRDenCondId is inhibited and cannot be reached.
		0x03 - 0xFF	reserved

▽

△

Description	This type contains all possible states of an additional IUMPR denominator condition to be broadcasted among OBD-relevant ECUs.
Variation	–
Available via	Rte_Dem_Type.h

⌋

8.6.1.14 Dem_MaxDataValueType

[SWS_Dem_01072] Definition of ImplementationDataType Dem_MaxDataValue Type ⌈

Name	Dem_MaxDataValueType		
Kind	Array	***Element type***	uint8
Size	size of largest Extended data class / Freeze frame record Elements		
Description	–		
Variation	–		
Available via	Rte_Dem_Type.h		

⌋

8.6.1.15 Dem_MonitorDataType

[SWS_Dem_91036] Definition of ImplementationDataType Dem_MonitorData Type ⌈

Name	Dem_MonitorDataType
Kind	Type
Derived from	uint32
Description	This type is used to pass monitoring data to the Dem.
Variation	–
Available via	Rte_Dem_Type.h

⌋

8.6.1.16 Dem_MonitorStatusType

[SWS_Dem_91005] Definition of ImplementationDataType Dem_MonitorStatus Type ⌈

Name	Dem_MonitorStatusType
Kind	Bitfield
Derived from	uint8

▽

△

Elements	Kind	Name	Mask	Description
	bit	DEM_MONITOR_STATUS_TF	0x01	Bit0: TestFailed
	bit	DEM_MONITOR_STATUS_TNCTOC	0x02	Bit1: TestNotCompletedThisOperationCycle
Description	This type contains possible monitor status values.			
Variation	–			
Available via	Rte_Dem_Type.h			

⌋

8.6.1.17 Dem_IndicatorStatusType

[SWS_Dem_00930] Definition of ImplementationDataType Dem_IndicatorStatusType ⌈

Name	Dem_IndicatorStatusType		
Kind	Type		
Derived from	uint8		
Range	DEM_INDICATOR_OFF	0x00	Indicator off mode
	DEM_INDICATOR_CONTINUOUS	0x01	Indicator continuously on mode
	DEM_INDICATOR_BLINKING	0x02	Indicator blinking mode
	DEM_INDICATOR_BLINK_CONT	0x03	Indicator blinking or continuously on mode
	DEM_INDICATOR_SLOW_FLASH	0x04	Indicator slow flashing mode
	DEM_INDICATOR_FAST_FLASH	0x05	Indicator fast flashing mode
	DEM_INDICATOR_ON_DEMAND	0x06	Indicator on-demand mode
	DEM_INDICATOR_SHORT	0x07	Indicator short mode
Description	Indicator mode used by Dem_GetIndicatorStatus().		
Variation	–		
Available via	Rte_Dem_Type.h		

⌋

8.6.1.18 Dem_PID21valueType

[SWS_Dem_01073] Definition of ImplementationDataType Dem_PID21valueType ⌈

Name	Dem_PID21valueType		
Kind	Array	**Element type**	uint8
Size	2 Elements		

▽

△

Description	–
Variation	–
Available via	Rte_Dem_Type.h

⌋

8.6.1.19 Dem_PID31valueType

[SWS_Dem_01074] Definition of ImplementationDataType Dem_PID31valueType ⌈

Name	Dem_PID31valueType		
Kind	Array	***Element type***	uint8
Size	2 Elements		
Description	–		
Variation	–		
Available via	Rte_Dem_Type.h		

⌋

8.6.1.20 Dem_RatioIdType

[SWS_Dem_00940] Definition of ImplementationDataType Dem_RatioIdType ⌈

Name	Dem_RatioIdType		
Kind	Type		
Derived from	***Basetype***	***Variation***	
	uint16	Configurable, size depends on system complexity (refer to range of configuration parameter DemRatioId)	
	uint8	Configurable, size depends on system complexity (refer to range of configuration parameter DemRatioId)	
Range	0..255, 0..65535	–	Configurable, size depends on system complexity (refer to range of configuration parameter Dem RatioId)
Description	OBD specific ratio Id (related to a specific event, a FID, and an IUMPR group). This type depends on the Dem configuration.		
Variation	–		
Available via	Rte_Dem_Type.h		

⌋

8.6.1.21 Dem_UdsStatusByteType

[SWS_Dem_00928] Definition of ImplementationDataType Dem_UdsStatusByteType ⌈

Name	Dem_UdsStatusByteType			
Kind	Bitfield			
Derived from	uint8			
Lower limit	0x00			
Upper limit	0xFF			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	DEM_UDS_STATUS_TF	0x01	bit 0: TestFailed
	bit	DEM_UDS_STATUS_TFTOC	0x02	bit 1: TestFailedThisOperation Cycle
	bit	DEM_UDS_STATUS_PDTC	0x04	bit 2: PendingDTC
	bit	DEM_UDS_STATUS_CDTC	0x08	bit 3: ConfirmedDTC
	bit	DEM_UDS_STATUS_TNCSLC	0x10	bit 4: TestNotCompletedSince LastClear
	bit	DEM_UDS_STATUS_TFSLC	0x20	bit 5: TestFailedSinceLastClear
	bit	DEM_UDS_STATUS_TNCTOC	0x40	bit 6: TestNotCompletedThis OperationCycle
	bit	DEM_UDS_STATUS_WIR	0x80	bit 7: WarningIndicator Requested
Description	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.			
Variation	–			
Available via	Rte_Dem_Type.h			

⌋

8.6.1.22 Dem_PID4DvalueType

[SWS_Dem_91028] Definition of ImplementationDataType Dem_PID4DvalueType ⌈

Name	Dem_PID4DvalueType		
Kind	Array	***Element type***	uint8
Size	2 Elements		
Description	–		
Variation	({ecuc(Dem/DemGeneral.DemOBDSupport)} == DEM_OBD_MASTER_ECU)		
Available via	Rte_Dem_Type.h		

⌋

8.6.1.23 Dem_PID4EvalueType

[SWS_Dem_91029] Definition of ImplementationDataType Dem_PID4EvalueType ⌈

Name	Dem_PID4EvalueType		
Kind	Array	***Element type***	uint8
Size	2 Elements		
Description	–		
Variation	({ecuc(Dem/DemGeneral.DemOBDSupport)} == DEM_OBD_MASTER_ECU)		
Available via	Rte_Dem_Type.h		

⌋

8.6.2 Sender-Receiver-Interfaces

8.6.2.1 DataServices_{Data}

[SWS_Dem_00850] Definition of SenderReceiverInterface DataServices_{Data} ⌈

Name	DataServices_{Data}	
Comment	–	
IsService	false	
Variation	({ecuc(Dem/DemGeneral/DemDataElementClass)} instanceof {ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass)}) Data = {ecuc(Dem/DemGeneral/DemDataElementClass.SHORT-NAME)}	
Data Elements	data	
	Type	Dem_DataType_{Data}
	Variation	Data = {ecuc(Dem/DemGeneral/DemDataElementClass.SHORT-NAME)}

⌋

8.6.3 Client-Server-Interfaces

8.6.3.1 CallbackClearEventAllowed

[SWS_Dem_00620] Definition of ClientServerInterface CallbackClearEventAllowed

Upstream requirements: RS_Diag_04117

⌈

Name	CallbackClearEventAllowed
Comment	If configured, it gets the permission to clear a specific event from the SW-C. For each event, there can be one port of this interface type.
IsService	true

▽

△

Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	ClearEventAllowed	
Comment	–	
Relates to	<Module>_DemClearEventAllowed<ForCondition>	
Variation	–	
Parameters	Allowed	
	Type	boolean
	Direction	OUT
	Comment	True - clearance of event is allowed False - clearance of event is not allowed
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.2 CallbackComponentStatusChanged

[SWS_Dem_01195] Definition of ClientServerInterface CallbackComponentStatusChanged ⌈

Name	CallbackComponentStatusChanged		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	–	–	–

Operation	ComponentStatusChanged	
Comment	–	
Relates to	–	
Variation	–	
Parameters	ComponentFailedStatus	
	Type	boolean
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	–	

⌋

8.6.3.3 CallbackDTCStatusChange

[SWS_Dem_00617] Definition of ClientServerInterface CallbackDTCStatusChange ⌈

Name	CallbackDTCStatusChange		
Comment	If configured it triggers SW-Cs on DTC status byte changes via DemCallbackDTCStatusChanged, DemCallbackOBDDTCStatusChanged, and / or DemCallbackJ1939DTCStatusChanged. There can be several ports of this interface type, provided globally by the Dem Service Component.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	DTCStatusChanged	
Comment	–	
Relates to	<Module>_DemTriggerOnDTCStatus	
Variation	–	
Parameters	DTC	
	Type	uint32
	Direction	IN
	Comment	–
	Variation	–
	DTCStatusOld	
	Type	Dem_UdsStatusByteType
	Direction	IN
	Comment	–
	Variation	–
	DTCStatusNew	
	Type	Dem_UdsStatusByteType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.4 CallbackEventDataChanged

[SWS_Dem_00618] Definition of ClientServerInterface CallbackEventDataChanged

Upstream requirements: RS_Diag_04155

⌈

Name	CallbackEventDataChanged		
Comment	If configured it triggers SW-Cs on event related data changes. For each event, there can be one port of this interface type.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful

Operation	EventDataChanged
Comment	–
Relates to	–
Variation	–
Possible Errors	–

⌋

8.6.3.5 CallbackEventUdsStatusChanged

[SWS_Dem_00615] Definition of ClientServerInterface CallbackEventUdsStatus Changed ⌈

Name	CallbackEventUdsStatusChanged		
Comment	If configured it triggers SW-Cs on event status byte changes. For each event, there can be several ports of this interface type.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful

Operation	CallbackEventUdsStatusChanged	
Comment	–	
Relates to	–	
Variation	–	
Parameters	EventStatusByteOld	
	Type	Dem_UdsStatusByteType
	Direction	IN
	Comment	–
	Variation	–
	EventStatusByteNew	
	Type	Dem_UdsStatusByteType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	–	

⌋

8.6.3.6 CallbackGetFaultDetectCounter

[SWS_Dem_00622] Definition of ClientServerInterface CallbackGetFaultDetect Counter ⌈

Name	CallbackGetFaultDetectCounter		
Comment	If configured it get the monitor-internal fault detection counter value of a specific event from the SW-C.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	GetFaultDetectionCounter	
Comment	–	
Relates to	<Module>_DemGetFaultDetectionCounter<ForEvent>	
Variation	–	
Parameters	FaultDetectionCounter	
	Type	sint8
	Direction	OUT
	Comment	Value of FaultDetectionCounter
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.7 CallbackInitMonitorForEvent

[SWS_Dem_00613] Definition of ClientServerInterface CallbackInitMonitorFor Event

Upstream requirements: SRS_BSW_00457

⌈

Name	CallbackInitMonitorForEvent		
Comment	If configure it triggers an event-specific initialization of the monitor part of the SW-C). For each event, there can be one port of this interface type.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	InitMonitorForEvent
Comment	–
Relates to	<Module>_DemInitMonitorFor<EventName>
Variation	–

▽

△

Parameters	InitMonitorReason	
	Type	Dem_InitMonitorReasonType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.8 CallbackMonitorStatusChange

[SWS_Dem_91011] Definition of ClientServerInterface CallbackMonitorStatusChange ⌈

Name	CallbackMonitorStatusChange		
Comment	If configured it triggers SW-Cs on monitor status changes. For each event, there can be several ports of this interface type.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful

Operation	MonitorStatusChanged
Comment	–
Relates to	–
Variation	–
Possible Errors	–

⌋

8.6.3.9 ClearDtcNotification

[SWS_Dem_91003] Definition of ClientServerInterface ClearDtcNotification ⌈

Name	ClearDtcNotification		
Comment	–		
IsService	true		
Variation	(count({ecuc(Dem/DemGeneral/DemEventMemorySet/DemClearDTCNotification/DemClearDtcNotificationFnc)} == NULL) > 0)		
Possible Errors	0	E_OK	Operation successful

Operation	ClearDtcNotification
Comment	–
Relates to	–

▽

△

Variation	–	
Parameters	DTC	
	Type	uint32
	Direction	IN
	Comment	–
	Variation	–
	DTCFormat	
	Type	Dem_DTCFormatType
	Direction	IN
	Comment	–
	Variation	–
	DTCOrigin	
	Type	Dem_DTCOriginType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK	

⌋

8.6.3.10 ClearDTC

[SWS_Dem_00666] Definition of ClientServerInterface ClearDTC ⌈

Name	ClearDTC		
Comment	Provides a SWC interface to trigger a ClearDTC operation within the Dem. One port of this interface type is provided globally by the Dem Service Component. It has ClientId as a port-defined argument.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	4	DEM_PENDING	Clearing the DTCs is currently in progress. The caller shall call this function again at a later moment.
	5	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.
	6	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).
	7	DEM_CLEAR_FAILED	DTC clearing failed
	8	DEM_WRONG_DTC	Selected DTC value in selected format does not exist or clearing is restricted by configuration to group of all DTCs only.
	9	DEM_WRONG_ DTCORIGIN	Selected DTCOrigin does not exist.

▽

△

	22	DEM_BUSY	A different Dem_SelectDTC dependent operation according to [SWS_Dem_01253] of this client is currently in progress.

Operation	ClearDTC
Comment	–
Relates to	Dem_ClearDTC
Variation	–
Possible Errors	E_OK E_NOT_OK DEM_PENDING DEM_CLEAR_BUSY DEM_CLEAR_MEMORY_ERROR DEM_CLEAR_FAILED DEM_WRONG_DTC DEM_WRONG_DTCORIGIN DEM_BUSY

Operation	SelectDTC	
Comment	–	
Relates to	Dem_SelectDTC	
Variation	–	
Parameters	DTC	
	Type	uint32
	Direction	IN
	Comment	–
	Variation	–
	DTCFormat	
	Type	Dem_DTCFormatType
	Direction	IN
	Comment	–
	Variation	–
	DTCOrigin	
	Type	Dem_DTCOriginType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK DEM_BUSY	

⌋

8.6.3.11 DTCSuppression

[SWS_Dem_00608] Definition of ClientServerInterface DTCSuppression

Upstream requirements: RS_Diag_04126

⌈

Name	DTCSuppression		
Comment	Provides the capability to control the suppression of DTCs. One port of this interface type is provided globally by the Dem Service Component.		
IsService	true		
Variation	({ecuc(Dem/DemGeneral/DemSuppressionSupport)} == DEM_DTC_SUPPRESSION)		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	4	DEM_PENDING	Clearing the DTCs is currently in progress. The caller shall call this function again at a later moment.
	8	DEM_WRONG_DTC	Selected DTC value in selected format does not exist or clearing is restricted by configuration to group of all DTCs only.
	9	DEM_WRONG_ DTCORIGIN	Selected DTCOrigin does not exist.

Operation	GetDTCSuppression	
Comment	–	
Relates to	Dem_GetDTCSuppression	
Variation	–	
Parameters	ClientID	
	Type	uint8
	Direction	IN
	Comment	Unique client id, assigned to the instance of the calling module.
	Variation	–
	SuppressionStatus	
	Type	boolean
	Direction	OUT
	Comment	Defines whether the respective DTC is suppressed (TRUE) or enabled (FALSE).
	Variation	–
Possible Errors	E_OK E_NOT_OK DEM_PENDING DEM_WRONG_DTC DEM_WRONG_DTCORIGIN	

Operation	SetDTCSuppression	
Comment	–	
Relates to	Dem_SetDTCSuppression	
Variation	–	
Parameters	ClientID	
	Type	uint8
	Direction	IN
	Comment	Unique client id, assigned to the instance of the calling module.
	Variation	–
	SuppressionStatus	
	Type	boolean
	Direction	IN
	Comment	–
	Variation	–

▽

△

Possible Errors	E_OK E_NOT_OK DEM_WRONG_DTC DEM_WRONG_DTCORIGIN

⌋

8.6.3.12 DataServices_{Data}

[SWS_Dem_00621] Definition of ClientServerInterface DataServices_{Data} ⌈

Name	DataServices_{Data}		
Comment	If configured it gets the data element value contained in a DID, a PID, or an extended data record from the respective SW-C via client/server or sender/receiver communication (refer to Figure "Dem and Dcm module requests PID data elements of SW-C via ReadData operation"). For each data element, one port of this interface type is provided by the SW-Cs.		
IsService	true		
Variation	(({ecuc(Dem/DemGeneral/DemDataElementClass)} instanceof {ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass)}) && ({ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementUsePort)} == true)) Data = {ecuc(Dem/DemGeneral/DemDataElementClass.SHORT-NAME)}		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	ReadData	
Comment	The server is not allowed to return E_NOT_OK, but shall always provide a valid data value (e.g. a default/replacement value in an error-case) to Dcm/Dem nevertheless the signature of the operation includes E_NOT_OK to ensure compatibility between server runnable and RTE Call API, since the RTE may return negative Std_Return values in certain cases (e.g. partition of server stopped)	
Relates to	<Module>_DemRead<DataElement>	
Variation	–	
Parameters	Data	
	Type	Dem_DataType_{Data}
	Direction	OUT
	Comment	–
	Variation	Data = {ecuc(Dem/DemGeneral/DemDataElementClass.SHORT-NAME)}
	monitorData0	
	Type	Dem_MonitorDataType
	Direction	IN
	Comment	–
	Variation	({ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementProvideMonitorData)} == true))
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.13 DTRCentralReport

[SWS_Dem_00769] Definition of ClientServerInterface DTRCentralReport ⌈

Name	DTRCentralReport		
Comment	Available if OBD support is configured.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SetDTR	
Comment	–	
Relates to	Dem_SetDTR	
Variation	–	
Parameters	TestResult	
	Type	sint32
	Direction	IN
	Comment	–
	Variation	–
	LowerLimit	
	Type	sint32
	Direction	IN
	Comment	–
	Variation	–
	UpperLimit	
	Type	sint32
	Direction	IN
	Comment	–
	Variation	–
	CtrlVal	
	Type	Dem_DTRControlType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

Note: One port of this interface type is provided per DtrId by the Dem Service Component. It has the DTRId as a port-defined argument.

8.6.3.14 DiagnosticInfo

[SWS_Dem_00599] Definition of ClientServerInterface DiagnosticInfo ⌈

Name	DiagnosticInfo		
Comment	Provides the capability to obtain the event information. One port of this interface type is provided per diagnostic event by the Dem Service Component. It has EventId as a port-defined argument.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	10	DEM_E_NO_DTC_AVAILABLE	there is no DTC configured in the requested format
	14	DEM_E_NO_FDC_AVAILABLE	there is no fault detection counter available for the requested event
	21	DEM_BUFFER_TOO_SMALL	The provided buffer size is too small
	48	DEM_NO_SUCH_ELEMENT	The requested event data is not currently stored (but the request was valid) OR The requested record number is not supported by the event OR The requested DID is not supported by the freeze frame (GetEventFreezeFrameDataEx)

Operation	GetDTCOfEvent	
Comment	–	
Relates to	Dem_GetDTCOfEvent	
Variation	–	
Parameters	DTCFormat	
	Type	Dem_DTCFormatType
	Direction	IN
	Comment	–
	Variation	–
	DTCOfEvent	
	Type	uint32
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DEM_E_NO_DTC_AVAILABLE	

Operation	GetDebouncingOfEvent	
Comment	–	
Relates to	Dem_GetDebouncingOfEvent	
Variation	(({ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass)} instanceof {ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass/DemDebounceCounterBased)}) || ({ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass)} instanceof {ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass/DemDebounceTimeBased)}))	
Parameters	DebouncingState	
	Type	Dem_DebouncingStateType
	Direction	OUT

▽

△

	Comment	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)
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	GetEventAvailable	
Comment	–	
Relates to	Dem_GetEventAvailable	
Variation	–	
Parameters	AvailableStatus	
	Type	boolean*
	Direction	OUT
	Comment	TRUE if the event is available. FALSE if the event is not available.
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	GetEventExtendedDataRecordEx	
Comment	–	
Relates to	Dem_GetEventExtendedDataRecordEx	
Variation	–	
Parameters	RecordNumber	
	Type	uint8
	Direction	IN
	Comment	–
	Variation	–
	DestBuffer	
	Type	Dem_MaxDataValueType
	Direction	OUT
	Comment	–
	Variation	–
	Bufsize	
	Type	uint16
	Direction	INOUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DEM_BUFFER_TOO_SMALL DEM_NO_SUCH_ELEMENT	

Operation	GetEventFreezeFrameDataEx
Comment	–
Relates to	Dem_GetEventFreezeFrameDataEx
Variation	–
Parameters	RecordNumber

▽

△

	Type	uint8
	Direction	IN
	Comment	–
	Variation	–
	DataId	
	Type	uint16
	Direction	IN
	Comment	–
	Variation	–
	DestBuffer	
	Type	Dem_MaxDataValueType
	Direction	OUT
	Comment	–
	Variation	–
	BufSize	
	Type	uint16
	Direction	INOUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DEM_BUFFER_TOO_SMALL DEM_NO_SUCH_ELEMENT	

Operation	GetEventUdsStatus	
Comment	Gets the current UDS status byte assigned to the DTC for the event	
Relates to	Dem_GetEventUdsStatus	
Variation	–	
Parameters	UDSStatusByte	
	Type	Dem_UdsStatusByteType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	GetFaultDetectionCounter	
Comment	–	
Relates to	Dem_GetFaultDetectionCounter	
Variation	–	
Parameters	FaultDetectionCounter	
	Type	sint8
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DEM_E_NO_FDC_AVAILABLE	

Operation	GetMonitorStatus	
Comment	–	
Relates to	Dem_GetMonitorStatus	
Variation	–	
Parameters	MonitorStatus	
	Type	Dem_MonitorStatusType
	Direction	OUT
	Comment	Monitor status byte of the requested event. If the return value of the function call is E_NOT_OK, this parameter does not contain valid data.
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

Note: These DiagnosticInfo ports are also available for each BSW event (so that also SW-Cs can access on).

8.6.3.15 DiagnosticMonitor

[SWS_Dem_00598] Definition of ClientServerInterface DiagnosticMonitor ⌈

Name	DiagnosticMonitor		
Comment	Provide the capability to modify the event information. One port of this interface type is provided per application-related diagnostic event by the Dem Service Component. It has Event Id as a port-defined argument.		
IsService	true		
Variation	(({Dem/DemGeneral/DemStorageCondition.DemStorageConditionReplacementEvent Ref})!=({Dem/DemConfigSet/DemEventParameter}) && ({Dem/DemConfigSet/DemEvent Parameter/DemEventReportingType} == STANDARD_REPORTING))		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	ClearPrestoredFreezeFrame
Comment	–
Relates to	Dem_ClearPrestoredFreezeFrame
Variation	{ecuc(Dem/DemGeneral/DemMaxNumberPrestoredFF)} > 0
Possible Errors	E_OK E_NOT_OK

Operation	PrestoreFreezeFrame
Comment	–
Relates to	Dem_PrestoreFreezeFrame
Variation	{ecuc(Dem/DemGeneral/DemMaxNumberPrestoredFF)} > 0
Possible Errors	E_OK E_NOT_OK

Operation	ResetEventDebounceStatus	
Comment	–	
Relates to	Dem_ResetEventDebounceStatus	
Variation	({ecuc(Dem/DemGeneral/DemDebounceCounterBasedSupport)} == true) || ({ecuc(Dem/DemGeneral/DemDebounceTimeBasedSupport)} == true)	
Parameters	DebounceResetStatus	
	Type	Dem_DebounceResetStatusType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	ResetEventStatus
Comment	–
Relates to	Dem_ResetEventStatus
Variation	–
Possible Errors	E_OK E_NOT_OK

Operation	ResetMonitorStatus
Comment	–
Relates to	Dem_ResetMonitorStatus
Variation	–
Possible Errors	E_OK E_NOT_OK

Operation	SetEventDisabled
Comment	–
Relates to	Dem_SetEventDisabled
Variation	{ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT
Possible Errors	E_OK E_NOT_OK

Operation	SetEventStatus	
Comment	–	
Relates to	Dem_SetEventStatus	
Variation	–	
Parameters	EventStatus	
	Type	Dem_EventStatusType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

Note: Each port of the DiagnosticMonitor interface is only connected to one monitor port.

Note: For internally used events (replacement event), no port will be generated.

Caveat: all operations within this client/server interface are mutual exclusive per server port (same eventId). Preempted invocations may be ignored and not processed by the Dem.

8.6.3.16 DiagnosticMonitor_MonitorData

[SWS_Dem_91038] Definition of ClientServerInterface DiagnosticMonitor_MonitorData ⌈

Name	DiagnosticMonitor_MonitorData		
Comment	Provide means to report diagnostic monitor status with monitor data.		
IsService	true		
Variation	({Dem/DemConfigSet/DemEventParameter/DemEventReportingType} == STANDARD_REPORTING_WITH_MONITOR_DATA)		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SetEventStatusWithMonitorData	
Comment	–	
Relates to	Dem_SetEventStatusWithMonitorData	
Variation	–	
Parameters	EventStatus	
	Type	Dem_EventStatusType
	Direction	IN
	Comment	–
	Variation	–
	monitorData0	
	Type	Dem_MonitorDataType
	Direction	IN
	Comment	–
	Variation	–
	monitorData1	
	Type	Dem_MonitorDataType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.17 EnableCondition

[SWS_Dem_00604] Definition of ClientServerInterface EnableCondition

Upstream requirements: RS_Diag_04192

⌈

Name	EnableCondition		
Comment	If at least one enable condition is configured it provide the capability to set an enable condition. One port of this interface type is provided per enable condition by the Dem Service Component. It has EnableConditionId as a port-defined argument.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SetEnableCondition	
Comment	–	
Relates to	Dem_SetEnableCondition	
Variation	–	
Parameters	ConditionFulfilled	
	Type	boolean
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.18 EventAvailable

[SWS_Dem_01193] Definition of ClientServerInterface EventAvailable ⌈

Name	EventAvailable		
Comment	–		
IsService	true		
Variation	{ecuc(Dem/DemGeneral/DemAvailabilitySupport)} == DEM_EVENT_AVAILABILITY		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SetEventAvailable	
Comment	–	
Relates to	Dem_SetEventAvailable	
Variation	–	
Parameters	AvailableStatus	
	Type	boolean
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.19 EventFailureCycleCounterThreshold

[SWS_Dem_91018] Definition of ClientServerInterface EventFailureCycleCounterThreshold ⌈

Name	EventFailureCycleCounterThreshold		
Comment	Provides the capability for dynamical adaptation of the failure cycle threshold. One port of this interface is provided per diagnostic event with EventId as a port-defined argument.		
IsService	true		
Variation	(count({ecuc(Dem/DemConfigSet/DemEventParameter/DemEventConfirmationThreshold CounterAdaptable)} == TRUE) > 0)		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SetEventFailureCycleCounterThreshold	
Comment	–	
Relates to	Dem_SetEventConfirmationThresholdCounter	
Variation	–	
Parameters	FailureCycleCounterThreshold	
	Type	uint8
	Direction	IN
	Comment	Failure cycle counter threshold of event to be set.
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.20 EvMemOverflowIndication

[SWS_Dem_00607] Definition of ClientServerInterface EvMemOverflowIndication

Upstream requirements: RS_Diag_04105

⌈

Name	EvMemOverflowIndication		
Comment	If the respective event memory is configured it provides the status of the event memory). One port of this interface type is provided per supported event memory by the Dem Service Component. It has DTCOrigin as a port-defined argument.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	GetEventMemoryOverflow
Comment	–
Relates to	Dem_GetEventMemoryOverflow

▽

△

Variation	–	
Parameters	OverflowIndication	
	Type	boolean
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	GetNumberOfEventMemoryEntries	
Comment	–	
Relates to	Dem_GetNumberOfEventMemoryEntries	
Variation	–	
Parameters	NumberOfEventMemoryEntries	
	Type	uint8
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.21 EventStatus

[SWS_Dem_00838] Definition of ClientServerInterface EventStatus ⌈

Name	EventStatus		
Comment	Provides the capability modify the event status. One port of this interface type is provided per application-related diagnostic event by the Dem Service Component. It has EventId as a port-defined argument.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SetWIRStatus	
Comment	–	
Relates to	Dem_SetWIRStatus	
Variation	–	
Parameters	WIRStatus	
	Type	boolean
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

Note: Each port of the EventStatus interface is only connected to one port of the failsafe SW-C.

8.6.3.22 GeneralCallbackEventDataChanged

[SWS_Dem_00619] Definition of ClientServerInterface GeneralCallbackEvent DataChanged

Upstream requirements: RS_Diag_04155

⌈

Name	GeneralCallbackEventDataChanged		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful

Operation	EventDataChanged	
Comment	–	
Relates to	–	
Variation	–	
Parameters	EventId	
	Type	Dem_EventIdType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	–	

⌋

8.6.3.23 GeneralCallbackEventUdsStatusChanged

[SWS_Dem_00616] Definition of ClientServerInterface GeneralCallbackEventUds StatusChanged ⌈

Name	GeneralCallbackEventUdsStatusChanged		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful

Operation	GeneralCallbackEventUdsStatusChanged
Comment	–
Relates to	–
Variation	–

▽

△

Parameters	EventId	
	Type	Dem_EventIdType
	Direction	IN
	Comment	–
	Variation	–
	EventStatusByteOld	
	Type	Dem_UdsStatusByteType
	Direction	IN
	Comment	–
	Variation	–
	EventStatusByteNew	
	Type	Dem_UdsStatusByteType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	–	

⌋

8.6.3.24 GeneralCallbackMonitorStatusChanged

[SWS_Dem_91013] Definition of ClientServerInterface GeneralCallbackMonitorStatusChanged ⌈

Name	GeneralCallbackMonitorStatusChanged		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful

Operation	MonitorStatusChanged	
Comment	–	
Relates to	–	
Variation	–	
Parameters	EventId	
	Type	Dem_EventIdType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	–	

⌋

8.6.3.25 GeneralDiagnosticInfo

[SWS_Dem_00600] Definition of ClientServerInterface GeneralDiagnosticInfo ⌈

Name	GeneralDiagnosticInfo		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	10	DEM_E_NO_DTC_ AVAILABLE	there is no DTC configured in the requested format
	14	DEM_E_NO_FDC_ AVAILABLE	there is no fault detection counter available for the requested event
	21	DEM_BUFFER_TOO_ SMALL	The provided buffer size is too small
	48	DEM_NO_SUCH_ ELEMENT	The requested event data is not currently stored (but the request was valid) OR The requested record number is not supported by the event OR The requested DID is not supported by the freeze frame (GetEventFreezeFrameDataEx)

Operation	GetDTCOfEvent	
Comment	–	
Relates to	Dem_GetDTCOfEvent	
Variation	–	
Parameters	EventId	
	Type	Dem_EventIdType
	Direction	IN
	Comment	–
	Variation	–
	DTCFormat	
	Type	Dem_DTCFormatType
	Direction	IN
	Comment	–
	Variation	–
	DTCOfEvent	
	Type	uint32
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DEM_E_NO_DTC_AVAILABLE	

Operation	GetDebouncingOfEvent
Comment	–
Relates to	Dem_GetDebouncingOfEvent

▽

△

Variation	(({ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass)} instanceof {ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass/DemDebounceCounterBased)}) || ({ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass)} instanceof {ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass/DemDebounceTimeBased)}))	
Parameters	EventId	
	Type	Dem_EventIdType
	Direction	IN
	Comment	–
	Variation	–
	DebouncingState	
	Type	Dem_DebouncingStateType
	Direction	OUT
	Comment	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)
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	GetEventExtendedDataRecordEx	
Comment	–	
Relates to	Dem_GetEventExtendedDataRecordEx	
Variation	–	
Parameters	EventId	
	Type	Dem_EventIdType
	Direction	IN
	Comment	–
	Variation	–
	RecordNumber	
	Type	uint8
	Direction	IN
	Comment	–
	Variation	–
	DestBuffer	
	Type	Dem_MaxDataValueType
	Direction	OUT
	Comment	–
	Variation	–
	Bufsize	
	Type	uint16
	Direction	INOUT
	Comment	–
	Variation	–

▽

△

Possible Errors	E_OK E_NOT_OK DEM_BUFFER_TOO_SMALL DEM_NO_SUCH_ELEMENT

Operation	GetEventFreezeFrameDataEx	
Comment	–	
Relates to	Dem_GetEventFreezeFrameDataEx	
Variation	–	
Parameters	EventId	
	Type	Dem_EventIdType
	Direction	IN
	Comment	–
	Variation	–
	RecordNumber	
	Type	uint8
	Direction	IN
	Comment	–
	Variation	–
	DataId	
	Type	uint16
	Direction	IN
	Comment	–
	Variation	–
	DestBuffer	
	Type	Dem_MaxDataValueType
	Direction	OUT
	Comment	–
	Variation	–
	BufSize	
	Type	uint16
	Direction	INOUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DEM_BUFFER_TOO_SMALL DEM_NO_SUCH_ELEMENT	

Operation	GetEventUdsStatus	
Comment	Gets the current UDS status byte assigned to the DTC for the event	
Relates to	Dem_GetEventUdsStatus	
Variation	–	
Parameters	EventId	
	Type	Dem_EventIdType
	Direction	IN
	Comment	–
	Variation	–
	UDSStatusByte	

▽

△

	Type	Dem_UdsStatusByteType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	GetFaultDetectionCounter	
Comment	–	
Relates to	Dem_GetFaultDetectionCounter	
Variation	–	
Parameters	EventId	
	Type	Dem_EventIdType
	Direction	IN
	Comment	–
	Variation	–
	FaultDetectionCounter	
	Type	sint8
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DEM_E_NO_FDC_AVAILABLE	

Operation	GetMonitorStatus	
Comment	–	
Relates to	Dem_GetMonitorStatus	
Variation	–	
Parameters	EventID	
	Type	Dem_EventIdType
	Direction	IN
	Comment	Identification of an event by assigned EventId.
	Variation	–
	MonitorStatus	
	Type	Dem_MonitorStatusType
	Direction	OUT
	Comment	Monitor status byte of the requested event. If the return value of the function call is E_NOT_OK, this parameter does not contain valid data.
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.26 GetDataOfPID21

[SWS_Dem_01092] Definition of ClientServerInterface GetDataOfPID21 ⌈

Name	GetDataOfPID21		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	GetDataOfPID21	
Comment	–	
Relates to	Dem_GetDataOfPID21	
Variation	–	
Parameters	PID21value	
	Type	Dem_PID21valueType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.27 IndicatorStatus

[SWS_Dem_00606] Definition of ClientServerInterface IndicatorStatus ⌈

Name	IndicatorStatus		
Comment	One port of this interface type is provided per indicator by the Dem Service Component. It has IndicatorId as a port-defined argument.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	GetIndicatorStatus	
Comment	–	
Relates to	Dem_GetIndicatorStatus	
Variation	–	
Parameters	IndicatorStatus	
	Type	Dem_IndicatorStatusType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.28 IUMPRDenominator

[SWS_Dem_00611] Definition of ClientServerInterface IUMPRDenominator ⌈

Name	IUMPRDenominator		
Comment	If OBD is configured it provides the capability to define the number of times the vehicle operation has been fulfilled. One port of this interface type is provided per ratio Id by the Dem Service Component. It has RatioID as a port-defined argument.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	RepIUMPRDenRelease
Comment	–
Relates to	Dem_RepIUMPRDenRelease
Variation	–
Possible Errors	E_OK E_NOT_OK

⌋

8.6.3.29 IUMPRDenominatorCondition

[SWS_Dem_00742] Definition of ClientServerInterface IUMPRDenominatorCondition ⌈

Name	IUMPRDenominatorCondition		
Comment	If OBD is configured it broadcasts the status information of the General Denominator and additional denominator conditions among all OBD relevant ECUs. One port of this interface type is provided per denominator condition Id by the Dem Service Component. It has Dem_IumprDenomCondId as a port-defined argument.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	GetIUMPRDenCondition	
Comment	–	
Relates to	Dem_GetIUMPRDenCondition	
Variation	–	
Parameters	ConditionStatus	
	Type	Dem_IumprDenomCondStatusType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	SetIUMPRDenCondition	
Comment	–	
Relates to	Dem_SetIUMPRDenCondition	
Variation	–	
Parameters	ConditionStatus	
	Type	Dem_IumprDenomCondStatusType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.30 IUMPRNumerator

[SWS_Dem_00610] Definition of ClientServerInterface IUMPRNumerator ⌈

Name	IUMPRNumerator		
Comment	If OBD is configured it provides the capability to define the number of times a fault could have been found. One port of this interface type is provided per ratio Id by the Dem Service Component. It has RatioID as a port-defined argument.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	RepIUMPRFaultDetect
Comment	–
Relates to	Dem_RepIUMPRFaultDetect
Variation	–
Possible Errors	E_OK E_NOT_OK

⌋

8.6.3.31 OperationCycle

[SWS_Dem_00601] Definition of ClientServerInterface OperationCycle ⌈

Name	OperationCycle
Comment	Provides the capability to set the state of an operation cycle. One port of this interface type is provided per operation cycle by the Dem Service Component. It has OperationCycleId as a port-defined argument.
IsService	true

▽

△

Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	GetCycleQualified	
Comment	–	
Relates to	Dem_GetCycleQualified	
Variation	–	
Parameters	isQualified	
	Type	boolean
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	RestartOperationCycle
Comment	–
Relates to	Dem_RestartOperationCycle
Variation	–
Possible Errors	E_OK E_NOT_OK

Operation	SetCycleQualified
Comment	–
Relates to	Dem_SetCycleQualified
Variation	–
Possible Errors	E_OK E_NOT_OK

⌋

8.6.3.32 PowerTakeOff

[SWS_Dem_00612] Definition of ClientServerInterface PowerTakeOff ⌈

Name	PowerTakeOff		
Comment	Available if OBD support is configured. One port of this interface type is provided by the Dem Service Component.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SetPtoStatus	
Comment	–	
Relates to	Dem_SetPtoStatus	
Variation	–	
Parameters	PtoStatus	
	Type	boolean
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.33 SetDataOfPID21

[SWS_Dem_00745] Definition of ClientServerInterface SetDataOfPID21 ⌈

Name	SetDataOfPID21		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SetDataOfPID21	
Comment	–	
Relates to	Dem_SetDataOfPID21	
Variation	–	
Parameters	PID21value	
	Type	Dem_PID21valueType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.34 SetDataOfPID31

[SWS_Dem_00746] Definition of ClientServerInterface SetDataOfPID31 ⌈

Name	SetDataOfPID31		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SetDataOfPID31	
Comment	–	
Relates to	Dem_SetDataOfPID31	
Variation	–	
Parameters	PID31value	
	Type	Dem_PID31valueType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.3.35 SetDataOfPID4D

[SWS_Dem_91027] Definition of ClientServerInterface SetDataOfPID4D ⌈

Name	SetDataOfPID4D		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful

Operation	SetDataOfPID4D	
Comment	–	
Relates to	Dem_SetDataOfPID4D	
Variation	–	
Parameters	PID4Dvalue	
	Type	Dem_PID4DvalueType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK	

⌋

8.6.3.36 SetDataOfPID4E

[SWS_Dem_91031] Definition of ClientServerInterface SetDataOfPID4E ⌈

Name	SetDataOfPID4E		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful

Operation	SetDataOfPID4E	
Comment	–	
Relates to	Dem_SetDataOfPID4E	
Variation	–	
Parameters	PID4Evalue	
	Type	Dem_PID4EvalueType
	Direction	IN
	Comment	–
	Variation	–
Possible Errors	E_OK	

⌋

8.6.3.37 StorageCondition

[SWS_Dem_00605] Definition of ClientServerInterface StorageCondition

Upstream requirements: RS_Diag_04219

⌈

Name	StorageCondition		
Comment	Provides the capability to set a storage condition if at least one storage condition is configured. One port of this interface type is provided per storage condition by the Dem Service Component. It has StorageConditionId as a port-defined argument.		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	SetStorageCondition	
Comment	–	
Relates to	Dem_SetStorageCondition	
Variation	–	
Parameters	ConditionFulfilled	
	Type	boolean
	Direction	IN

▽

△

	Comment	This parameter specifies whether the enable condition assigned to the Enable ConditionID is fulfilled (TRUE) or not fulfilled (FALSE).
	Variation	–
Possible Errors	E_OK E_NOT_OK	

⌋

8.6.4 Ports

8.6.4.1 CBClrEvt

[SWS_Dem_91024] Definition of Port CBClrEvt_{Name} required by module Dem

⌈

Name	CBClrEvt_{Name}		
Kind	RequiredPort	***Interface***	CallbackClearEventAllowed
Description	–		
Variation	(({ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackClearEventAllowed)} != NULL) && ({ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackClearEventAllowed/DemCallback ClearEventAllowedFnc)} == NULL)) Name = {ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackClearEvent Allowed.SHORT-NAME)}		

⌋

8.6.4.2 CBDataEvt

[SWS_Dem_01003] Definition of Port CBDataEvt_{Name} required by module Dem

Upstream requirements: RS_Diag_04155

⌈

Name	CBDataEvt_{Name}		
Kind	RequiredPort	***Interface***	CallbackEventDataChanged
Description	–		
Variation	(({ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackEventDataChanged)} != NULL) && ({ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackEventDataChanged/DemCallback EventDataChangedFnc)} == NULL)) Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

⌋

8.6.4.3 CBFaultDetectCtr

[SWS_Dem_01004] Definition of Port CBFaultDetectCtr_{Name} required by module Dem

Upstream requirements: RS_Diag_04125

⌈

Name	CBFaultDetectCtr_{Name}		
Kind	RequiredPort	***Interface***	CallbackGetFaultDetectCounter
Description	–		
Variation	({ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass/DemDebounceMonitorInternal/DemCallbackGetFDC/DemCallbackGetFDCFnc)} == NULL) Name = {ecuc(Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass/DemDebounceMonitorInternal/DemCallbackGetFDC.SHORT-NAME)}		

⌋

8.6.4.4 CBInitEvt

[SWS_Dem_01005] Definition of Port CBInitEvt_{Name} required by module Dem

Upstream requirements: SRS_BSW_00457

⌈

Name	CBInitEvt_{Name}		
Kind	RequiredPort	***Interface***	CallbackInitMonitorForEvent
Description	–		
Variation	(({ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackInitMForE)} != NULL) && ({ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackInitMForE/DemCallbackInitMForEFnc)} == NULL)) Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

⌋

8.6.4.5 CBStatusDTC

[SWS_Dem_01007] Definition of Port CBStatusDTC_{Name} required by module Dem ⌈

Name	CBStatusDTC_{Name}		
Kind	RequiredPort	***Interface***	CallbackDTCStatusChange
Description	This is the generic port interface for a DTC status changed callback. The Dem supports 3 kind of DTCs: UDS DTCs, J1939 DTCs, and 2 Byte OBD DTCs. As the interface is the same for all 3 kinds of DTCs, this generic port interface addresses all the 3 types. The name of the interface is derived from the shortname of the configured callback container. So technically if a single diagnostic event is using more than one DTC, there could be more than one callback function for the same DTC, one for each type. In case the 2 byte OBD DTC callback is used, within the DTCStatusChanged operation of the C/S interface CallbackDTCStatusChange, the 2 byte DTC value is mapped to the uint32 parameter DTC.		
Variation	((({ecuc(Dem/DemGeneral/DemClient/DemCallbackDTCStatusChanged)} != NULL) && ({ecuc(Dem/DemGeneral/DemClient/DemCallbackDTCStatusChanged/DemCallbackDTCStatus ChangedFnc)} == NULL))|| (({ecuc(Dem/DemGeneral/DemGeneralJ1939)} != NULL) && ({ecuc(Dem/DemGeneral/DemGeneralJ1939/DemCallbackJ1939DTCStatusChanged)} != NULL) && ({ecuc(Dem/DemGeneral/DemGeneralJ1939/DemCallbackJ1939DTCStatusChanged/Dem CallbackDTCStatusChangedFnc)} == NULL))|| (({ecuc(Dem/DemGeneral/DemGeneralOBD)} != NULL) && ({ecuc(Dem/DemGeneral/DemGeneralOBD/DemCallbackOBDDTCStatusChanged)} != NULL) && ({ecuc(Dem/DemGeneral/DemGeneralOBD/DemCallbackOBDDTCStatusChanged/ DemCallbackDTCStatusChangedFnc)} == NULL))) Name = {ecuc(Dem/DemGeneral/DemClient/DemCallbackDTCStatusChanged.SHORT-NAME)} || {ecuc(Dem/DemGeneral/DemGeneralJ1939/DemCallbackJ1939DTCStatus Changed.SHORT-NAME)} || {ecuc(Dem/DemGeneral/DemGeneralOBD/DemCallback OBDDTCStatusChanged.SHORT-NAME)}		

⌋

8.6.4.6 CBEventUdsStatusChanged

[SWS_Dem_01008] Definition of Port CBEventUdsStatusChanged_{Event Name}_{CallbackName} required by module Dem ⌈

Name	CBEventUdsStatusChanged_{EventName}_{CallbackName}		
Kind	RequiredPort	***Interface***	CallbackEventUdsStatusChanged
Description	–		
Variation	EventName = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)} CallbackName = {ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackEventUdsStatus Changed.SHORT-NAME)}		

⌋

8.6.4.7 CBMonitorStatusChanged

[SWS_Dem_91012] Definition of Port CBMonitorStatusChanged_{EventName}_{CallbackName} required by module Dem ⌈

Name	CBMonitorStatusChanged_{EventName}_{CallbackName}		
Kind	RequiredPort	***Interface***	CallbackMonitorStatusChange
Description	–		
Variation	({ecuc(Dem/DemConfigSet/DemEventParameter.DemEventKind)} == DEM_EVENT_KIND_SWC) EventName = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)} CallbackName = {ecuc(Dem/DemConfigSet/DemEventParameter/DemCallbackMonitorStatus Changed.SHORT-NAME)}		

⌋

8.6.4.8 CBStatusComp

[SWS_Dem_01197] Definition of Port CBStatusComp_{ComponentName} required by module Dem ⌈

Name	CBStatusComp_{ComponentName}		
Kind	RequiredPort	***Interface***	CallbackComponentStatusChanged
Description	–		
Variation	({ecuc(Dem/DemConfigSet/DemComponent.DemComponentFailedCallbackUsePort)}== TRUE) ComponentName = {ecuc(Dem/DemConfigSet/DemComponent.SHORT-NAME)}		

⌋

8.6.4.9 ClearDTC

[SWS_Dem_01009] Definition of Port ClearDTC_{Client} provided by module Dem ⌈

Name	ClearDTC_{Client}		
Kind	ProvidedPort	***Interface***	ClearDTC
Description	–		
Port Defined Argument Value(s)	***Type***	uint8	
	Value	{ecuc(Dem/DemGeneral/DemClient.Short-Name)}	
Variation	({ecuc(Dem/DemGeneral/DemClient/DemClientUsesRte)}==True) && ({ecuc(Dem/DemGeneral/DemClient/DemClientFunctionality)} == DEM_CLIENT_USES_FULL_FUNCTIONALITY) Client = {ecuc(Dem/DemGeneral/DemClient.SHORT-NAME)}		

⌋

8.6.4.10 ClearDtcNotification

[SWS_Dem_01357] Definition of Port ClearDtcNotification_{EventMemorySet}_{Notification} required by module Dem ⌈

Name	ClearDtcNotification_{EventMemorySet}_{Notification}		
Kind	RequiredPort	***Interface***	ClearDtcNotification
Description	called by the Dem when performing a clear DTC operation		
Variation	({ecuc(Dem/DemGeneral/DemEventMemorySet/DemClearDTCNotification/DemClearDtc NotificationFnc)}== NULL) EventMemorySet = {ecuc(Dem/DemGeneral/DemEventMemorySet.SHORT-NAME)} Notification = {ecuc(Dem/DemGeneral/DemEventMemorySet/DemClear DTCNotification.SHORT-NAME)}		

⌋

8.6.4.11 ControlDTCSuppression

[SWS_Dem_01010] Definition of Port ControlDTCSuppression provided by module Dem

Upstream requirements: RS_Diag_04126

⌈

Name	ControlDTCSuppression		
Kind	ProvidedPort	***Interface***	DTCSuppression
Description	–		
Variation	({ecuc(Dem/DemGeneral/DemSuppressionSupport)} == DEM_DTC_SUPPRESSION)		

⌋

8.6.4.12 EventAvailable

[SWS_Dem_01011] Definition of Port EventAvailable_{Name} provided by module Dem ⌈

Name	EventAvailable_{Name}		
Kind	ProvidedPort	***Interface***	EventAvailable
Description	–		
Port Defined Argument Value(s)	***Type***	Dem_EventIdType	
	Value	{ecuc(Dem/DemConfigSet/DemEventParameter/DemEventId.value)}	
Variation	{ecuc(Dem/DemGeneral/DemAvailabilitySupport)} == DEM_EVENT_AVAILABILITY Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

⌋

8.6.4.13 ControlEventFailureCycleCounterThreshold

[SWS_Dem_91019] Definition of Port ControlEventFailureCycleCounterThreshold_{Name} provided by module Dem ⌈

Name	ControlEventFailureCycleCounterThreshold_{Name}		
Kind	ProvidedPort	***Interface***	EventFailureCycleCounterThreshold
Description	–		
Port Defined Argument Value(s)	***Type***	Dem_EventIdType	
	Value	{ecuc(Dem/DemConfigSet/DemEventParameter/DemEventId.value)}	
Variation	{ecuc(Dem/DemConfigSet/DemEventParameter/DemEventConfirmationThresholdCounter Adaptable)} == TRUE Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

⌋

8.6.4.14 DataServices_{Data}

[SWS_Dem_01012] Definition of Port DataServices_{Data} required by module Dem ⌈

Name	DataServices_{Data}		
Kind	RequiredPort	***Interface***	DataServices_{Data}, DataServices_{Data}
Description	–		
Variation	(({ecuc(Dem/DemGeneral/DemDataElementClass)} instanceof {ecuc(Dem/DemGeneral/DemData ElementClass/DemExternalCSDataElementClass)}) && ({ecuc(Dem/DemGeneral/DemData ElementClass/DemExternalCSDataElementClass/DemDataElementUsePort)} == TRUE)) || ({ecuc(Dem/DemGeneral/DemDataElementClass)} instanceof {ecuc(Dem/DemGeneral/DemData ElementClass/DemExternalSRDataElementClass)}) Data = {ecuc(Dem/DemGeneral/DemDataElementClass.SHORT-NAME)}		

⌋

8.6.4.15 DTR

[SWS_Dem_01039] Definition of Port DTR_{Name} provided by module Dem ⌈

Name	DTR_{Name}		
Kind	ProvidedPort	***Interface***	DTRCentralReport
Description	–		
Port Defined Argument Value(s)	***Type***	uint16	
	Value	{ecuc(Dem/DemConfigSet/DemDtrs/DemDtr/DemDtrId.value)}	
Variation	({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT) Name = {ecuc(Dem/DemConfigSet/DemDtrs/DemDtr.SHORT-NAME)}		

⌋

8.6.4.16 EnableCond

[SWS_Dem_01038] Definition of Port EnableCond_{Name} provided by module Dem

Upstream requirements: RS_Diag_04192

⌈

Name	EnableCond_{Name}		
Kind	ProvidedPort	***Interface***	EnableCondition
Description	–		
Port Defined Argument Value(s)	***Type***	uint8	
	Value	{ecuc(Dem/DemGeneral/DemEnableCondition/DemEnableConditionId.value)}	
Variation	Name = {ecuc(Dem/DemGeneral/DemEnableCondition.SHORT-NAME)}		

⌋

8.6.4.17 Event

[SWS_Dem_01037] Definition of Port Event_{Name} provided by module Dem

Upstream requirements: RS_Diag_04151

⌈

Name	Event_{Name}		
Kind	ProvidedPort	***Interface***	DiagnosticMonitor
Description	–		
Port Defined Argument Value(s)	***Type***	Dem_EventIdType	
	Value	{ecuc(Dem/DemConfigSet/DemEventParameter/DemEventId.value)}	
Variation	Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

⌋

8.6.4.18 DiagnosticMonitor_MonitorData

[SWS_Dem_91039] Definition of Port DiagnosticMonitor_MonitorData_{Name} provided by module Dem ⌈

Name	DiagnosticMonitor_MonitorData_{Name}		
Kind	ProvidedPort	***Interface***	DiagnosticMonitor_MonitorData
Description	–		
Port Defined Argument Value(s)	***Type***	Dem_EventIdType	
	Value	{ecuc(Dem/DemConfigSet/DemEventParameter/DemEventId.value)}	
Variation	Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

⌋

8.6.4.19 EventStatus

[SWS_Dem_01036] Definition of Port EventStatus_{Name} provided by module Dem ⌈

Name	EventStatus_{Name}		
Kind	ProvidedPort	***Interface***	EventStatus
Description	–		
Port Defined Argument Value(s)	***Type***	Dem_EventIdType	
	Value	{ecuc(Dem/DemConfigSet/DemEventParameter/DemEventId.value)}	
Variation	Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

⌋

8.6.4.20 EventInfo

[SWS_Dem_01034] Definition of Port EventInfo_{Name} provided by module Dem

Upstream requirements: RS_Diag_04151

⌈

Name	EventInfo_{Name}		
Kind	ProvidedPort	***Interface***	DiagnosticInfo
Description	–		
Port Defined Argument Value(s)	***Type***	Dem_EventIdType	
	Value	{ecuc(Dem/DemConfigSet/DemEventParameter/DemEventId.value)}	
Variation	Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

⌋

8.6.4.21 GeneralCBDataEvt

[SWS_Dem_01041] Definition of Port GeneralCBDataEvt required by module Dem
⌈

Name	GeneralCBDataEvt		
Kind	RequiredPort	***Interface***	GeneralCallbackEventDataChanged
Description	–		
Variation	({ecuc(Dem/DemGeneral/DemGeneralInterfaceSupport)} == True)		

⌋

8.6.4.22 GeneralCBMonitorStatusChanged

[SWS_Dem_91014] Definition of Port GeneralCBMonitorStatusChanged required by module Dem ⌈

Name	GeneralCBMonitorStatusChanged		
Kind	RequiredPort	***Interface***	GeneralCallbackMonitorStatusChanged
Description	–		
Variation	({ecuc(Dem/DemGeneral/DemGeneralInterfaceSupport)} == True)		

⌋

8.6.4.23 GeneralCBStatusEvt

[SWS_Dem_01032] Definition of Port GeneralCBStatusEvt required by module Dem ⌈

Name	GeneralCBStatusEvt		
Kind	RequiredPort	***Interface***	GeneralCallbackEventUdsStatusChanged
Description	–		
Variation	({ecuc(Dem/DemGeneral/DemGeneralInterfaceSupport)} == True)		

⌋

8.6.4.24 GeneralEvtInfo

[SWS_Dem_01031] Definition of Port GeneralEvtInfo provided by module Dem ⌈

Name	GeneralEvtInfo		
Kind	ProvidedPort	***Interface***	GeneralDiagnosticInfo
Description	–		
Variation	({ecuc(Dem/DemGeneral/DemGeneralInterfaceSupport)} == True)		

⌋

8.6.4.25 IndStatus

[SWS_Dem_01030] Definition of Port IndStatus_{Name} provided by module Dem ⌈

Name	IndStatus_{Name}		
Kind	ProvidedPort	***Interface***	IndicatorStatus
Description	–		

▽

△

Port Defined Argument Value(s)	***Type***	uint8
	Value	{ecuc(Dem/DemGeneral/DemEventMemorySet/DemIndicator/DemIndicatorID.value)}
Variation	Name = {ecuc(Dem/DemGeneral/DemEventMemorySet/DemIndicator.SHORT-NAME)}	

⌋

8.6.4.26 IUMPRDenominator

[SWS_Dem_01029] Definition of Port IUMPRDenominator_{Name} provided by module Dem ⌈

Name	IUMPRDenominator_{Name}		
Kind	ProvidedPort	***Interface***	IUMPRDenominator
Description	–		
Port Defined Argument Value(s)	***Type***	Dem_RatioIdType	
	Value	{ecuc(Dem/DemGeneral/DemRatio/DemRatioId.value)}	
Variation	({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT) Name = {ecuc(Dem/DemGeneral/DemRatio/DemRatioId.SHORT-NAME)}		

⌋

8.6.4.27 IUMPRDenominatorCondition

[SWS_Dem_01028] Definition of Port IUMPRDenominatorCondition_{Name} provided by module Dem ⌈

Name	IUMPRDenominatorCondition_{Name}		
Kind	ProvidedPort	***Interface***	IUMPRDenominatorCondition
Description	–		
Port Defined Argument Value(s)	***Type***	Dem_IumprDenomCondIdType	
	Value	TBD	
Variation	({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT) Name = {ecuc(Dem/DemGeneral/DemRatio/DemIUMPRDenGroup.SHORT-NAME)}		

⌋

8.6.4.28 IUMPRNumerator

[SWS_Dem_01027] Definition of Port IUMPRNumerator_{Name} provided by module Dem ⌈

Name	IUMPRNumerator_{Name}		
Kind	ProvidedPort	***Interface***	IUMPRNumerator
Description	–		
Port Defined Argument Value(s)	***Type***	Dem_RatioIdType	
	Value	{ecuc(Dem/DemGeneral/DemRatio/DemRatioId.value)}	
Variation	({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT) Name = {ecuc(Dem/DemGeneral/DemRatio/DemRatioId.SHORT-NAME)}		

⌋

8.6.4.29 OpCycle

[SWS_Dem_01026] Definition of Port OpCycle_{Name} provided by module Dem ⌈

Name	OpCycle_{Name}		
Kind	ProvidedPort	***Interface***	OperationCycle
Description	–		
Port Defined Argument Value(s)	***Type***	uint8	
	Value	{ecuc(Dem/DemGeneral/DemOperationCycle/DemOperationCycleId.value)}	
Variation	Name = {ecuc(Dem/DemGeneral/DemOperationCycle.SHORT-NAME)}		

⌋

8.6.4.30 OverflowIndPrimaryMemory

[SWS_Dem_01023] Definition of Port OverflowIndPrimaryMemory_{Client} provided by module Dem

Upstream requirements: RS_Diag_04093

⌈

Name	OverflowIndPrimaryMemory_{Client}		
Kind	ProvidedPort	***Interface***	EvMemOverflowIndication
Description	–		
Port Defined Argument Value(s)	***Type***	uint8	
	Value	–	
	Type	Dem_DTCOriginType	
	Value	0x01	

▽

△

Variation	(({ecuc(Dem/DemGeneral/DemClient/DemClientUsesRte)} ==TRUE)) Client= {ecuc(Dem/DemGeneral/DemClient.SHORT-NAME})

⌋

8.6.4.31 OverflowIndUserDefinedMemory

[SWS_Dem_01022] Definition of Port OverflowIndUserDefinedMemory_{Client}_{UserDefinedMemory} provided by module Dem ⌈

Name	OverflowIndUserDefinedMemory_{Client}_{UserDefinedMemory}		
Kind	ProvidedPort	***Interface***	EvMemOverflowIndication
Description	–		
Port Defined Argument Value(s)	***Type***	uint8	
	Value	–	
	Type	Dem_DTCOriginType	
	Value	{ecuc(Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory/DemUserDefinedMemoryIdentifier.value)}	
Variation	({ecuc(Dem/DemGeneral/DemClient/DemEventMemorySetRef->DemEventMemorySet/DemUserDefinedMemory != NULL)}) (({ecuc(Dem/DemGeneral/DemClient/DemClientUsesRte)} ==TRUE)) Client= {ecuc(Dem/DemGeneral/DemEventMemorySet.SHORT-NAME}) UserDefinedMemory = {ecuc(Dem/DemGeneral/DemClient/DemEventMemorySetRef->DemEventMemorySet/DemUserDefinedMemory.SHORT-NAME)}		

⌋

8.6.4.32 PowerTakeOffStatus

[SWS_Dem_01020] Definition of Port PowerTakeOffStatus provided by module Dem ⌈

Name	PowerTakeOffStatus		
Kind	ProvidedPort	***Interface***	PowerTakeOff
Description	–		
Variation	{ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT		

⌋

8.6.4.33 GetDataOfPID21

[SWS_Dem_01094] Definition of Port GetDataOfPID21 provided by module Dem
⌈

Name	GetDataOfPID21		
Kind	ProvidedPort	***Interface***	GetDataOfPID21
Description	–		
Variation	({ecuc(Dem/DemGeneral/DemGeneralOBD.DemOBDCentralizedPID21Handling)} == true) && ({ecuc(Dem/DemGeneral.DemOBDSupport)} == DEM_OBD_MASTER_ECU)		

⌋

8.6.4.34 SetDataOfPID21

[SWS_Dem_01017] Definition of Port SetDataOfPID21 provided by module Dem
⌈

Name	SetDataOfPID21		
Kind	ProvidedPort	***Interface***	SetDataOfPID21
Description	–		
Variation	({ecuc(Dem/DemGeneral/DemGeneralOBD.DemOBDCentralizedPID21Handling)} == true) && ({ecuc(Dem/DemGeneral.DemOBDSupport)} == DEM_OBD_PRIMARY_ECU)		

⌋

8.6.4.35 SetDataOfPID31

[SWS_Dem_01016] Definition of Port SetDataOfPID31 provided by module Dem
⌈

Name	SetDataOfPID31		
Kind	ProvidedPort	***Interface***	SetDataOfPID31
Description	–		
Variation	({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT) && ({ecuc(Dem/DemGeneral/DemGeneralOBD.DemOBDCentralizedPID31Handling)} == true)		

⌋

8.6.4.36 SetDataOfPID4D

[SWS_Dem_91030] Definition of Port SetDataOfPID4D provided by module Dem

⌈

Name	SetDataOfPID4D		
Kind	ProvidedPort	***Interface***	SetDataOfPID4D
Description	–		
Variation	({ecuc(Dem/DemGeneral.DemOBDSupport)} == DEM_OBD_MASTER_ECU)		

⌋

8.6.4.37 SetDataOfPID4E

[SWS_Dem_91032] Definition of Port SetDataOfPID4E provided by module Dem

⌈

Name	SetDataOfPID4E		
Kind	ProvidedPort	***Interface***	SetDataOfPID4E
Description	–		
Variation	({ecuc(Dem/DemGeneral.DemOBDSupport)} == DEM_OBD_MASTER_ECU)		

⌋

8.6.4.38 StorageCond

[SWS_Dem_01015] Definition of Port StorageCond_{Name} provided by module Dem

Upstream requirements: RS_Diag_04219

⌈

Name	StorageCond_{Name}		
Kind	ProvidedPort	***Interface***	StorageCondition
Description	–		
Port Defined Argument Value(s)	***Type***	uint8	
	Value	{ecuc(Dem/DemGeneral/DemStorageCondition/DemStorageConditionId.value)}	
Variation	Name = {ecuc(Dem/DemGeneral/DemStorageCondition.SHORT-NAME)}		

⌋

9 Sequence Diagrams

9.1 ControlDTCSetting

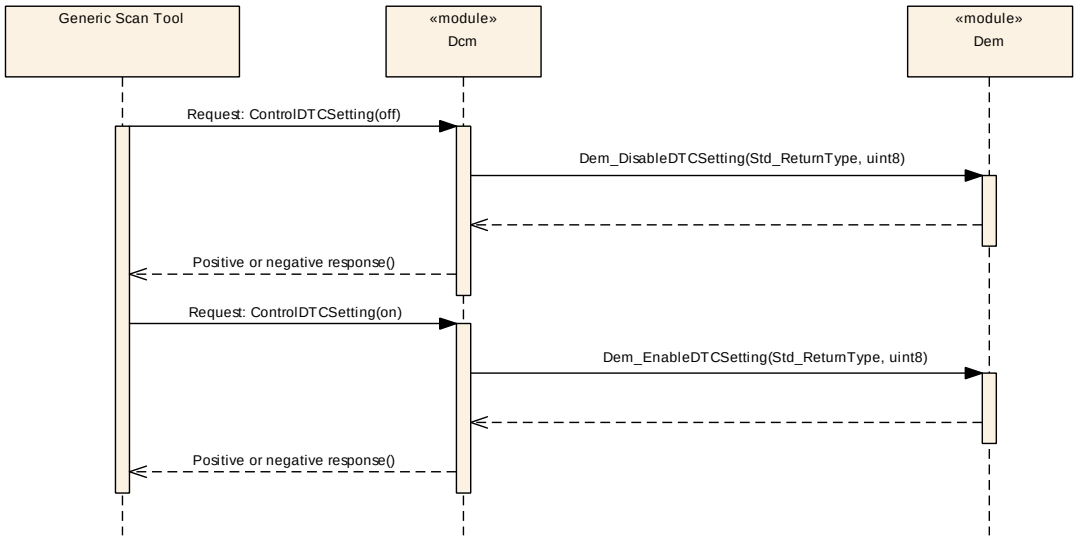


Figure 9.1: Sequence diagram of ControlDTCSetting

9.2 Dem_ClearDTC

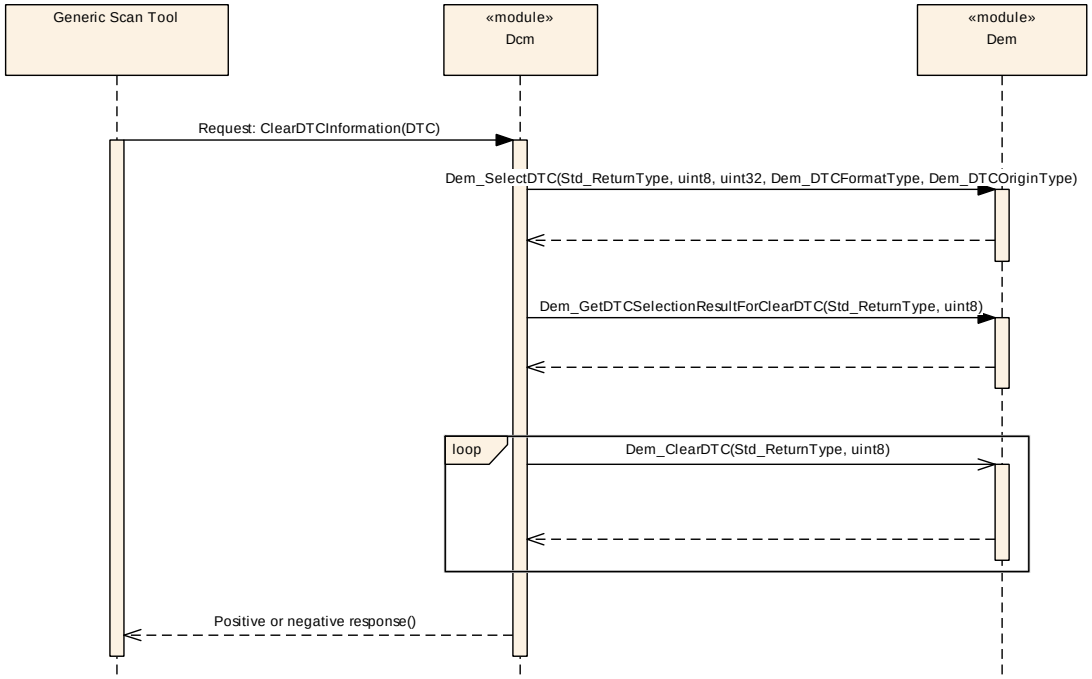


Figure 9.2: Sequence diagram of Dem_ClearDTC

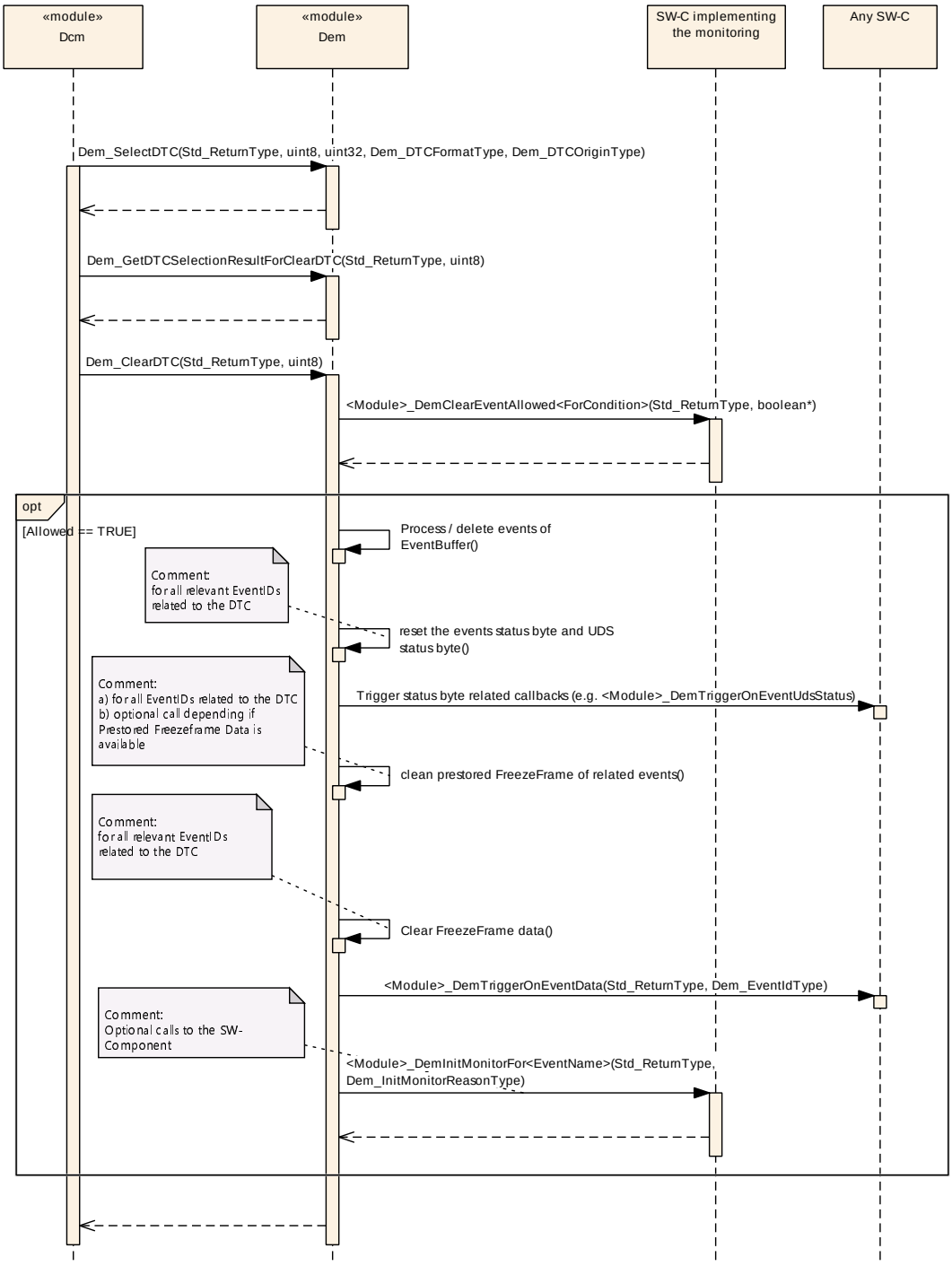


Figure 9.3: Sequence diagram of Dem_ClearDTC clearing a single DTC Dem-internally

9.3 Dem_GetDTCByOccurrenceTime

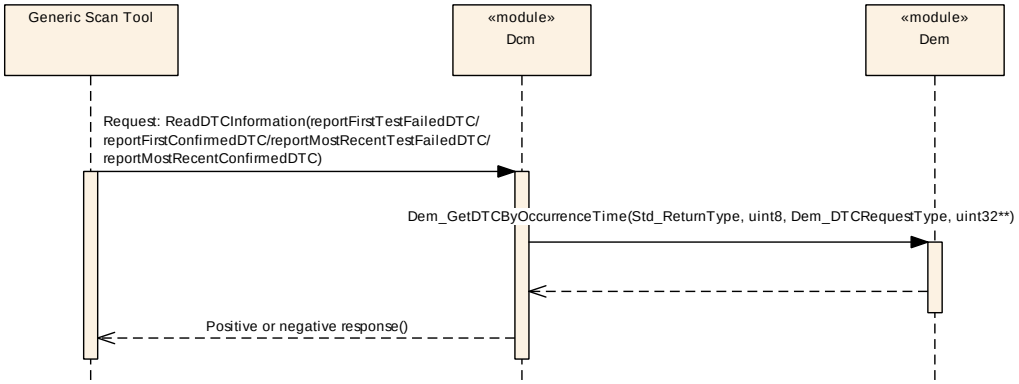


Figure 9.4: Sequence diagram of Dem_DcmGetDTCByOccurenceTime

9.4 Dem_GetNextExtendedDataRecord

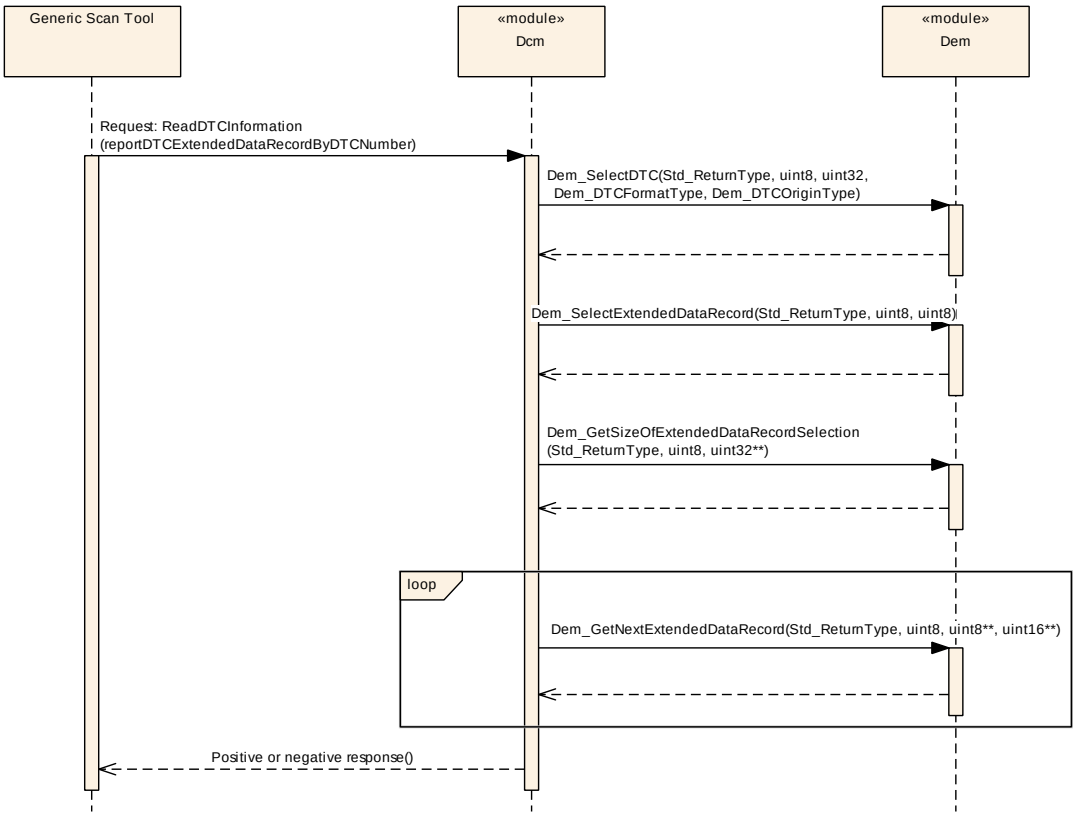


Figure 9.5: Sequence diagram of Dem_GetNextExtendedDataRecord

9.5 Dem_DcmGetStatusOfDTC

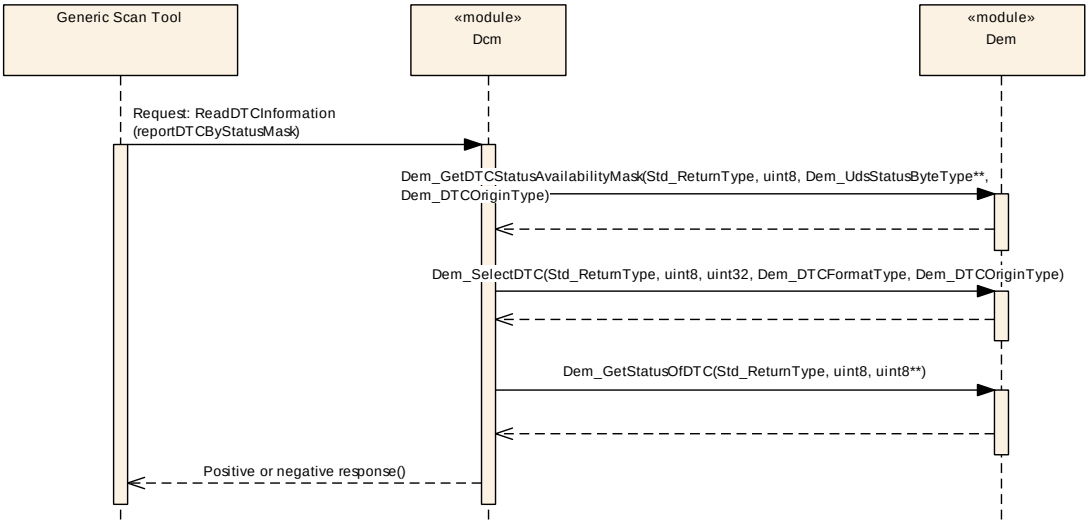


Figure 9.6: Sequence diagram of `Dem_GetStatusOfDTC`

9.6 Retrieving freeze frames

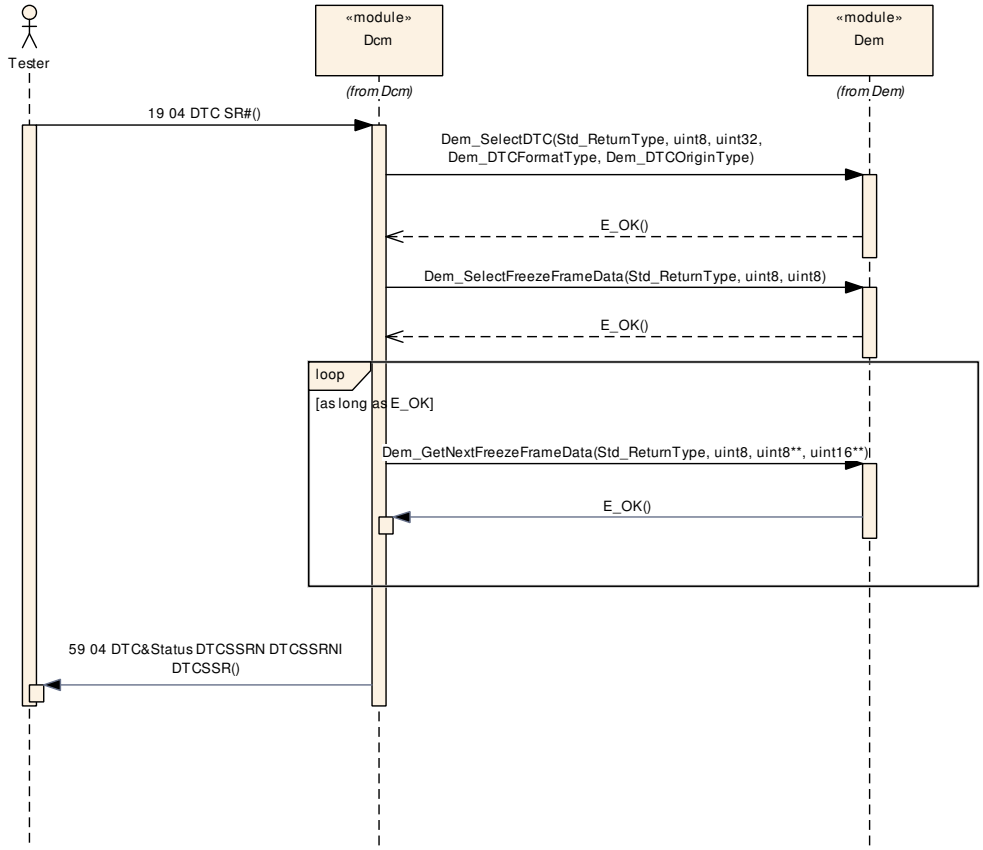


Figure 9.7: Sequence diagram to retrieve freeze frames

9.7 GetOBDFaultInformation

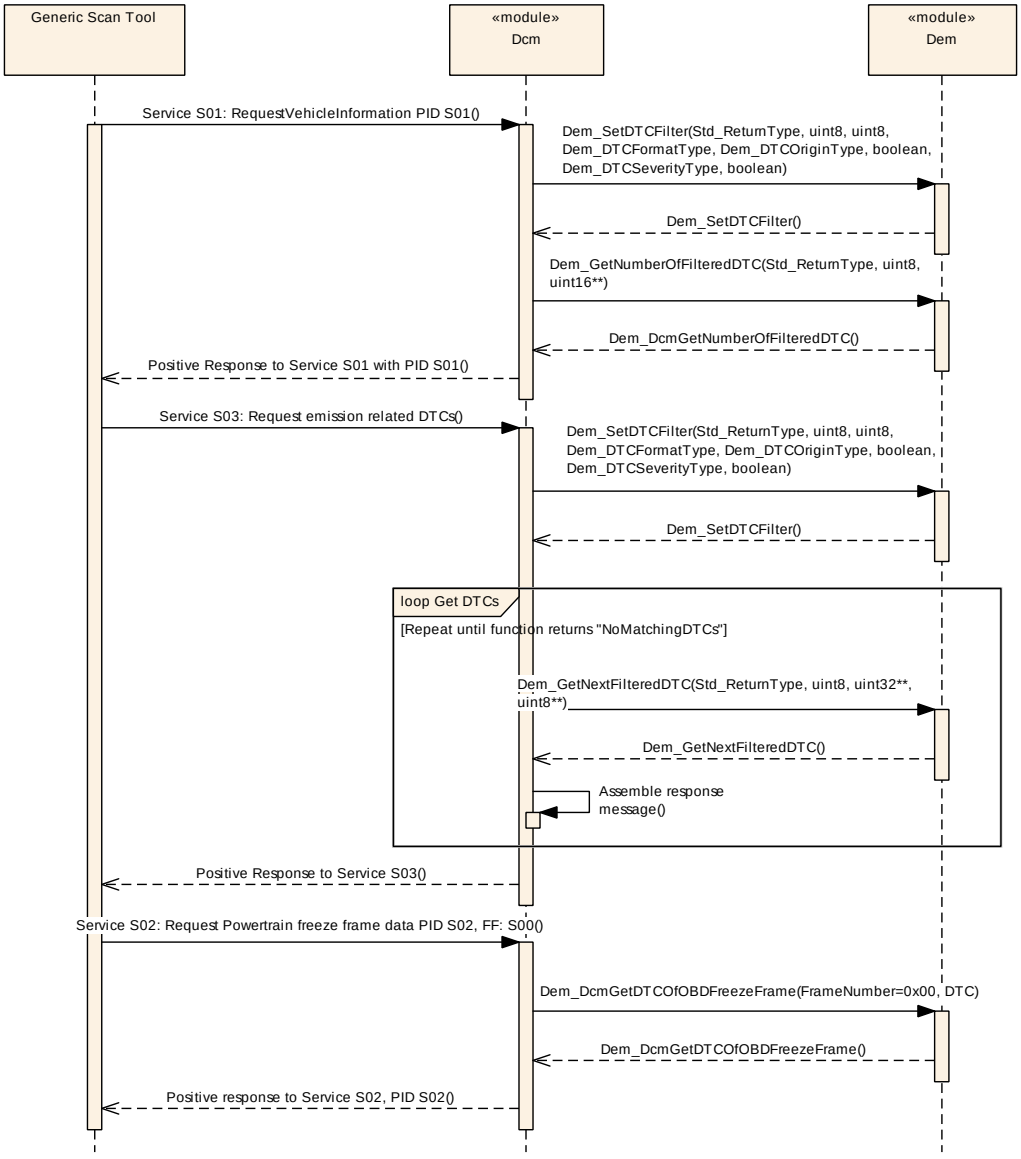


Figure 9.8: Sequence diagram of GetOBDFaultInformation

9.8 J1939DcmGetNextFilteredDTC



Figure 9.9: Sequence diagram of `Dem_J1939DcmGetNextFilteredDTC`

9.9 ReportDTCByStatusMask

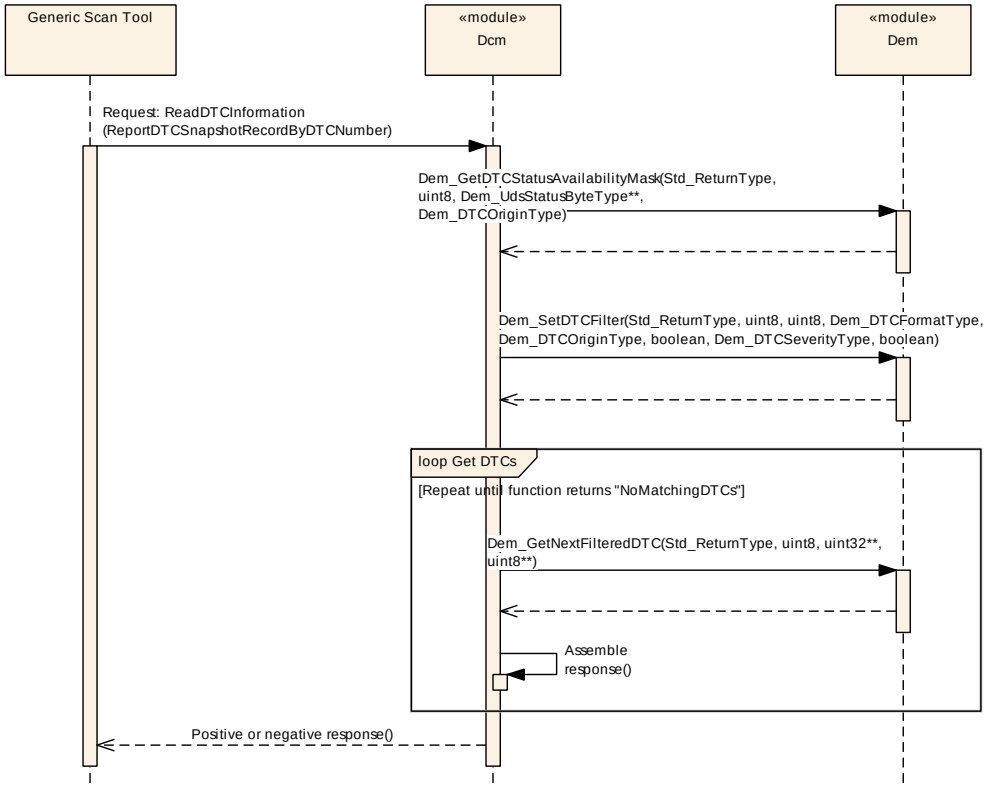


Figure 9.10: Sequence diagram of ReportDTCStatusMask

9.10 FiM_DemTriggerOnMonitorStatus

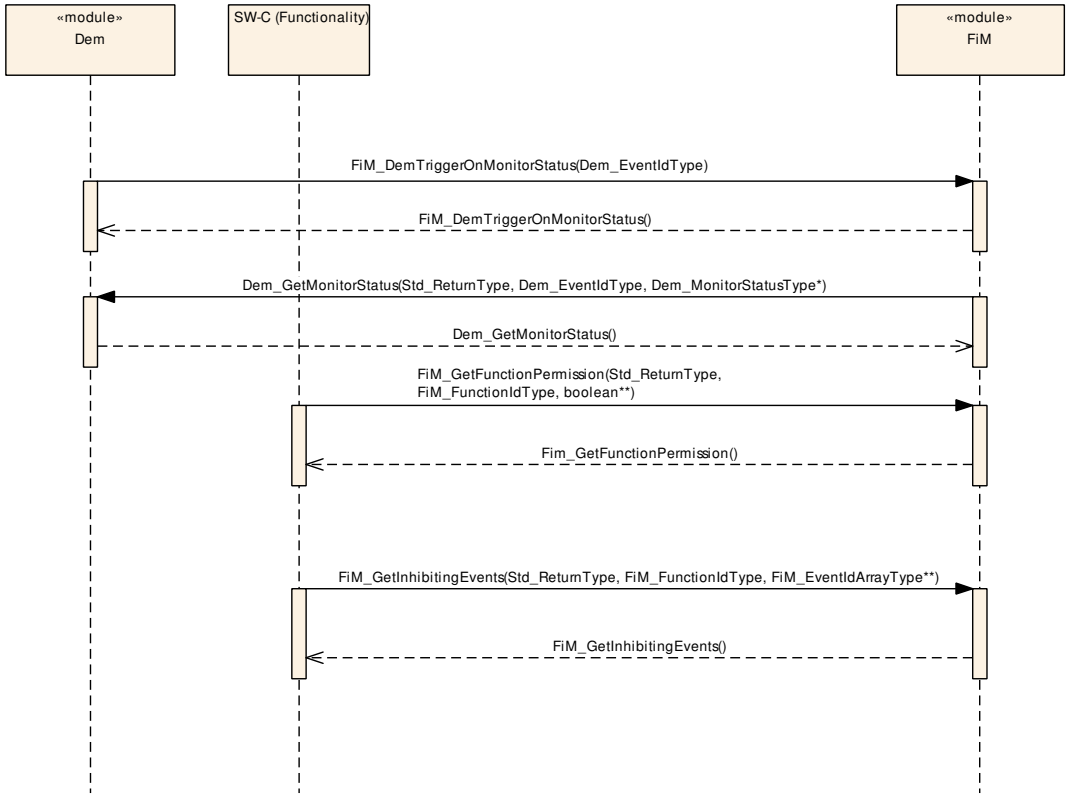


Figure 9.11: Sequence diagram of FiM_DemTriggerOnMonitorStatus

9.11 ProcessEvent (Example)

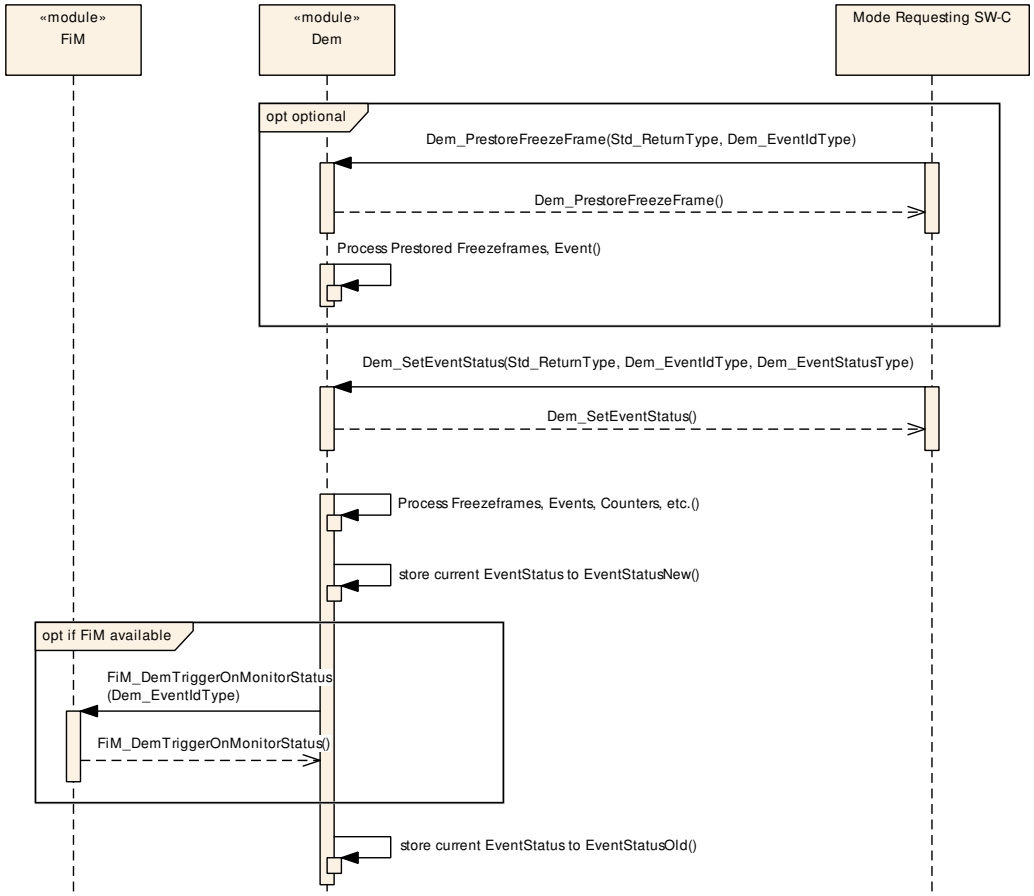


Figure 9.12: Sequence diagram for an example of ProcessEvent Dem-internally

9.12 Retrieving DTCs based on supported extended data records

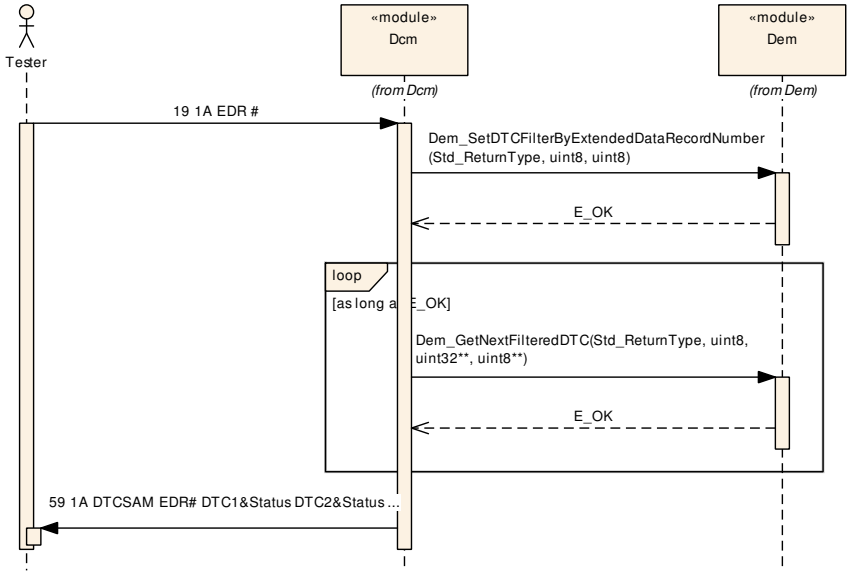


Figure 9.13: Sequence diagram of Dem_GetDTCBasedOnSupportedExtendedDataRecords

9.13 Retrieving DTCs assigned to a DTC readiness group

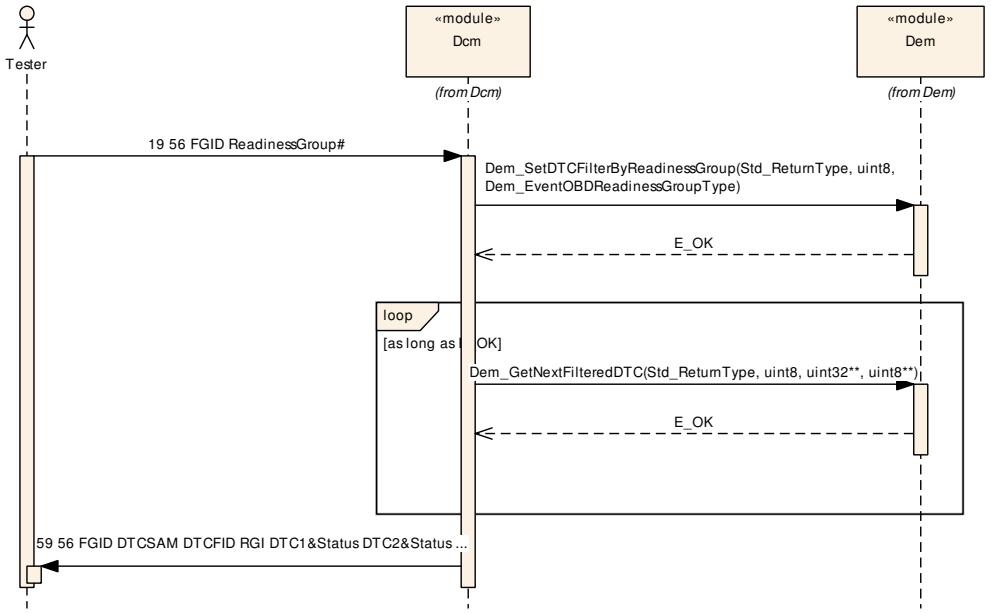


Figure 9.14: Sequence diagram of Dem_GetDTCAssignedToDTCReadinessGroup

9.14 Retrieving stored snapshot records for all DTCs

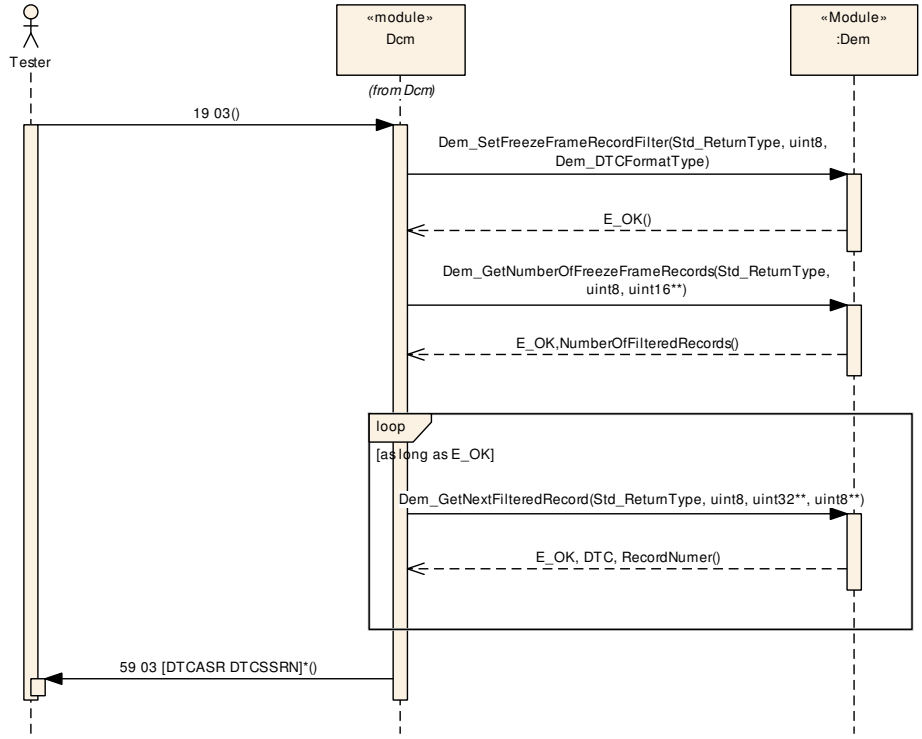


Figure 9.15: Sequence diagram of Dem_RetrievingStoredSnapshotRecordsForAllDTCs

10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers. In order to support the specification Chapter chapter 10.1 describes fundamentals. It also specifies a template (table) you shall use for the parameter specification. We intend to leave chapter 10.1 in the specification to guarantee comprehension.

- chapter 10.2 specifies the structure (containers) and the parameters of the module Dem
- chapter 10.3 specifies published information of the module Dem

10.1 How to read this chapter

For details refer to [6] Chapter 10.1 *"Introduction to configuration specification"*.

10.2 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meaning of the parameters are described in chapter 7 and chapter 8.

10.2.1 Dem

[ECUC_Dem_00928] Definition of EcucModuleDef Dem ⌈

Module Name	Dem
Description	Configuration of the Dem (Diagnostic Event Manager) module.
Post-Build Selectable Support	true
Post-Build Loadable Support	true

Included Containers		
Container Name	**Multiplicity**	**Description**
DemConfigSet	1	This container contains the configuration parameters and sub containers of the Dem module supporting multiple configuration sets.
DemGeneral	1	This container contains the configuration (parameters) of the BSW Dem

⌋

10.2.2 General

10.2.2.1 DemGeneral

[ECUC_Dem_00677] Definition of EcucParamConfContainerDef DemGeneral ⌈

Container Name	DemGeneral
Parent Container	Dem
Upstream mapping defined	–
Description	This container contains the configuration (parameters) of the BSW Dem
Multiplicity	1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemAgingRequiresNotFailedCycle	1	[ECUC_Dem_00918]
DemAgingRequiresTestedCycle	1	[ECUC_Dem_00877]
DemAvailabilitySupport	1	[ECUC_Dem_00878]
DemBswErrorBufferSize	0..1	[ECUC_Dem_00625]
DemClearDTCBehavior	1	[ECUC_Dem_00766]
DemClearDTCLimitation	1	[ECUC_Dem_00790]
DemDataElementDefaultEndianness	1	[ECUC_Dem_00858]
DemDebounceCounterBasedSupport	1	[ECUC_Dem_00724]
DemDebounceTimeBasedSupport	1	[ECUC_Dem_00725]
DemDevErrorDetect	1	[ECUC_Dem_00648]
DemEventCombinationSupport	1	[ECUC_Dem_00740]
DemGeneralCallbackEventUdsStatusChangedFnc	0..1	[ECUC_Dem_00996]
DemGeneralCallbackMonitorStatusChangedFnc	0..1	[ECUC_Dem_00938]
DemGeneralInterfaceSupport	1	[ECUC_Dem_00880]
DemMaxNumberPrestoredFF	1	[ECUC_Dem_00692]
DemOBDSupport	1	[ECUC_Dem_00698]
DemPTOSupport	1	[ECUC_Dem_00704]
DemResetConfirmedBitOnOverflow	1	[ECUC_Dem_00799]
DemResetPendingBitOnOverflow	1	[ECUC_Dem_00991]
DemStatusBitHandlingTestFailedSinceLastClear	1	[ECUC_Dem_00784]
DemSuppressionSupport	1	[ECUC_Dem_00793]
DemTaskTime	1	[ECUC_Dem_00715]
DemTriggerFiMReports	1	[ECUC_Dem_00719]
DemTriggerMonitorInitBeforeClearOk	1	[ECUC_Dem_00765]
DemClearEventsWithoutDTCEventMemoryRef	0..1	[ECUC_Dem_00941]
DemOBDEventMemorySetRef	0..1	[ECUC_Dem_00940]

Included Containers		
Container Name	**Multiplicity**	**Description**
DemClient	1..255	This container contains possible clients that are using the Dem APIs.
DemDataElementClass	0..65535	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.
DemDidClass	0..65535	This container contains the configuration (parameters) for a data Id class. It is assembled out of one or several data elements.
DemEnableCondition	0..255	This container contains the configuration (parameters) for enable conditions.
DemEnableConditionGroup	0..255	This container contains the configuration (parameters) for enable condition groups.
DemEventMemorySet	1..255	This container is a collection of referenced event memories and related information for a Dem client.
DemExtendedDataClass	0..65535	This class contains the combinations of extended data records for an extended data class.
DemExtendedDataRecordClass	0..253	This container contains the configuration (parameters) for an extended data record class. It is assembled out of one or several data elements.
DemFreezeFrameRecNumClass	0..65535	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 according freeze frame records.
DemFreezeFrameRecordClass	0..65535	This container contains a list of dedicated, different freeze frame record numbers.
DemGeneralJ1939	0..1	This container contains the general J1939-specific configuration (parameters) of the Dem module. If the container exists the J1939 support is enabled.
DemGeneralOBD	0..1	This container contains the general OBD-specific configuration (parameters) of the Dem module.
DemNvRamBlockId	0..*	This container contains the configuration (parameters) for a non-volatile memory block, which is used from the Dem. If no permanent storage of event memory entries is required, no block needs to be configured. The number of blocks which are necessary depends on the implementation and configuration (e.g. number of used event memories) of the Dem module.
DemOperationCycle	1..256	This container holds all parameters that are relevant to configure an operation cycle.
DemRatio	0..65535	This container contains the OBD-specific in-use-monitor performance ratio configuration. It is related to a specific event, a FID, and an IUMPR group.
DemStorageCondition	0..255	This container contains the configuration (parameters) for storage conditions.
DemStorageConditionGroup	0..255	This container contains the configuration (parameters) for storage condition groups.

⌋

[ECUC_Dem_00918] Definition of EcucBooleanParamDef DemAgingRequiresNotFailedCycle ⌈

Parameter Name	DemAgingRequiresNotFailedCycle
Parent Container	DemGeneral
Upstream mapping defined	–
Description	Defines if the aging cycle counter is processed in operation cycles with test failed report or not. True: Aging cycle counter is processed only in operation cycles without test failed. False (Default): No effect on aging cycle counter processing.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_Dem_00877] Definition of EcucBooleanParamDef DemAgingRequiresTestedCycle ⌈

Parameter Name	DemAgingRequiresTestedCycle
Parent Container	DemGeneral
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	Defines if the aging cycle counter is processed every aging cycles or if only tested aging cycle are considered. true: only tested aging cycle are considered for aging cycle counter false: aging cycle counter is processed every aging cycle
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00878] Definition of EcucEnumerationParamDef DemAvailabilitySupport ⌈

Parameter Name	DemAvailabilitySupport	
Parent Container	DemGeneral	
Upstream mapping defined	–	
Description	This configuration switch defines, whether support for availability is enabled or not.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_EVENT_AVAILABILITY	Support availability by Event
	DEM_NO_AVAILABILITY	Availability is not supported
Pre-Compile only	true	

⌋

[ECUC_Dem_00625] Definition of EcucIntegerParamDef DemBswErrorBufferSize ⌈

Parameter Name	DemBswErrorBufferSize	
Parent Container	DemGeneral	
Upstream mapping defined	–	
Description	Maximum number of elements in buffer for handling of BSW errors (ref. to SWS_Dem_00207).	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00766] Definition of EcucEnumerationParamDef DemClearDTCBehavior ⌈

Parameter Name	DemClearDTCBehavior	
Parent Container	DemGeneral	
Upstream mapping defined	–	
Description	Defines the clearing process of diagnostic information for volatile and non-volatile memory and the positive response handling for the Dcm module.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	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
Pre-Compile only	true	

⌋

[ECUC_Dem_00790] Definition of EcucEnumerationParamDef DemClearDTCLimitation ⌈

Parameter Name	DemClearDTCLimitation	
Parent Container	DemGeneral	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Defines the supported Dem_<...>ClearDTC API scope.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_ALL_SUPPORTED_DTCS	Dem_<...>ClearDTC accepts all supported DTC values, as well as all DTC values which are configured in DemGroupDTCs and DEM_DTC_GROUP_ALL_DTCS.
	DEM_ONLY_CLEAR_ALL_DTCS	Dem_<...>ClearDTC accepts ClearAllDTCs only.
Default value	DEM_ALL_SUPPORTED_DTCS	

▽

△

Pre-Compile only	true

⌋

[ECUC_Dem_00858] Definition of EcucEnumerationParamDef DemDataElement DefaultEndianness ⌈

Parameter Name	DemDataElementDefaultEndianness	
Parent Container	DemGeneral	
Upstream mapping defined	–	
Description	Defines the default endianness of the data belonging to a data element which is applicable if the DemExternalSRDataElementClass does not define a endianness.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	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
Pre-Compile only	true	

⌋

[ECUC_Dem_00724] Definition of EcucBooleanParamDef DemDebounceCounter BasedSupport ⌈

Parameter Name	DemDebounceCounterBasedSupport
Parent Container	DemGeneral
Upstream mapping defined	–
Description	This configuration switch defines, whether support for counter based debouncing is enabled or not. true: counter based debouncing support is enabled false: counter based debouncing support is disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00725] Definition of EcucBooleanParamDef DemDebounceTime BasedSupport ⌈

Parameter Name	DemDebounceTimeBasedSupport
Parent Container	DemGeneral
Upstream mapping defined	–
Description	This configuration switch defines, whether support for time based debouncing is enabled or not. true: time based debouncing support is enabled false: time based debouncing support is disabled

▽

△

Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00648] Definition of EcucBooleanParamDef DemDevErrorDetect ⌈

Parameter Name	DemDevErrorDetect
Parent Container	DemGeneral
Upstream mapping defined	–
Description	Switches the development error detection and notification on or off. • true: detection and notification is enabled. • false: detection and notification is disabled.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_Dem_00740] Definition of EcucEnumerationParamDef DemEventCombinationSupport ⌈

Parameter Name	DemEventCombinationSupport	
Parent Container	DemGeneral	
Upstream mapping defined	–	
Description	This parameter defines the type of event combination supported by the Dem.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_EVCOMB_DISABLED	No event combination supported
	DEM_EVCOMB_ONSTORAGE	Event combination on storage
Pre-Compile only	true	

⌋

[ECUC_Dem_00996] Definition of EcucFunctionNameDef DemGeneralCallbackEventUdsStatusChangedFnc ⌈

Parameter Name	DemGeneralCallbackEventUdsStatusChangedFnc
Parent Container	DemGeneral
Upstream mapping defined	–
Description	Function name of prototype "<DemGeneralTriggerOnEventUdsStatus>". Function for General Event UDS Status Changed notification.
Multiplicity	0..1
Type	EcucFunctionNameDef
Default value	–
Regular Expression	–

▽

△

Pre-Compile only	true

⌋

[ECUC_Dem_00938] Definition of EcucFunctionNameDef DemGeneralCallback MonitorStatusChangedFnc ⌈

Parameter Name	DemGeneralCallbackMonitorStatusChangedFnc
Parent Container	DemGeneral
Upstream mapping defined	–
Description	Function name of prototype "<Module>_DemGeneralTriggerOnMonitorStatus". Function for the General Monitor Status Changed notification.
Multiplicity	0..1
Type	EcucFunctionNameDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

[ECUC_Dem_00880] Definition of EcucBooleanParamDef DemGeneralInterface Support ⌈

Parameter Name	DemGeneralInterfaceSupport
Parent Container	DemGeneral
Upstream mapping defined	–
Description	The interfaces GeneralEvtInfo, GeneralCallbackEventDataChanged, GeneralCallback MonitorStatusChanged and GeneralCallbackEventUdsStatusChange are provided if DemGeneralInterfaceSupport is equal to true.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00692] Definition of EcucIntegerParamDef DemMaxNumberPrestoredFF ⌈

Parameter Name	DemMaxNumberPrestoredFF	
Parent Container	DemGeneral	
Upstream mapping defined	–	
Description	Defines the maximum number for prestored freeze frames. If set to 0, then freeze frame prestorage is not supported by the ECU.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00698] Definition of EcucEnumerationParamDef DemOBDSupport ⌈

Parameter Name	DemOBDSupport	
Parent Container	DemGeneral	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This configuration switch defines OBD support and kind of OBD ECU.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_OBD_DEP_SEC_ECU	Kind of OBD ECU: OBD Dependend / Secondary ECU
	DEM_OBD_MASTER_ECU	Kind of OBD ECU: Master ECU
	DEM_OBD_NO_OBD_SUPPORT	OBD is not supported within this ECU
	DEM_OBD_PRIMARY_ECU	Kind of OBD ECU: Pimary ECU
Pre-Compile only	true	

⌋

[ECUC_Dem_00704] Definition of EcucBooleanParamDef DemPTOSupport ⌈

Parameter Name	DemPTOSupport
Parent Container	DemGeneral
Upstream mapping defined	–
Description	This configuration switch defines, whether PTO support (and therefore PID $1E support) is enabled or not.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00799] Definition of EcucBooleanParamDef DemResetConfirmedBitOnOverflow ⌈

Parameter Name	DemResetConfirmedBitOnOverflow
Parent Container	DemGeneral
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This configuration switch defines, whether the confirmed bit is reset or not while an event memory entry will be displaced.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	true
Pre-Compile only	true

⌋

[ECUC_Dem_00991] Definition of EcucBooleanParamDef DemResetPendingBitOnOverflow ⌈

Parameter Name	DemResetPendingBitOnOverflow
Parent Container	DemGeneral
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This configuration switch defines, whether the pending bit is reset or not while an event memory entry will be displaced.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	true
Pre-Compile only	true

⌋

[ECUC_Dem_00784] Definition of EcucEnumerationParamDef DemStatusBitHandlingTestFailedSinceLastClear ⌈

Parameter Name	DemStatusBitHandlingTestFailedSinceLastClear	
Parent Container	DemGeneral	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This configuration switch defines, whether the aging and displacement mechanism shall be applied to the "TestFailedSinceLastClear" status bits.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	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
Default value	DEM_STATUS_BIT_NORMAL	
Pre-Compile only	true	

⌋

[ECUC_Dem_00793] Definition of EcucEnumerationParamDef DemSuppressionSupport ⌈

Parameter Name	DemSuppressionSupport	
Parent Container	DemGeneral	
Upstream mapping defined	–	
Description	This configuration switch defines, whether support for suppression is enabled or not.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_DTC_SUPPRESSION	Support suppression by DTC
	DEM_NO_SUPPRESSION	Suppression is not supported
Pre-Compile only	true	

⌋

[ECUC_Dem_00715] Definition of EcucFloatParamDef DemTaskTime ⌈

Parameter Name	DemTaskTime	
Parent Container	DemGeneral	
Upstream mapping defined	–	
Description	Allow to configure the time for the periodic cyclic task. Please note: This configuration value shall be equal to the value in the Basic Software Scheduler configuration of the RTE module. The AUTOSAR configuration standard is to use SI units, so this parameter is defined as float value in seconds. Dem configuration tools must convert this float value to the appropriate value format for the use in the software implementation of Dem. min: A negative value is not allowed. max: After 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). upperMultiplicity: Exactly one TaskTime must be specified per configuration. lowerMultiplicity: Exactly one TaskTime must be specified per configuration.	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	]0 .. INF[	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00719] Definition of EcucBooleanParamDef DemTriggerFiMReports ⌈

Parameter Name	DemTriggerFiMReports
Parent Container	DemGeneral
Upstream mapping defined	–
Description	Activate/Deactivate the notification to the Function Inhibition Manager. true: FiM notification activated false: FiM notification deactivated
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00765] Definition of EcucBooleanParamDef DemTriggerMonitorInitBeforeClearOk ⌈

Parameter Name	DemTriggerMonitorInitBeforeClearOk
Parent Container	DemGeneral
Upstream mapping defined	–
Description	Defines if the monitor re-initialization has to be triggered before or after the Dem module returns DEM_CLEAR_OK. true: trigger re-initialization before DEM_CLEAR_OK false: trigger re-initialization after DEM_CLEAR_OK
Multiplicity	1
Type	EcucBooleanParamDef

▽

△

Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00941] Definition of EcucReferenceDef DemClearEventsWithoutDTCEventMemoryRef ⌈

Parameter Name	DemClearEventsWithoutDTCEventMemoryRef
Parent Container	DemGeneral
Upstream mapping defined	–
Description	Indicating the event memory used as trigger to clear events without assigned DTCs.
Multiplicity	0..1
Type	Reference to DemPrimaryMemory
Pre-Compile only	true

⌋

[ECUC_Dem_00940] Definition of EcucReferenceDef DemOBDEventMemorySetRef ⌈

Parameter Name	DemOBDEventMemorySetRef
Parent Container	DemGeneral
Upstream mapping defined	–
Description	References the DemEventMemorySet used for OBD ECU.
Multiplicity	0..1
Type	Reference to DemEventMemorySet
Pre-Compile only	true

⌋

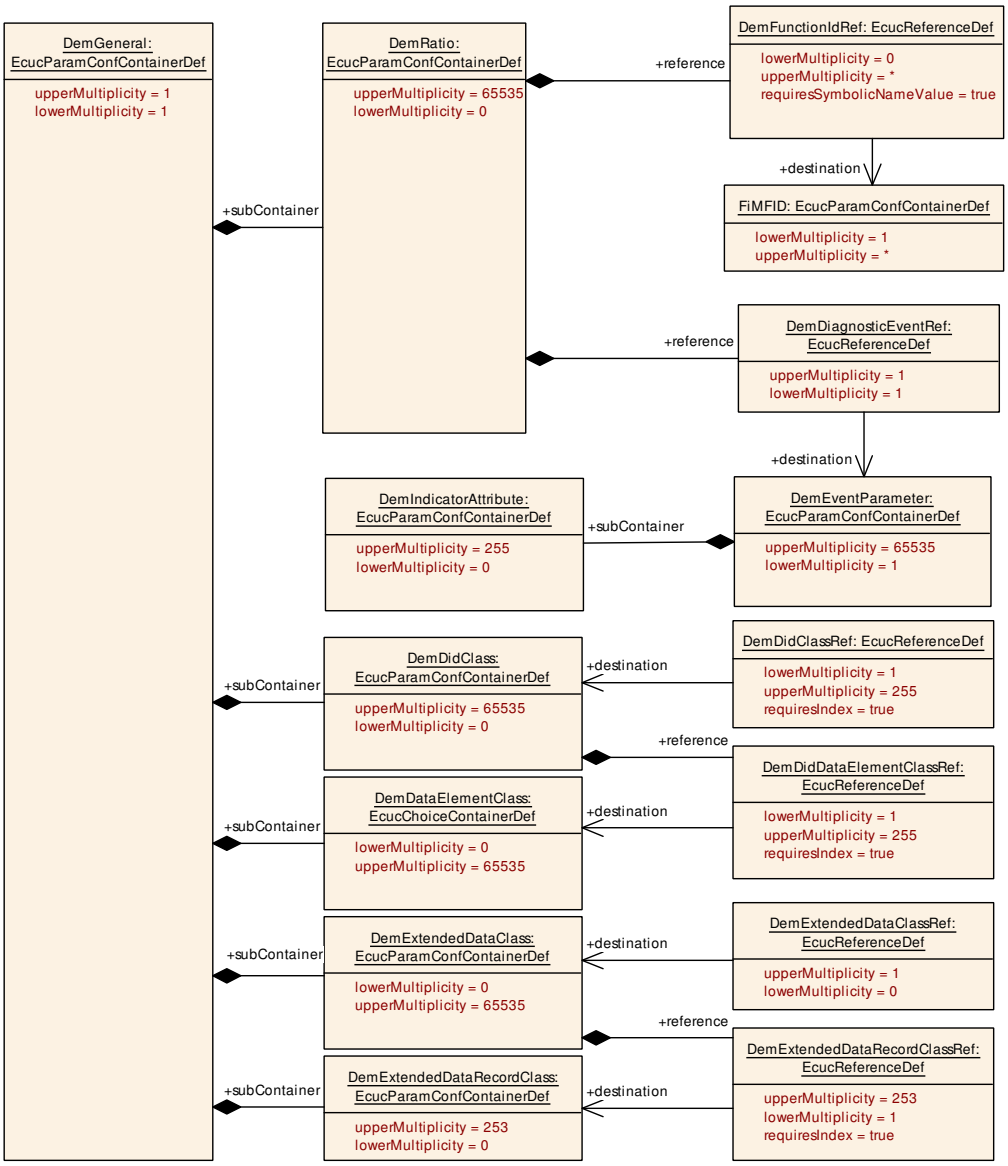


Figure 10.1: DemGeneral configuration overview (part I)

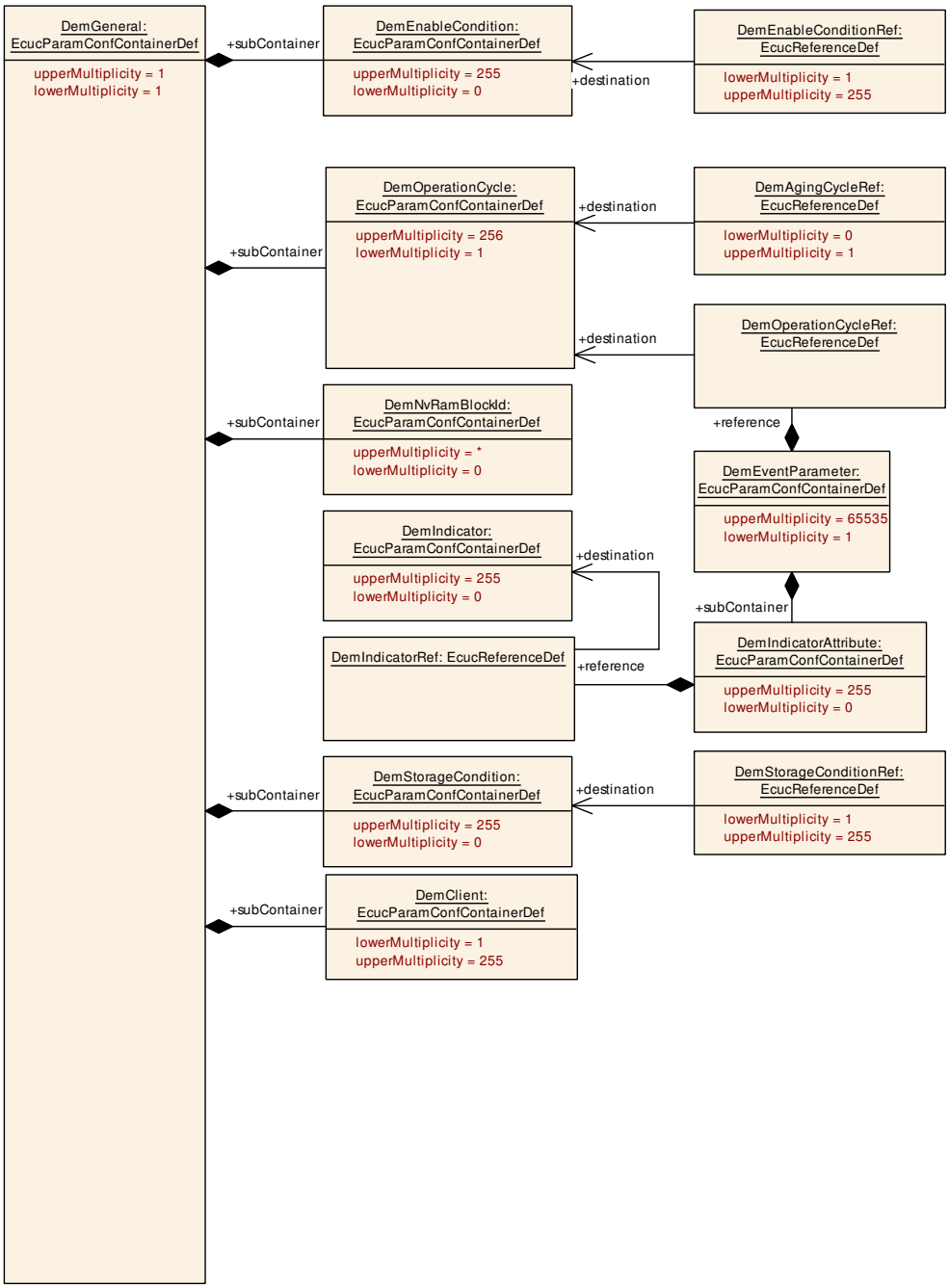


Figure 10.2: DemGeneral configuration overview (part II)

10.2.2.2 DemConfigSet

[ECUC_Dem_00634] Definition of EcucParamConfContainerDef DemConfigSet ⌈

Container Name	DemConfigSet
Parent Container	Dem
Upstream mapping defined	–
Description	This container contains the configuration parameters and sub containers of the Dem module supporting multiple configuration sets.
Multiplicity	1
Pre-Compile only	true
Configuration Parameters	

No Included Parameters

Included Containers		
Container Name	Multiplicity	Description
DemComponent	0..*	This container configures the monitored components and system dependencies.
DemDTC	0..65535	This container contains the configuration (parameters) for Dem UdsDTC.
DemDTCAttributes	0..65535	This container contains the configuration (parameters) for Dem DTCAttributes.
DemDebounceCounterBasedClass	0..65535	This container contains the configuration of Debounce Counter Based Class
DemDebounceTimeBasedClass	0..65535	This container contains the configuration of Debounce Time Based Class.
DemDtrs	0..1	This container holds the configuration of DTRs collection.
DemEventParameter	1..65535	This container contains the configuration (parameters) for events.
DemFreezeFrameClass	0..65535	This container contains the combinations of DIDs for freeze frames of UDS and emission related diagnostics over UDS.
DemMultiEventTriggering	0..65535	Configures an event that will trigger other events whenever the event is reported.
DemObdDTC	0..65535	This container contains the configuration (parameters) for Dem ObdDTC.
DemPidClass	0..255	This container contains the different PIDs for the single global OBD relevant freeze frame class. It is assembled out of one or several data elements.

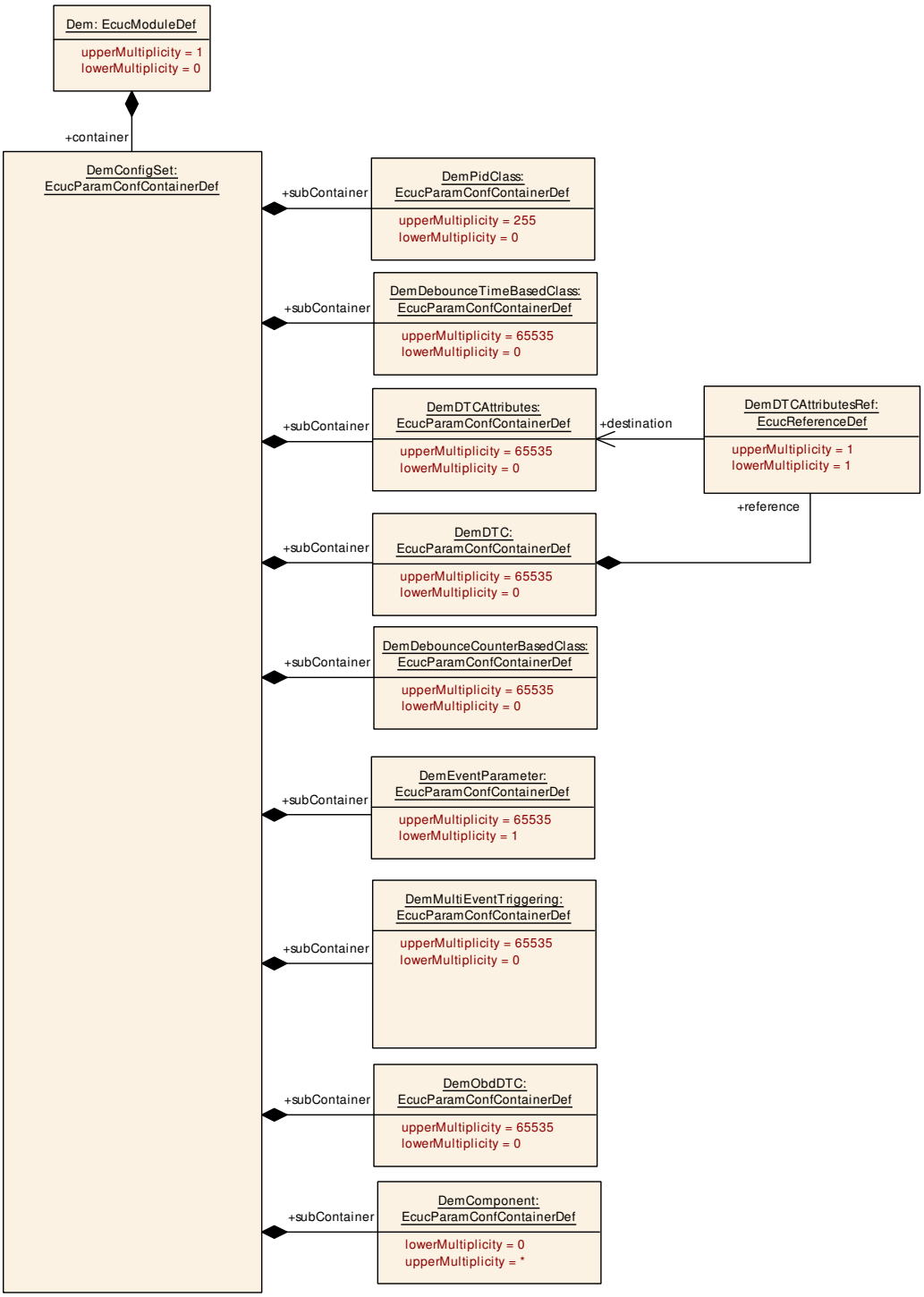


Figure 10.3: DemConfigSet configuration overview

10.2.2.3 DemClient

[ECUC_Dem_00931] Definition of EcucParamConfContainerDef DemClient ⌈

Container Name	DemClient
Parent Container	DemGeneral
Upstream mapping defined	–
Description	This container contains possible clients that are using the Dem APIs.
Multiplicity	1..255
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemClientFunctionality	1	[ECUC_Dem_00943]
DemClientId	1	[ECUC_Dem_00932]
DemClientUsesRte	1	[ECUC_Dem_00933]
DemEventMemorySetRef	1	[ECUC_Dem_00942]

Included Containers		
Container Name	**Multiplicity**	**Description**
DemCallbackDTCStatusChanged	0..*	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. In case there is a DemCallbackDTCStatusChangedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackDTCStatusChangedFnc, the Dem will have an R-Port requiring the interface Callback DTCStatusChanged. Status change notifications are supported for DTCs in primary memory only.

⌋

[ECUC_Dem_00943] Definition of EcucEnumerationParamDef DemClientFunctionality ⌈

Parameter Name	DemClientFunctionality	
Parent Container	DemClient	
Upstream mapping defined	–	
Description	Functionality provided by Dem for the DemClient.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_CLIENT_ONLY_USES_EVENTOVERFLOW_INTERFACE	The Dem provides only the client related functionality EvMemOverflowIndication for this client Id. Other functionality like DTCFilter or Clear is not allowed to be used by this client.
	DEM_CLIENT_USES_FULL_FUNCTIONALITY	The Dem provides all client related functionality for the selected client (e.g. clear APIs and overflow indications).
Default value	DEM_CLIENT_USES_FULL_FUNCTIONALITY	
Post-Build Value	-	Selectable

⌋

[ECUC_Dem_00932] Definition of EcucIntegerParamDef DemClientId ⌈

Parameter Name	DemClientId	
Parent Container	DemClient	
Upstream mapping defined	–	
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	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00933] Definition of EcucBooleanParamDef DemClientUsesRte ⌈

Parameter Name	DemClientUsesRte
Parent Container	DemClient
Upstream mapping defined	–
Description	If set to true, this client shall only use the DEM via RTE (Dem will provide the C/S interfaces: CddIf, EvMemOverflowIndication). The client must not do any C_API calls to the DEM.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_Dem_00942] Definition of EcucReferenceDef DemEventMemorySetRef ⌈

Parameter Name	DemEventMemorySetRef
Parent Container	DemClient
Upstream mapping defined	–
Description	References to the client assigned event memory container that contains client specific settings and event memories.
Multiplicity	1
Type	Reference to DemEventMemorySet
Pre-Compile only	true

⌋

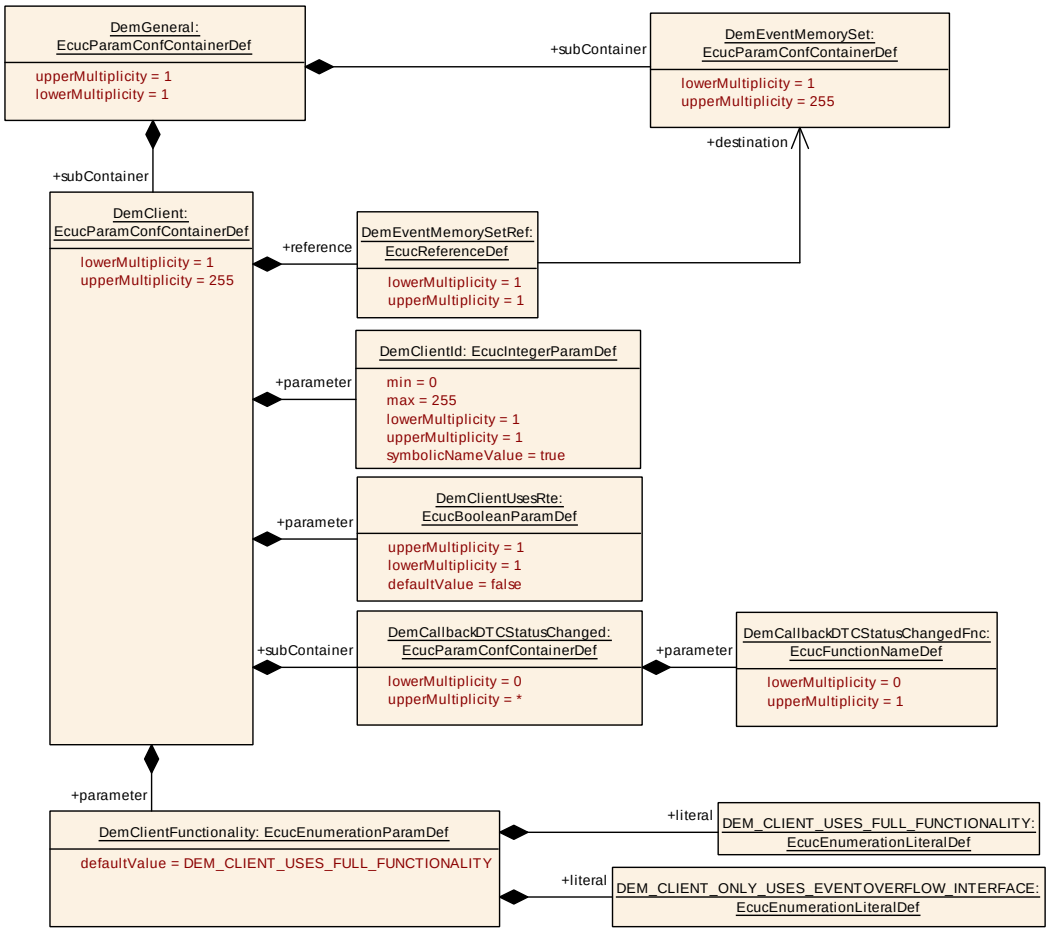


Figure 10.4: DemClient configuration overview

10.2.2.4 DemDTCAttributes

[ECUC_Dem_00641] Definition of EcucParamConfContainerDef DemDTCAttributes ⌈

Container Name	DemDTCAttributes
Parent Container	DemConfigSet
Upstream mapping defined	–
Description	This container contains the configuration (parameters) for DemDTCAttributes.
Multiplicity	0..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemAgingCycleCounterThreshold	0..1	[ECUC_Dem_00623]
DemDTCPriority	1	[ECUC_Dem_00662]
DemDTCSignificance	0..1	[ECUC_Dem_00779]
DemMaxNumberFreezeFrameRecords	0..1	[ECUC_Dem_00605]
DemAgingCycleRef	0..1	[ECUC_Dem_00624]
DemExtendedDataClassRef	0..1	[ECUC_Dem_00667]
DemFreezeFrameClassRef	0..1	[ECUC_Dem_00674]
DemFreezeFrameRecNumClassRef	0..1	[ECUC_Dem_00776]
DemJ1939ExpandedFreezeFrameClassRef	0..1	[ECUC_Dem_00834]
DemJ1939FreezeFrameClassRef	0..1	[ECUC_Dem_00835]
DemMemoryDestinationRef	1	[ECUC_Dem_00890]
DemUDSOBDFreezeFrameClassRef	0..1	[ECUC_Dem_00911]

No Included Containers

⌋

[ECUC_Dem_00623] Definition of EcucIntegerParamDef DemAgingCycleCounterThreshold ⌈

Parameter Name	DemAgingCycleCounterThreshold	
Parent Container	DemDTCAttributes	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Number of aging cycles needed to unlearn/delete the event.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	1 .. 255	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00662] Definition of EcucIntegerParamDef DemDTCPriority ⌈

Parameter Name	DemDTCPriority	
Parent Container	DemDTCAttributes	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Priority of the event/dtc, in view of full event memory. A lower value means higher priority.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	1 .. 255	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00779] Definition of EcucEnumerationParamDef DemDTCSignificance ⌈

Parameter Name	DemDTCSignificance	
Parent Container	DemDTCAttributes	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Significance of the event, which indicates additional information concerning fault classification and resolution. It can be mapped as Dem-internal data element. It shall be configured, if it is a part of event related data.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	DEM_EVENT_SIGNIFICANCE_FAULT	failure, which affects the component/ECU itself
	DEM_EVENT_SIGNIFICANCE_OCCURRENCE	issue, which indicates additional information concerning insufficient system behavior
Pre-Compile only	true	

⌋

[ECUC_Dem_00605] Definition of EcucIntegerParamDef DemMaxNumberFreezeFrameRecords ⌈

Parameter Name	DemMaxNumberFreezeFrameRecords	
Parent Container	DemDTCAttributes	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This parameter defines the number of according freeze frame records, which can maximal be stored for this event. Therefore all these freeze frame records have the same freeze frame class. This parameter is only required for calculated record numeration (refer to DemTypeOfFreezeFrameRecordNumeration).	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00624] Definition of EcucReferenceDef DemAgingCycleRef ⌈

Parameter Name	DemAgingCycleRef	
Parent Container	DemDTCAttributes	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Reference to the cycle which is triggering the aging of the event.	
Multiplicity	0..1	
Type	Reference to DemOperationCycle	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00667] Definition of EcucReferenceDef DemExtendedDataClassRef ⌈

Parameter Name	DemExtendedDataClassRef	
Parent Container	DemDTCAttributes	
Upstream mapping defined	–	
Description	This reference defines the link to an extended data class sampler.	
Multiplicity	0..1	
Type	Reference to DemExtendedDataClass	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00674] Definition of EcucReferenceDef DemFreezeFrameClassRef ⌈

Parameter Name	DemFreezeFrameClassRef
Parent Container	DemDTCAttributes
Upstream mapping defined	–
Description	These references define the links to a freeze frame class sampler.
Multiplicity	0..1
Type	Reference to DemFreezeFrameClass
Pre-Compile only	true

⌋

[ECUC_Dem_00776] Definition of EcucReferenceDef DemFreezeFrameRecNumClassRef ⌈

Parameter Name	DemFreezeFrameRecNumClassRef
Parent Container	DemDTCAttributes
Upstream mapping defined	–
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). This parameter is only required for configured record numeration (refer to DemTypeOfFreezeFrameRecordNumeration).
Multiplicity	0..1
Type	Reference to DemFreezeFrameRecNumClass
Pre-Compile only	true

⌋

[ECUC_Dem_00834] Definition of EcucReferenceDef DemJ1939ExpandedFreezeFrameClassRef ⌈

Parameter Name	DemJ1939ExpandedFreezeFrameClassRef
Parent Container	DemDTCAttributes
Upstream mapping defined	–
Description	These references define the links to a J1939 freeze frame class sampler.

▽

△

Multiplicity	0..1
Type	Reference to DemJ1939FreezeFrameClass
Pre-Compile only	true

⌋

[ECUC_Dem_00835] Definition of EcucReferenceDef DemJ1939FreezeFrameClassRef ⌈

Parameter Name	DemJ1939FreezeFrameClassRef
Parent Container	DemDTCAttributes
Upstream mapping defined	–
Description	These references define the links to a J1939 freeze frame class sampler.
Multiplicity	0..1
Type	Reference to DemJ1939FreezeFrameClass
Pre-Compile only	true

⌋

[ECUC_Dem_00890] Definition of EcucChoiceReferenceDef DemMemoryDestinationRef ⌈

Parameter Name	DemMemoryDestinationRef
Parent Container	DemDTCAttributes
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	The memory destination assigns DTCs to one memory destination.
Multiplicity	1
Type	Choice reference to [DemPrimaryMemory, DemUserDefinedMemory]
Pre-Compile only	true

⌋

[ECUC_Dem_00911] Definition of EcucReferenceDef DemUDSOBDFreezeFrameClassRef ⌈

Parameter Name	DemUDSOBDFreezeFrameClassRef	
Parent Container	DemDTCAttributes	
Upstream mapping defined	–	
Description	This reference defines the link to the freeze frame for emission related diagnostics over UDS.	
Multiplicity	0..1	
Type	Reference to DemFreezeFrameClass	
Post-Build Multiplicity	Loadable	Selectable

⌋

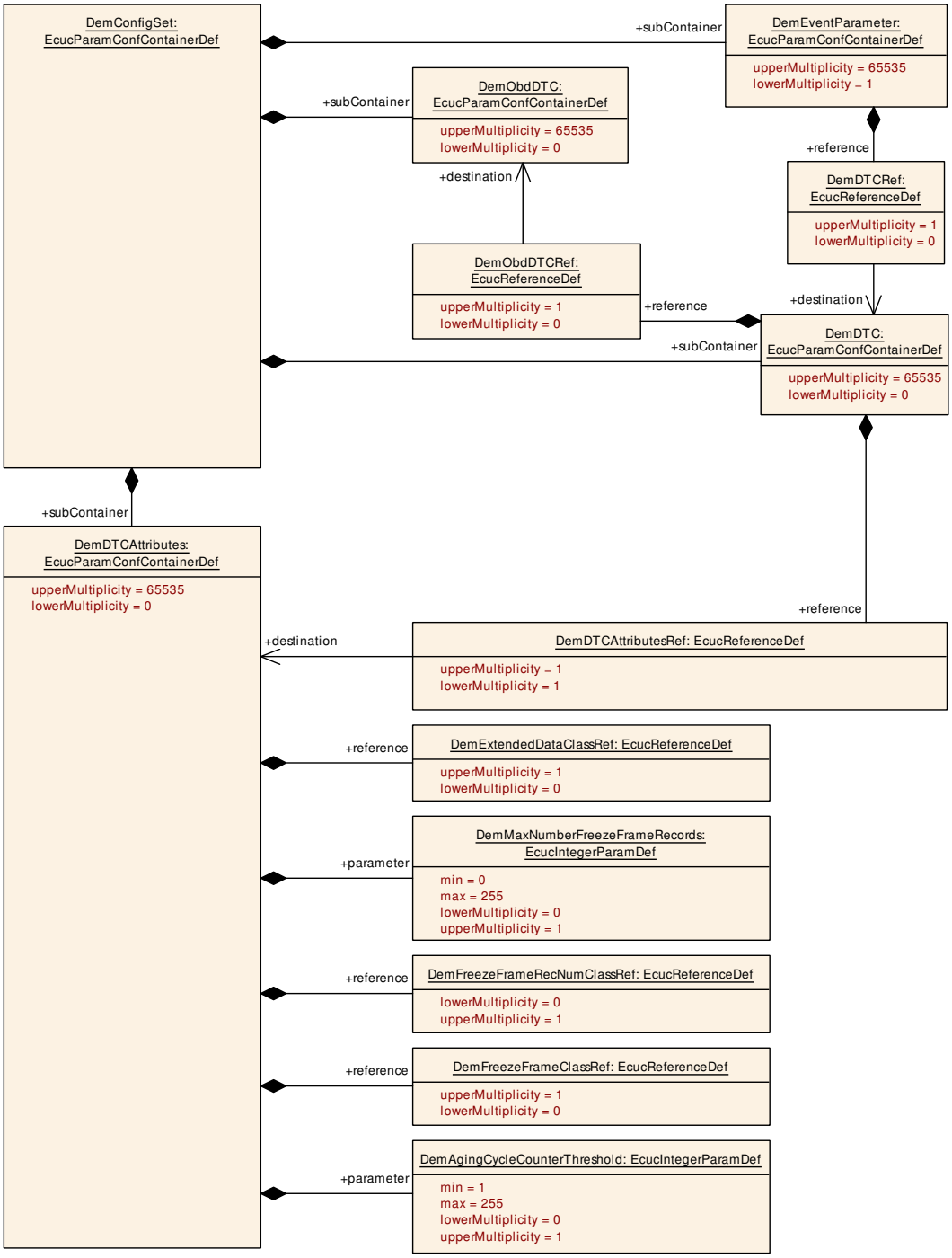


Figure 10.5: DTC attribute configuration overview (part I)

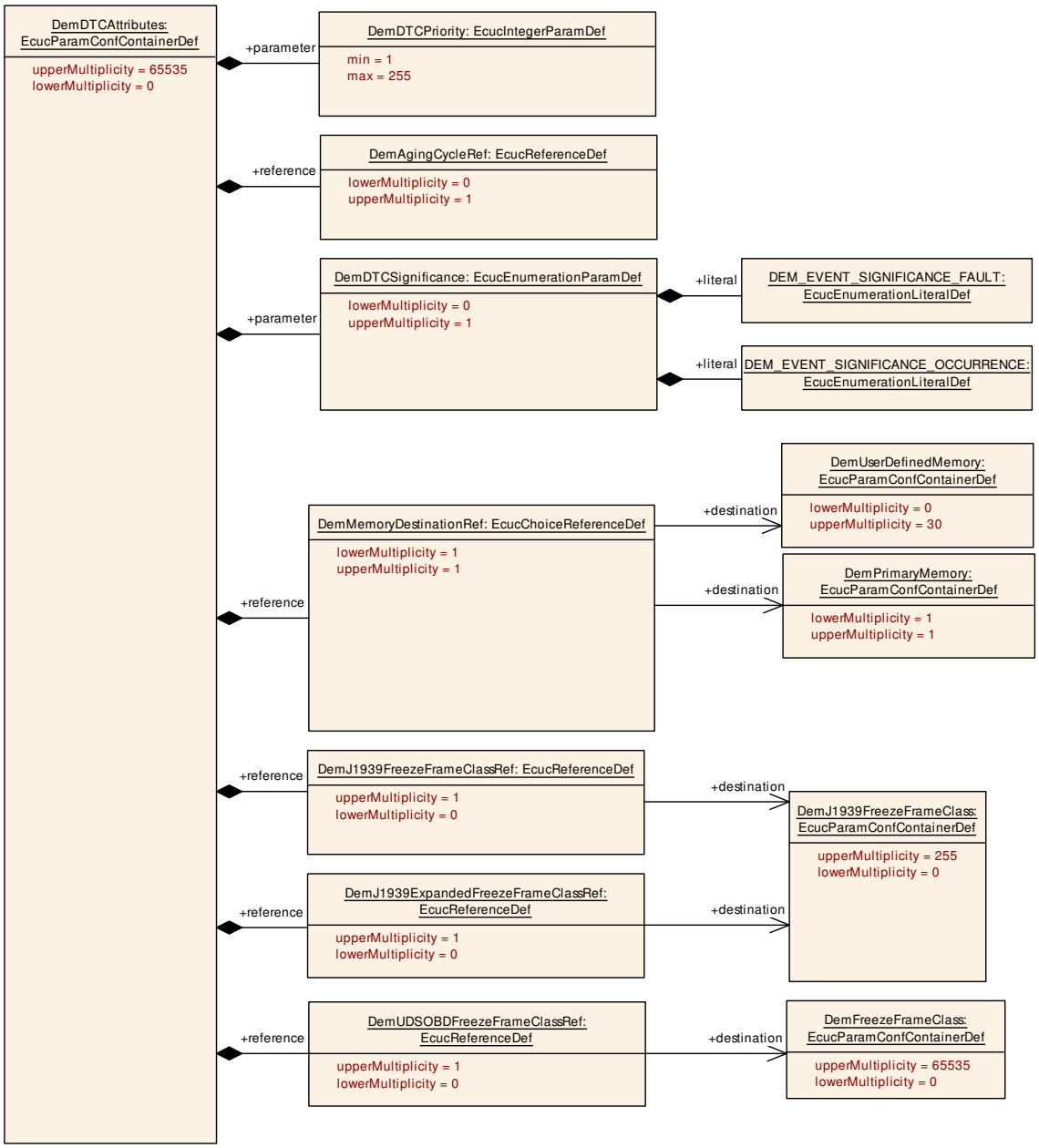


Figure 10.6: DTC attribute configuration overview (part II)

10.2.2.5 DemEventParameter

[ECUC_Dem_00661] Definition of EcucParamConfContainerDef DemEventParameter ⌈

Container Name	DemEventParameter
Parent Container	DemConfigSet
Upstream mapping defined	[24, CP_TPS_BSWModuleDescriptionTemplate]

▽

△

Description	This container contains the configuration (parameters) for events.
Multiplicity	1..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemCausalityDelayTime	0..1	[ECUC_Dem_00921]
DemComponentPriority	0..1	[ECUC_Dem_00909]
DemEventAssociatedIdentification	0..1	[ECUC_Dem_00969]
DemEventAvailable	1	[ECUC_Dem_00792]
DemEventAvailableInVariant	0..1	[ECUC_Dem_00968]
DemEventConfirmationThreshold	1	[ECUC_Dem_00924]
DemEventConfirmationThresholdCounterAdaptable	0..1	[ECUC_Dem_00929]
DemEventId	1	[ECUC_Dem_00659]
DemEventKind	1	[ECUC_Dem_00660]
DemEventRecoverableInSameOperationCycle	0..1	[ECUC_Dem_00916]
DemEventReportingType	1	[ECUC_Dem_00970]
DemFFPrestorageInNvm	0..1	[ECUC_Dem_00948]
DemFFPrestorageSupported	1	[ECUC_Dem_00671]
DemReportBehavior	0..1	[ECUC_Dem_00894]
DemComponentClassRef	0..1	[ECUC_Dem_00908]
DemDTCRef	0..1	[ECUC_Dem_00888]
DemEnableConditionGroupRef	0..1	[ECUC_Dem_00746]
DemOBDGroupingAssociativeEventsRef	0..1	[ECUC_Dem_00839]
DemOperationCycleRef	1	[ECUC_Dem_00702]
DemStorageConditionGroupRef	0..1	[ECUC_Dem_00769]

Included Containers		
Container Name	**Multiplicity**	**Description**
DemCallbackClearEventAllowed	0..1	The presence of this container indicates that the Dem has access to a "ClearEventAllowed" callback. In case there is a DemCallbackClearEventAllowedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackClearEventAllowedFnc, the Dem will have an R-Port requiring the interface CallbackClear EventAllowed whose name is generated by using the unique callback-prefix followed by the event name.
DemCallbackEventDataChanged	0..1	The presence of this container indicates that the Dem has access to an "EventDataChanged" callback. In case there is a DemCallbackEventDataChangedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackEventDataChangedFnc, the Dem will have an R-Port requiring the interface CallbackEvent DataChanged whose name is generated by using the unique callback-prefix followed by the event name.

▽

△

Included Containers		
Container Name	**Multiplicity**	**Description**
DemCallbackEventUdsStatus Changed	0..*	The presence of this container indicates, that the Dem has access to an "EventUdsStatusChanged" callback, which the Dem will call to notify other components about the change in the status of an event. In case there is a DemCallbackEventUdsStatusChangedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackEventUdsStatusChangedFnc, the Dem will have an R-Port requiring the interface Callback EventUdsStatusChanged, whose name is generated by using the unique callback-prefix followed by the event name.
DemCallbackInitMForE	0..1	The presence of this container indicates, that the Dem has access to an "InitMonitorForEvent" callback, which the Dem will call to initialize a monitor. In case the container has a DemCallbackInitMForEFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackInitMForEFnc, the Dem will have an R-Port requiring the interface CallbackInitMonitorForEvent, whose name is generated by using the unique callback-prefix followed by the event name.
DemCallbackMonitorStatus Changed	0..*	The presence of this container indicates, that the Dem has access to an "MonitorStatusChanged" callback, which the Dem will call to notify other components about the change in the status of an event. In case there is a DemCallbackMonitorStatusChangedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackMonitorStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackMonitor StatusChanged, whose name is generated by using the unique callback-prefix followed by the event name.
DemDebounceAlgorithmClass	1	Debounce algorithm class: counter based, time based, or monitor internal.
DemIndicatorAttribute	0..255	This container contains the event specific configuration of Indicators.

⌋

[ECUC_Dem_00921] Definition of EcucFloatParamDef DemCausalityDelayTime ⌈

Parameter Name	DemCausalityDelayTime	
Parent Container	DemEventParameter	
Upstream mapping defined	–	
Description	Time to wait until the event is considered as causal. The parameter is specified in seconds.	
Multiplicity	0..1	
Type	EcucFloatParamDef	
Range	[0 .. 2.5]	
Default value	0	
Pre-Compile only	true	

⌋

[ECUC_Dem_00909] Definition of EcucIntegerParamDef DemComponentPriority ⌈

Parameter Name	DemComponentPriority	
Parent Container	DemEventParameter	
Upstream mapping defined	–	
Description	Specifies the priority within the component. A lower value means higher priority.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00969] Definition of EcucIntegerParamDef DemEventAssociatedIdentification ⌈

Parameter Name	DemEventAssociatedIdentification	
Parent Container	DemEventParameter	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Event associated identifier that allows to identify an event. This value can be reported as internal data element in snapshot records or extended data records.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 65535	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00792] Definition of EcucBooleanParamDef DemEventAvailable ⌈

Parameter Name	DemEventAvailable	
Parent Container	DemEventParameter	
Upstream mapping defined	–	
Description	This parameter configures an Event as unavailable. It is treated by Dem as if it does not exist. true = Event is available false = Event is not available	
Multiplicity	1	
Type	EcucBooleanParamDef	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00968] Definition of EcucBooleanParamDef DemEventAvailableIn Variant ⌈

Parameter Name	DemEventAvailableInVariant
Parent Container	DemEventParameter
Upstream mapping defined	–

▽

△

Description	This parameter defines the enabled or disabled status of an event. If set to FALSE, the Event is generally disabled, even though the SW-C is integrated and the Event is described in the ECUC. The Dem shall ignore all calls from the SW-C. If set to TRUE, the availability of the event depends on other availability sources.	
Multiplicity	0..1	
Type	EcucBooleanParamDef	
Default value	true	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00924] Definition of EcucIntegerParamDef DemEventConfirmationThreshold ⌈

Parameter Name	DemEventConfirmationThreshold	
Parent Container	DemEventParameter	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Defines the operation cycle threshold of the DTC confirmation status according "Confirmation Threshold" of ISO 14229-1.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	1 .. 255	
Default value	1	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00929] Definition of EcucBooleanParamDef DemEventConfirmationThresholdCounterAdaptable ⌈

Parameter Name	DemEventConfirmationThresholdCounterAdaptable
Parent Container	DemEventParameter
Upstream mapping defined	–
Description	Indicates whether the events confirmation Cycle threshold can be adapted by Dem_SetEventConfirmationThresholdCounter.
Multiplicity	0..1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_Dem_00659] Definition of EcucIntegerParamDef DemEventId ⌈

Parameter Name	DemEventId
Parent Container	DemEventParameter
Upstream mapping defined	–

▽

△

Description	Unique identifier of a diagnostic event. 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. The events should be sequentially ordered beginning with 1 and no gaps in between.	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	1 .. 65535	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00660] Definition of EcucEnumerationParamDef DemEventKind ⌈

Parameter Name	DemEventKind	
Parent Container	DemEventParameter	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This parameter is used to distinguish between SW-C and BSW events.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_EVENT_KIND_BSW	The event is a assigned to a BSW module
	DEM_EVENT_KIND_SWC	The event is a assigned to a SW-C
Default value	DEM_EVENT_KIND_SWC	
Pre-Compile only	true	

⌋

[ECUC_Dem_00916] Definition of EcucBooleanParamDef DemEventRecoverable InSameOperationCycle ⌈

Parameter Name	DemEventRecoverableInSameOperationCycle	
Parent Container	DemEventParameter	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	If parameter is configured to FALSE, reporting of PASSED will be ignored if the event is already "testfailed this operation cycle".	
Multiplicity	0..1	
Type	EcucBooleanParamDef	
Default value	true	
Post-Build Value	-	Selectable

⌋

[ECUC_Dem_00970] Definition of EcucEnumerationParamDef DemEventReportingType

Status: DRAFT

⌈

Parameter Name	DemEventReportingType	
Parent Container	DemEventParameter	
Upstream mapping defined	–	
Description	This parameter defines the way a monitor can report an event to the Dem. There are various ways to report an event and the Dem provides different APIs for it. **Tags:** atp.Status=draft	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	STANDARD_REPORTING	The event is reported over the API Dem_SetEventStatus. **Tags:** atp.Status=draft
	STANDARD_REPORTING_WITH_MONITOR_DATA	The event is reported over the API Dem_SetEventStatusWithMonitorData. **Tags:** atp.Status=draft
Default value	STANDARD_REPORTING	
Pre-Compile only	true	

⌋

[ECUC_Dem_00948] Definition of EcucBooleanParamDef DemFFPrestorageInNvm ⌈

Parameter Name	DemFFPrestorageInNvm
Parent Container	DemEventParameter
Upstream mapping defined	[22, CP_TPS_SoftwareComponentTemplate], [23, CP_TPS_DiagnosticExtractTemplate]
Description	If the event uses a pre-stored freeze-frame this attribute indicates if the event requires the pre-stored data to be stored in non-volatile memory. TRUE = store the pre-stored data in non-volatile memory, FALSE = pre-stored data is not stored in non-volatile memory.
Multiplicity	0..1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00671] Definition of EcucBooleanParamDef DemFFPrestorageSupported ⌈

Parameter Name	DemFFPrestorageSupported
Parent Container	DemEventParameter
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	If this parameter is set to true, then the Prestorage of FreezeFrames is supported by the assigned event. This parameter is useful to calculate the buffer size.

▽

△

Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_Dem_00894] Definition of EcucEnumerationParamDef DemReportBehavior ⌈

Parameter Name	DemReportBehavior	
Parent Container	DemEventParameter	
Upstream mapping defined	–	
Description	Indicates the reporting behavior of the BSW Module (DemEventKind == DEM_EVENT_KIND_BSW) in order to determine the size of the reporting queue. If the parameter is not defined it means REPORT_BEFORE_INIT.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	REPORT_AFTER_INIT	Indicates that the Event will not be reported before Dem_Init().
	REPORT_BEFORE_INIT	Indicates that the Event may be reported before Dem_Init().
Default value	REPORT_BEFORE_INIT	
Pre-Compile only	true	

⌋

[ECUC_Dem_00908] Definition of EcucReferenceDef DemComponentClassRef ⌈

Parameter Name	DemComponentClassRef
Parent Container	DemEventParameter
Upstream mapping defined	–
Description	Reference to the monitored component.
Multiplicity	0..1
Type	Reference to DemComponent
Pre-Compile only	true

⌋

[ECUC_Dem_00888] Definition of EcucReferenceDef DemDTCRef ⌈

Parameter Name	DemDTCRef
Parent Container	DemEventParameter
Upstream mapping defined	–
Description	This parameter defines the DTC configuration (typically Uds) associated with the diagnostic event. 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 Dem DTCAttributes can be used from several events, to combine these (refer to chapter "Combination of diagnostic event").

▽

△

Multiplicity	0..1
Type	Reference to DemDTC
Pre-Compile only	true

⌋

[ECUC_Dem_00746] Definition of EcucReferenceDef DemEnableConditionGroupRef ⌈

Parameter Name	DemEnableConditionGroupRef	
Parent Container	DemEventParameter	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	References an enable condition group.	
Multiplicity	0..1	
Type	Reference to DemEnableConditionGroup	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00839] Definition of EcucReferenceDef DemOBDGroupingAssociativeEventsRef ⌈

Parameter Name	DemOBDGroupingAssociativeEventsRef
Parent Container	DemEventParameter
Upstream mapping defined	–
Description	This parameter defines a reference which points to a representative event of one group of associate events. The "reverence event" must refer to it self. Note: One event is only allowed to be reverenced to only one group of associate events.
Multiplicity	0..1
Type	Reference to DemEventParameter
Pre-Compile only	true

⌋

[ECUC_Dem_00702] Definition of EcucReferenceDef DemOperationCycleRef ⌈

Parameter Name	DemOperationCycleRef	
Parent Container	DemEventParameter	
Upstream mapping defined	–	
Description	Kind of operation cycle for the event (e.g. power cycle, driving cycle, ...)	
Multiplicity	1	
Type	Reference to DemOperationCycle	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00769] Definition of EcucReferenceDef DemStorageConditionGroupRef ⌈

Parameter Name	DemStorageConditionGroupRef
Parent Container	DemEventParameter
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	References a storage condition group.
Multiplicity	0..1
Type	Reference to DemStorageConditionGroup
Pre-Compile only	true

⌋

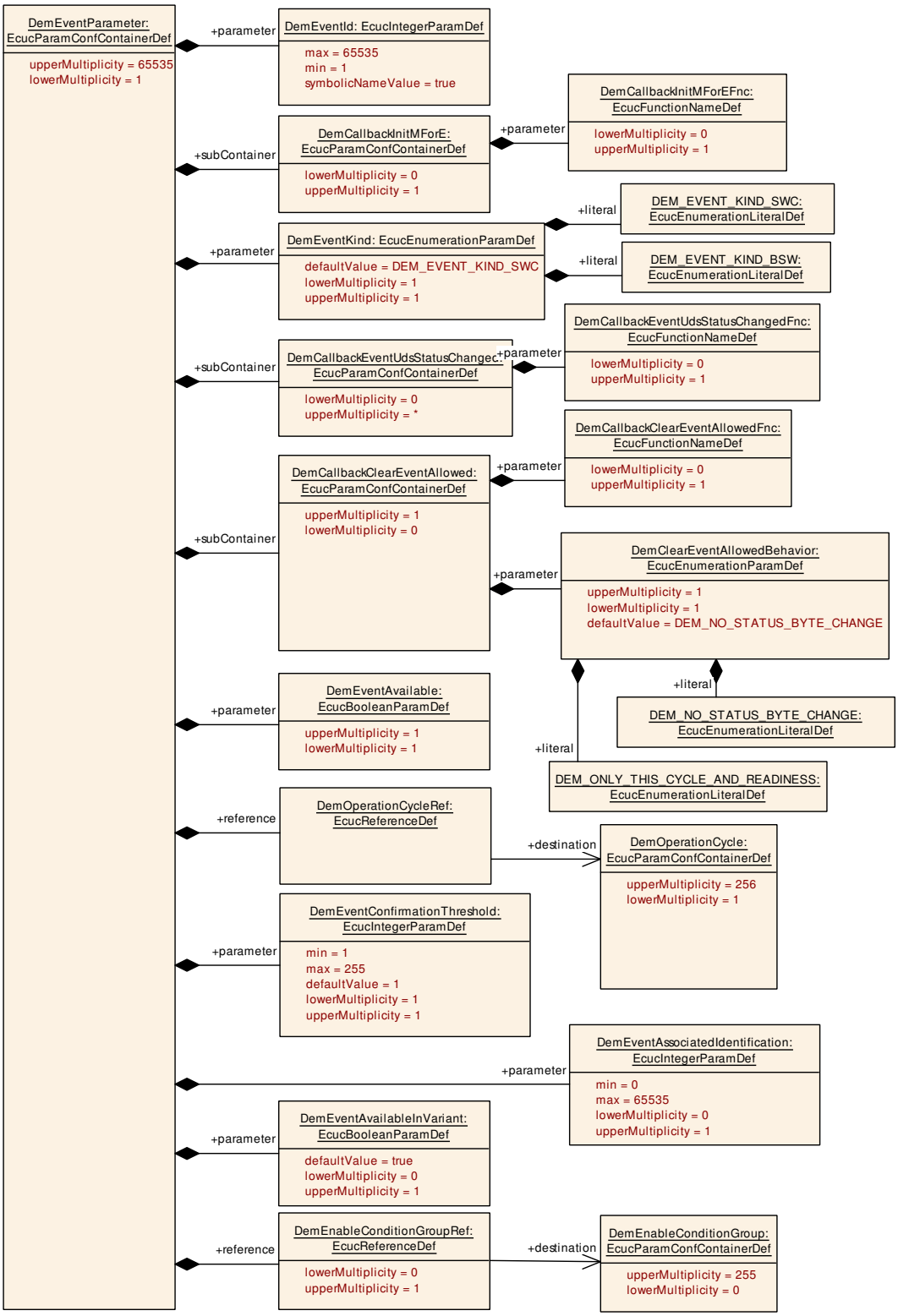


Figure 10.7: DemEventParameter configuration overview (part I)

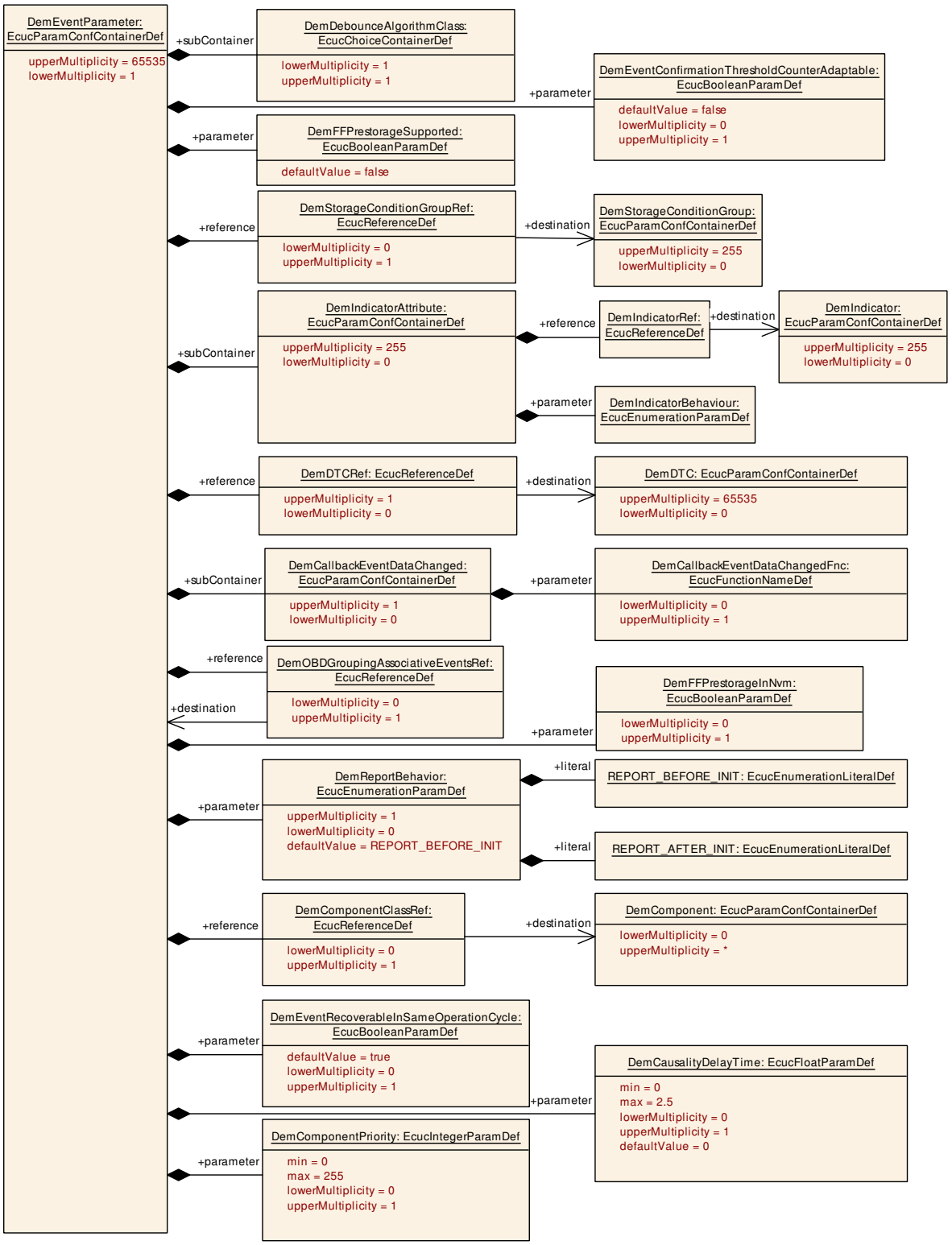


Figure 10.8: DemEventParameter configuration overview (part II)

10.2.2.6 DemMultiEventTriggering

[ECUC_Dem_00944] Definition of EcucParamConfContainerDef DemMultiEventTriggering ⌈

Container Name	DemMultiEventTriggering
Parent Container	DemConfigSet
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	Configures an event that will trigger other events whenever the event is reported.
Multiplicity	0..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemMultiEventTriggeringMasterEventRef	1	[ECUC_Dem_00945]
DemMultiEventTriggeringSlaveEventRef	1..255	[ECUC_Dem_00946]

No Included Containers

⌋

[ECUC_Dem_00945] Definition of EcucReferenceDef DemMultiEventTriggering MasterEventRef ⌈

Parameter Name	DemMultiEventTriggeringMasterEventRef
Parent Container	DemMultiEventTriggering
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	Reference to the event that will trigger other events upon reception of this event.
Multiplicity	1
Type	Reference to DemEventParameter
Pre-Compile only	true

⌋

[ECUC_Dem_00946] Definition of EcucReferenceDef DemMultiEventTriggering SlaveEventRef ⌈

Parameter Name	DemMultiEventTriggeringSlaveEventRef
Parent Container	DemMultiEventTriggering
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	Reference to the event that is triggered upon triggering the master event.
Multiplicity	1..255
Type	Reference to DemEventParameter
Pre-Compile only	true

⌋

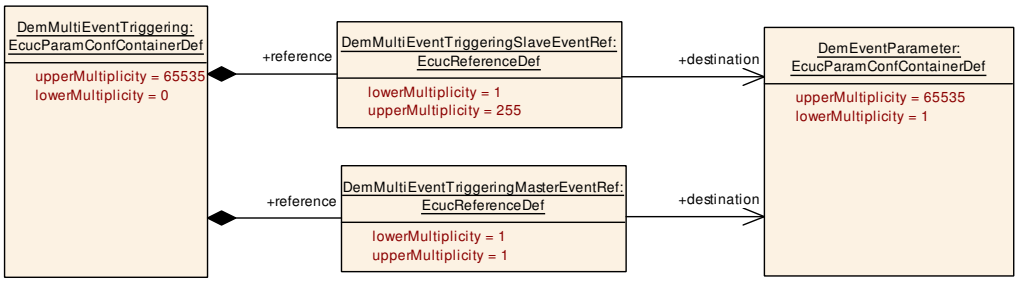


Figure 10.9: DemMultiEventTriggering configuration overview

10.2.2.7 DemComponent

[ECUC_Dem_00904] Definition of EcucParamConfContainerDef DemComponent ⌈

Container Name	DemComponent
Parent Container	DemConfigSet
Upstream mapping defined	–
Description	This container configures the monitored components and system dependencies.
Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemComponentFailedCallbackFnc	0..1	[ECUC_Dem_00906]
DemComponentFailedCallbackUsePort	0..1	[ECUC_Dem_00947]
DemComponentId	1	[ECUC_Dem_00930]
DemComponentIgnoresPriority	0..1	[ECUC_Dem_00905]
DemImmediateChildComponentRef	0..*	[ECUC_Dem_00907]

No Included Containers

⌋

[ECUC_Dem_00906] Definition of EcucFunctionNameDef DemComponentFailedCallbackFnc ⌈

Parameter Name	DemComponentFailedCallbackFnc
Parent Container	DemComponent
Upstream mapping defined	–
Description	Specifies the function to be called on component failed status changes.
Multiplicity	0..1
Type	EcucFunctionNameDef
Default value	–

▽

△

Regular Expression	–
Pre-Compile only	true

⌋

[ECUC_Dem_00947] Definition of EcucFunctionNameDef DemComponentFailedCallbackUsePort ⌈

Parameter Name	DemComponentFailedCallbackUsePort
Parent Container	DemComponent
Upstream mapping defined	–
Description	Specifies whether CBStatusComp_{ComponentName} for component change notifications shall be provided.
Multiplicity	0..1
Type	EcucFunctionNameDef
Default value	false
Regular Expression	–
Pre-Compile only	true

⌋

[ECUC_Dem_00930] Definition of EcucIntegerParamDef DemComponentId ⌈

Parameter Name	DemComponentId
Parent Container	DemComponent
Upstream mapping defined	–
Description	Unique identifier of a DemComponent. Component Id should be configured in sequential order beginning with 1 and no gaps in between. 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
Multiplicity	1
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)
Range	1 .. 65535
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00905] Definition of EcucBooleanParamDef DemComponentIgnoresPriority ⌈

Parameter Name	DemComponentIgnoresPriority
Parent Container	DemComponent
Upstream mapping defined	–
Description	This configuration switch defines, whether the priority of events at this component shall be ignored.
Multiplicity	0..1
Type	EcucBooleanParamDef
Default value	false

▽

△

Pre-Compile only	true

⌋

[ECUC_Dem_00907] Definition of EcucReferenceDef DemImmediateChildComponentRef ⌈

Parameter Name	DemImmediateChildComponentRef
Parent Container	DemComponent
Upstream mapping defined	–
Description	Reference to all immediate children of the current component.
Multiplicity	0..*
Type	Reference to DemComponent
Pre-Compile only	true

⌋

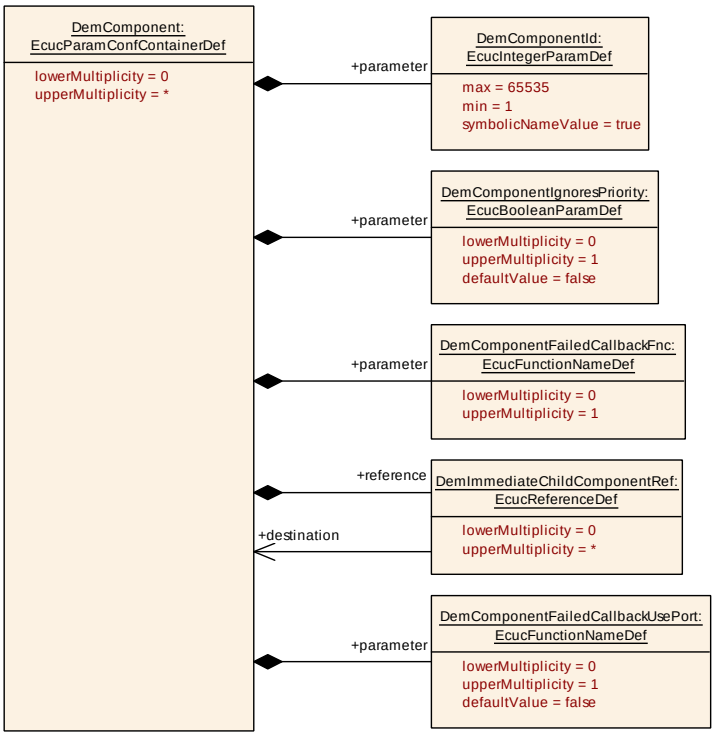


Figure 10.10: DemComponent configuration overview

10.2.2.8 DemDTC

[ECUC_Dem_00886] Definition of EcucParamConfContainerDef DemDTC ⌈

Container Name	DemDTC
Parent Container	DemConfigSet
Upstream mapping defined	–
Description	This container contains the configuration (parameters) for DemUdsDTC.
Multiplicity	0..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
DemDTCFunctionalUnit	0..1	[ECUC_Dem_00643]
DemDtcOBDRelevance	0..1	[ECUC_Dem_00994]
DemDTCSeverity	0..1	[ECUC_Dem_00645]
DemDtcValue	0..1	[ECUC_Dem_00887]
DemNvStorageStrategy	0..1	[ECUC_Dem_00127]
DemWWHOBDDTCClass	0..1	[ECUC_Dem_00912]
DemDTCAttributesRef	1	[ECUC_Dem_00642]
DemObdDTCRef	0..1	[ECUC_Dem_00889]

No Included Containers

⌋

[ECUC_Dem_00643] Definition of EcucIntegerParamDef DemDTCFunctionalUnit

⌈

Parameter Name	DemDTCFunctionalUnit	
Parent Container	DemDTC	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	DTCFuncitonalUnit is a 1-byte value which identifies the corresponding basic vehicle / system function which reports the DTC. This parameter is necessary for the report of severity information. If this parameter is configured for no DTC, the Dem provides no DTC functional unit information.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00994] Definition of EcucBooleanParamDef DemDtcOBDRelevance

⌈

Parameter Name	DemDtcOBDRelevance
Parent Container	DemDTC
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]

▽

△

Description	Defines is this DTC is relevant for OBD. OBD relevant DTCs are reported in the OBD related APIs and it is possible to configure the calculation of further OBD related data such as readiness groups, IUMPR, etc. If set to false, the DTC is not OBD relevant If set to true, the DTC is OBD relevant.	
Multiplicity	0..1	
Type	EcucBooleanParamDef	
Default value	–	
Post-Build Multiplicity	-	Selectable
Post-Build Value	-	Selectable

⌋

[ECUC_Dem_00645] Definition of EcucEnumerationParamDef DemDTCSeverity

⌈

Parameter Name	DemDTCSeverity	
Parent Container	DemDTC	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	DTC severity according to ISO 14229-1. This parameter depends on the automotive manufacturer. If it is not configured, the value is counted as 'no severity'. If this parameter is configured for no DTC, the Dem provides no DTC severity information.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	DEM_SEVERITY_CHECK_AT_NEXT_HALT	Check at next halt
	DEM_SEVERITY_CHECK_IMMEDIATELY	Check immediately
	DEM_SEVERITY_MAINTENANCE_ONLY	Maintenance required
	DEM_SEVERITY_NO_SEVERITY	No severity information available
Default value	DEM_SEVERITY_NO_SEVERITY	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00887] Definition of EcucIntegerParamDef DemDtcValue ⌈

Parameter Name	DemDtcValue	
Parent Container	DemDTC	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	3 Byte value of the UDS Diagnostic Trouble Code (DTC). The value is unique within a fault memory. The ranges: 0x000000-0x0000FF and 0xFFFFFF are reserved by ISO 14229-1.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	256 .. 16777214	
Default value	–	

▽

△

Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00127] Definition of EcucEnumerationParamDef DemNvStorageStrategy ⌈

Parameter Name	DemNvStorageStrategy	
Parent Container	DemDTC	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This parameter defines when the event related data of the DTC is stored in the NVRAM.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	DURING_SHUTDOWN	Store during shutdown
	IMMEDIATE_AT_FIRST_OCCURRENCE	Store at first occurrence of DemEventMemoryEntryStorageTrigger trigger and during shutdown
Default value	DURING_SHUTDOWN	
Pre-Compile only	true	

⌋

[ECUC_Dem_00912] Definition of EcucEnumerationParamDef DemWWHOBDDTCClass ⌈

Parameter Name	DemWWHOBDDTCClass	
Parent Container	DemDTC	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	DTC Class according to ISO 14229-1 [2013 version]. This parameter depends on the automotive manufacturer. If it is not configured, the value is marked as 'unclassified'. If this parameter is configured for no DTC, the Dem provides no DTC WWHOBD class information.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	DEM_DTC_WWHOBD_CLASS_A	Class A
	DEM_DTC_WWHOBD_CLASS_B1	Class B1
	DEM_DTC_WWHOBD_CLASS_B2	Class B2
	DEM_DTC_WWHOBD_CLASS_C	Class C
	DEM_DTC_WWHOBD_CLASS_NOCLASS	No Class information
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00642] Definition of EcucReferenceDef DemDTCAttributesRef ⌈

Parameter Name	DemDTCAttributesRef
Parent Container	DemDTC
Upstream mapping defined	–
Description	This parameter defines the DTC Attributes associated with the DemDTC.
Multiplicity	1
Type	Reference to DemDTCAttributes
Pre-Compile only	true

⌋

[ECUC_Dem_00889] Definition of EcucReferenceDef DemObdDTCRef ⌈

Parameter Name	DemObdDTCRef
Parent Container	DemDTC
Upstream mapping defined	–
Description	This parameter defines the OBD DTC configuration associated with the DemDTC. It is allowed to have events without a OBD DTC.
Multiplicity	0..1
Type	Reference to DemObdDTC
Pre-Compile only	true

⌋

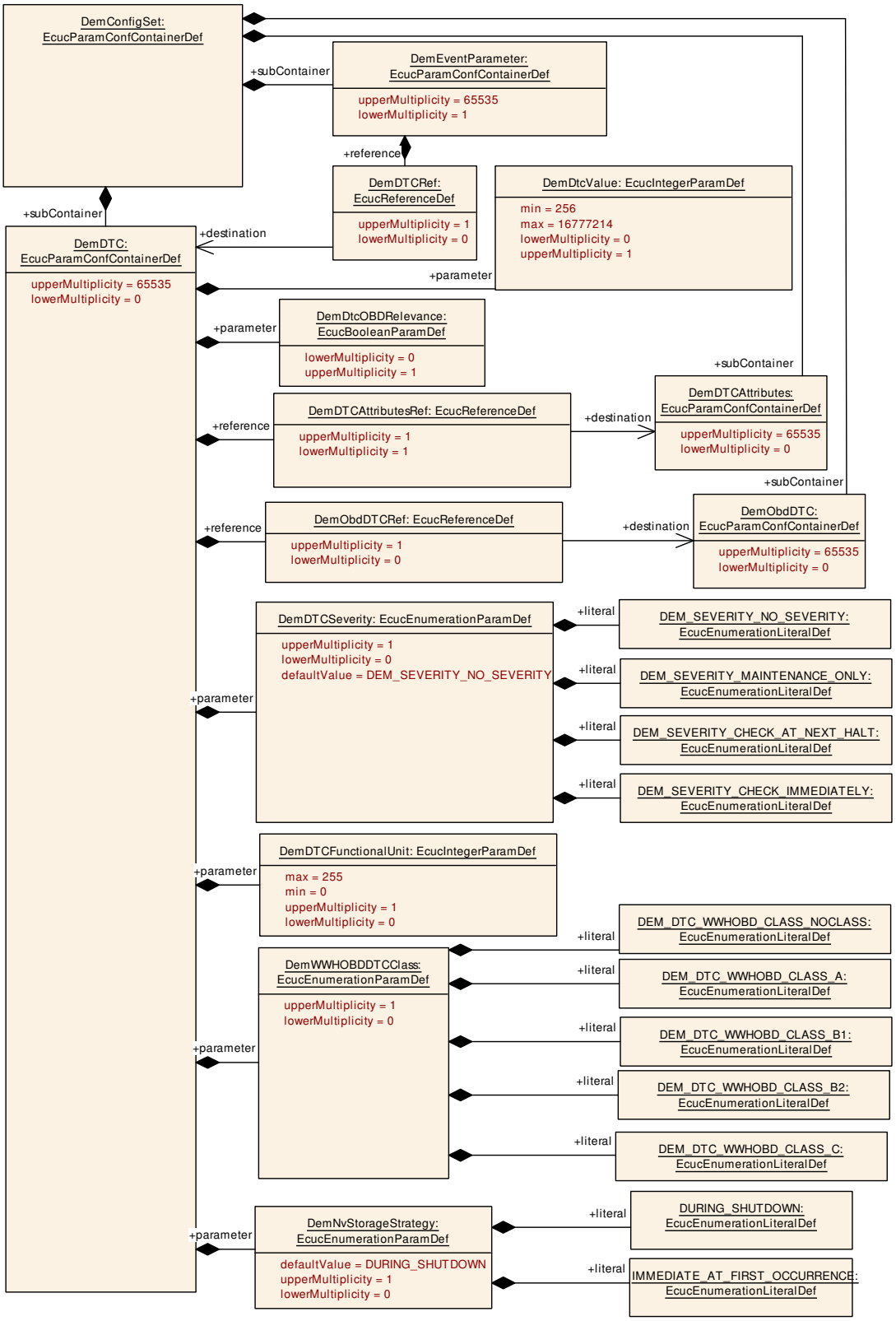


Figure 10.11: DemDTC configuration overview

10.2.2.9 DemGroupOfDTC

[ECUC_Dem_00679] Definition of EcucParamConfContainerDef DemGroupOfDTC ⌈

Container Name	DemGroupOfDTC
Parent Container	DemPrimaryMemory, DemUserDefinedMemory
Upstream mapping defined	–
Description	This container contains the configuration (parameters) for DTC groups.
Multiplicity	0..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemGroupDTCs	1	[ECUC_Dem_00678]

No Included Containers

⌋

[ECUC_Dem_00678] Definition of EcucIntegerParamDef DemGroupDTCs ⌈

Parameter Name	DemGroupDTCs	
Parent Container	DemGroupOfDTC	
Upstream mapping defined	–	
Description	DTC values of the selected group of DTC (Range: 3 byte, 0xFFFFFF is reserved for 'all DTCs', according to ISO14229-1 Annex D.1) The DTC group 'all DTCs' is always available and will not be configured. The following ranges are reserved by ISO 14229-1 : 0x000000 to 0x0000ff and 0xffff00 to 0xffffff.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	256 .. 16776959	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

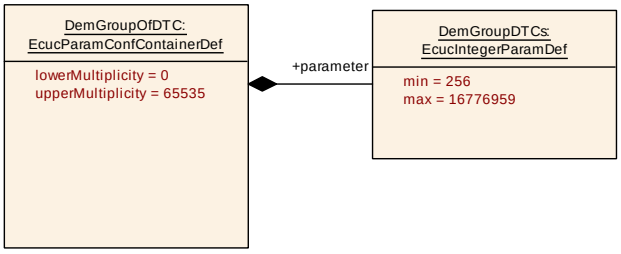


Figure 10.12: DemGroupOfDTC configuration overview

10.2.2.10 DemOperationCycle

[ECUC_Dem_00701] Definition of EcucParamConfContainerDef DemOperationCycle ⌈

Container Name	DemOperationCycle
Parent Container	DemGeneral
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This container holds all parameters that are relevant to configure an operation cycle.
Multiplicity	1..256
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemOperationCycleId	1	[ECUC_Dem_00898]
DemLeadingCycleRef	0..1	[ECUC_Dem_00919]

No Included Containers

⌋

[ECUC_Dem_00898] Definition of EcucIntegerParamDef DemOperationCycleId ⌈

Parameter Name	DemOperationCycleId
Parent Container	DemOperationCycle
Upstream mapping defined	–
Description	This parameter's value is used, together with the aggregating container, to define a symbolic name of the operation cycle.
Multiplicity	1
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)
Range	0 .. 255
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00919] Definition of EcucReferenceDef DemLeadingCycleRef ⌈

Parameter Name	DemLeadingCycleRef
Parent Container	DemOperationCycle
Upstream mapping defined	–
Description	Defines the operation cycle, which is relevant for processing this operation cycle.
Multiplicity	0..1
Type	Symbolic name reference to DemOperationCycle
Pre-Compile only	true

⌋

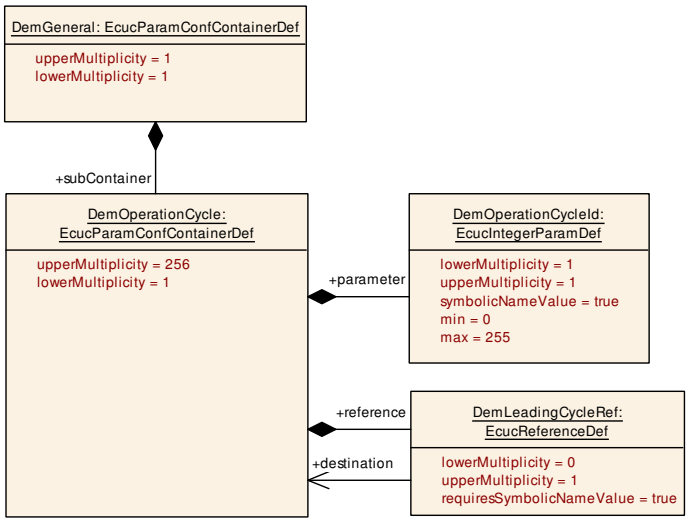


Figure 10.13: DemOperationCycle configuration overview

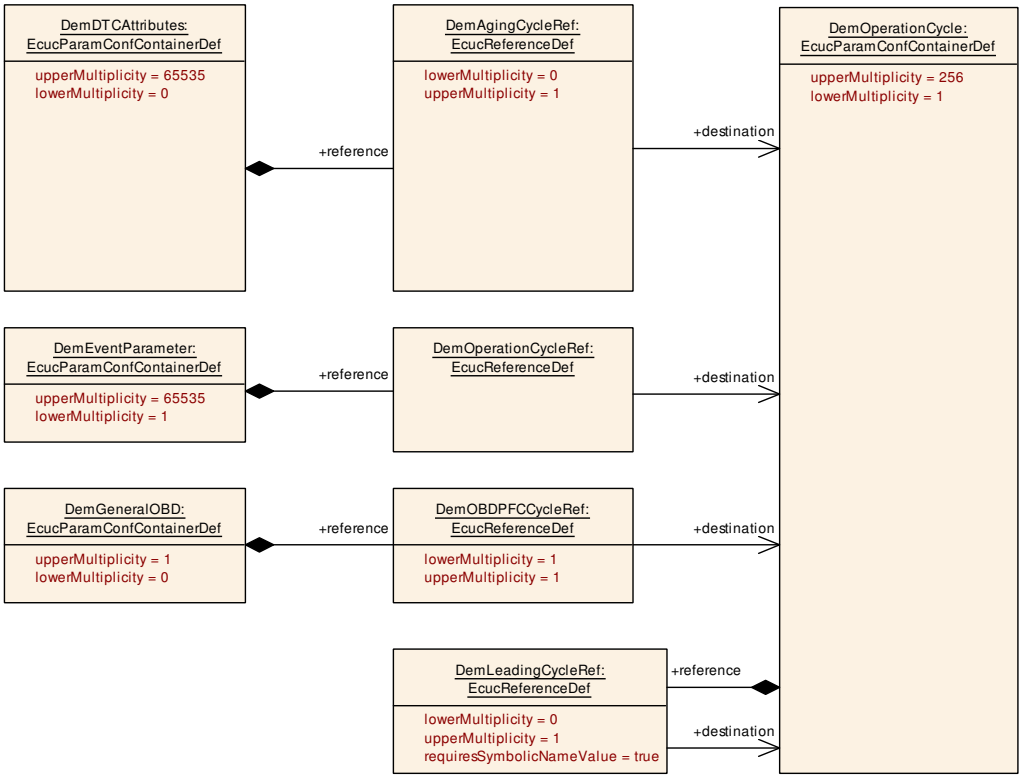


Figure 10.14: DemOperationCycle usage overview

10.2.2.11 DemIndicator

[ECUC_Dem_00680] Definition of EcucParamConfContainerDef DemIndicator ⌈

Container Name	DemIndicator
Parent Container	DemEventMemorySet
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This container contains the configuration (parameters) for Indicators.
Multiplicity	0..255
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
DemIndicatorID	1	[ECUC_Dem_00683]

No Included Containers

⌋

[ECUC_Dem_00683] Definition of EcucIntegerParamDef DemIndicatorID ⌈

Parameter Name	DemIndicatorID	
Parent Container	DemIndicator	
Upstream mapping defined	–	
Description	Unique identifier of an indicator.	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

10.2.2.12 DemIndicatorAttribute

[ECUC_Dem_00681] Definition of EcucParamConfContainerDef DemIndicatorAttribute ⌈

Container Name	DemIndicatorAttribute
Parent Container	DemEventParameter
Upstream mapping defined	–
Description	This container contains the event specific configuration of Indicators.
Multiplicity	0..255
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemIndicatorBehaviour	1	[ECUC_Dem_00682]
DemIndicatorFailureCycleCounterThreshold	0..1	[ECUC_Dem_00750]
DemIndicatorHealingCycleCounterThreshold	1	[ECUC_Dem_00748]
DemIndicatorRef	1	[ECUC_Dem_00687]

No Included Containers

⌋

[ECUC_Dem_00682] Definition of EcucEnumerationParamDef DemIndicatorBehaviour ⌈

Parameter Name	DemIndicatorBehaviour	
Parent Container	DemIndicatorAttribute	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Behaviour of the linked indicator	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_INDICATOR_BLINKING	The indicator blinks when the event has status FAILED Not relevant with J1939.
	DEM_INDICATOR_BLINK_CONT	The indicator is active and blinks when the event has status FAILED Not relevant with J1939.
	DEM_INDICATOR_ CONTINUOUS	The indicator is active when the even has status FAILED
	DEM_INDICATOR_FAST_FLASH	Flash Indicator Lamp should be set to 'Fast Flash'
	DEM_INDICATOR_SLOW_ FLASH	Flash Indicator Lamp should be set to 'Slow Flash'
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00750] Definition of EcucIntegerParamDef DemIndicatorFailureCycleCounterThreshold ⌈

Parameter Name	DemIndicatorFailureCycleCounterThreshold	
Parent Container	DemIndicatorAttribute	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Defines the number of failure cycles for the WarningIndicatorOnCriteria.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	1 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00748] Definition of EcucIntegerParamDef DemIndicatorHealingCycleCounterThreshold ⌈

Parameter Name	DemIndicatorHealingCycleCounterThreshold	
Parent Container	DemIndicatorAttribute	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Defines the number of healing cycles for the WarningIndicatorOffCriteria.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00687] Definition of EcucReferenceDef DemIndicatorRef ⌈

Parameter Name	DemIndicatorRef
Parent Container	DemIndicatorAttribute
Upstream mapping defined	–
Description	Reference to the used indicator.
Multiplicity	1
Type	Reference to DemIndicator
Pre-Compile only	true

⌋

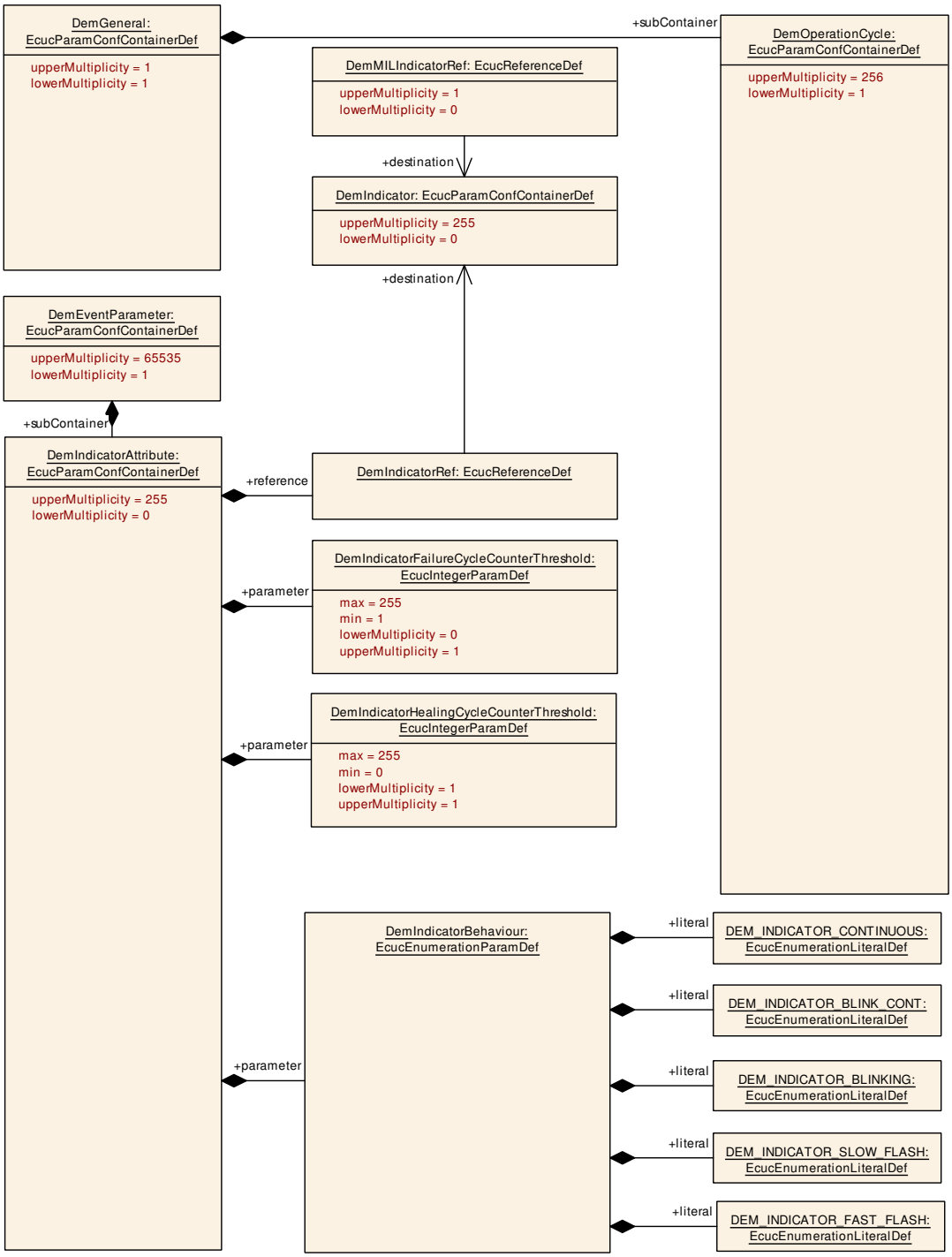


Figure 10.15: DemIndicator configuration overview

10.2.2.13 DemNvRamBlockId

[ECUC_Dem_00696] Definition of EcucParamConfContainerDef DemNvRamBlockId ⌈

Container Name	DemNvRamBlockId
Parent Container	DemGeneral
Upstream mapping defined	–
Description	This container contains the configuration (parameters) for a non-volatile memory block, which is used from the Dem. If no permanent storage of event memory entries is required, no block needs to be configured. The number of blocks which are necessary depends on the implementation and configuration (e.g. number of used event memories) of the Dem module.
Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
DemNvRamBlockIdRef	1	[ECUC_Dem_00697]

No Included Containers

⌋

[ECUC_Dem_00697] Definition of EcucReferenceDef DemNvRamBlockIdRef ⌈

Parameter Name	DemNvRamBlockIdRef
Parent Container	DemNvRamBlockId
Upstream mapping defined	–
Description	This reference contains the link to a non-volatile memory block. For post build time configurations worst case scenario shall be used.
Multiplicity	1
Type	Symbolic name reference to NvMBlockDescriptor
Pre-Compile only	true

⌋

10.2.3 OBD

10.2.3.1 DemGeneralOBD

[ECUC_Dem_00756] Definition of EcucParamConfContainerDef DemGeneralOBD ⌈

Container Name	DemGeneralOBD
Parent Container	DemGeneral
Upstream mapping defined	–
Description	This container contains the general OBD-specific configuration (parameters) of the Dem module.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemFreezeMANWhenGeneralDenominatorInhibited	1	[ECUC_Dem_00995]
DemOBDCentralizedPID21Handling	1	[ECUC_Dem_00794]
DemOBDCentralizedPID31Handling	1	[ECUC_Dem_00879]
DemOBDCompliancy	1	[ECUC_Dem_00795]
DemOBDDelayedDCYConfirmedAndMIL	0..1	[ECUC_Dem_00917]
DemOBDEngineType	0..1	[ECUC_Dem_00900]
DemOBDEventDisplacement	1	[ECUC_Dem_00796]
DemSupportedObdUdsDtcSeparation	0..1	[ECUC_Dem_00989]
DemOBDDrivingCycleRef	1	[ECUC_Dem_00923]
DemOBDInputAcceleratorPedalInformation	0..1	[ECUC_Dem_00763]
DemOBDInputAmbientPressure	0..1	[ECUC_Dem_00762]
DemOBDInputAmbientTemperature	0..1	[ECUC_Dem_00761]
DemOBDInputDistanceInformation	0..1	[ECUC_Dem_00759]
DemOBDInputEngineSpeed	0..1	[ECUC_Dem_00757]
DemOBDInputEngineTemperature	0..1	[ECUC_Dem_00772]
DemOBDInputProgrammingEvent	0..1	[ECUC_Dem_00760]
DemOBDInputVehicleSpeed	0..1	[ECUC_Dem_00758]
DemOBDPFCCycleRef	1	[ECUC_Dem_00920]
DemOBDTimeSinceEngineStart	0..1	[ECUC_Dem_00827]
DemOBDWarmUpCycleRef	1	[ECUC_Dem_00922]

Included Containers		
Container Name	**Multiplicity**	**Description**
DemCallbackOBDDTCStatus Changed	0..*	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. In case there is a DemCallbackDTCStatusChangedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackDTCStatusChangedFnc, the Dem will have an R-Port requiring the interface Callback DTCStatusChanged. Status change notifications are supported for DTCs in primary memory only.

⌋

[ECUC_Dem_00995] Definition of EcucBooleanParamDef DemFreezeMANWhenGeneralDenominatorInhibited ⌈

Parameter Name	DemFreezeMANWhenGeneralDenominatorInhibited
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Controls if the numerator of the monitor activity ratios will not be incremented at the end of the driving cycle if the general denominator is inhibited. If set to true, this effectively means that none of the monitor activity numerators is incremented in case the general denominator is inhibted. As result the entire SMAD calculation is froozen.
Multiplicity	1

▽

△

Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_Dem_00794] Definition of EcucBooleanParamDef DemOBDCentralized PID21Handling ⌈

Parameter Name	DemOBDCentralizedPID21Handling
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Switch to enable the centralized handling of PID $21. true: centralized handling of PID $21 enabled false: centralized handling of PID $21 disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00879] Definition of EcucBooleanParamDef DemOBDCentralized PID31Handling ⌈

Parameter Name	DemOBDCentralizedPID31Handling
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Switch to enable the centralized handling of PID $31. true: centralized handling of PID $31 enabled false: centralized handling of PID $31 disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00795] Definition of EcucIntegerParamDef DemOBDCompliancy ⌈

Parameter Name	DemOBDCompliancy
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	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. Notice as well J1979 or the "DiagnosticReadiness 1" DM05 message of J1939-73
Multiplicity	1
Type	EcucIntegerParamDef

▽

△

Range	0 .. 255	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00917] Definition of EcucBooleanParamDef DemOBDDelayedDCY-ConfirmedAndMIL ⌈

Parameter Name	DemOBDDelayedDCYConfirmedAndMIL
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Controls the delayed calculation of the confirmed status for the OBD driving cycle.
Multiplicity	0..1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_Dem_00900] Definition of EcucEnumerationParamDef DemOBDEngine Type ⌈

Parameter Name	DemOBDEngineType	
Parent Container	DemGeneralOBD	
Upstream mapping defined	–	
Description	Switch to provide either Gasoline or Diesel parameters.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	DEM_IGNITION_ COMPRESSION	Diesel engine type
	DEM_IGNITION_SPARK	Gasoline engine type
Pre-Compile only	true	

⌋

[ECUC_Dem_00796] Definition of EcucBooleanParamDef DemOBDEventDisplacement ⌈

Parameter Name	DemOBDEventDisplacement
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Activate/Deactivate a different displacement behavior for OBD events. OBD events with special Conditions (e.g. Pending, MIL_On...) shall not be displaced.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00989] Definition of EcucBooleanParamDef DemSupportedObdUdsDtcSeparation ⌈

Parameter Name	DemSupportedObdUdsDtcSeparation	
Parent Container	DemGeneralOBD	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	If this parameter is set to true, the OBD UDS DTC separation is enabled. If this parameter is set to false, the OBD UDS DTC separation is disabled.	
Multiplicity	0..1	
Type	EcucBooleanParamDef	
Default value	false	
Post-Build Multiplicity	-	Selectable
Post-Build Value	-	Selectable

⌋

[ECUC_Dem_00923] Definition of EcucReferenceDef DemOBDDrivingCycleRef ⌈

Parameter Name	DemOBDDrivingCycleRef
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Defines the operationCycle which denotes the OBD driving cycle.
Multiplicity	1
Type	Reference to DemOperationCycle
Pre-Compile only	true

⌋

[ECUC_Dem_00763] Definition of EcucChoiceReferenceDef DemOBDInputAcceleratorPedalInformation ⌈

Parameter Name	DemOBDInputAcceleratorPedalInformation
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Input variable for the accelerator padal information, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.
Multiplicity	0..1
Type	Choice reference to [DemExternalCSDataElementClass, DemExternalSRData ElementClass]
Pre-Compile only	true

⌋

[ECUC_Dem_00762] Definition of EcucChoiceReferenceDef DemOBDInputAmbientPressure ⌈

Parameter Name	DemOBDInputAmbientPressure
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Input variable for the ambient pressure, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.

▽

△

Multiplicity	0..1
Type	Choice reference to [DemExternalCSDataElementClass, DemExternalSRData ElementClass]
Pre-Compile only	true

⌋

[ECUC_Dem_00761] Definition of EcucChoiceReferenceDef DemOBDInputAmbientTemperature ⌈

Parameter Name	DemOBDInputAmbientTemperature
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Input variable for the ambient temperature, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.
Multiplicity	0..1
Type	Choice reference to [DemExternalCSDataElementClass, DemExternalSRData ElementClass]
Pre-Compile only	true

⌋

[ECUC_Dem_00759] Definition of EcucChoiceReferenceDef DemOBDInputDistanceInformation ⌈

Parameter Name	DemOBDInputDistanceInformation
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Input variable for the distance information, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.
Multiplicity	0..1
Type	Choice reference to [DemExternalCSDataElementClass, DemExternalSRData ElementClass]
Pre-Compile only	true

⌋

[ECUC_Dem_00757] Definition of EcucChoiceReferenceDef DemOBDInputEngineSpeed ⌈

Parameter Name	DemOBDInputEngineSpeed
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Input variable for the engine speed, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.
Multiplicity	0..1
Type	Choice reference to [DemExternalCSDataElementClass, DemExternalSRData ElementClass]
Pre-Compile only	true

⌋

[ECUC_Dem_00772] Definition of EcucChoiceReferenceDef DemOBDInputEngineTemperature ⌈

Parameter Name	DemOBDInputEngineTemperature
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Input variable for the engine temperature, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.
Multiplicity	0..1
Type	Choice reference to [DemExternalCSDataElementClass, DemExternalSRDataElementClass]
Pre-Compile only	true

⌋

[ECUC_Dem_00760] Definition of EcucChoiceReferenceDef DemOBDInputProgrammingEvent ⌈

Parameter Name	DemOBDInputProgrammingEvent
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Input variable for the programming event, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.
Multiplicity	0..1
Type	Choice reference to [DemExternalCSDataElementClass, DemExternalSRDataElementClass]
Pre-Compile only	true

⌋

[ECUC_Dem_00758] Definition of EcucChoiceReferenceDef DemOBDInputVehicleSpeed ⌈

Parameter Name	DemOBDInputVehicleSpeed
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Input variable for the vehicle speed, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.
Multiplicity	0..1
Type	Choice reference to [DemExternalCSDataElementClass, DemExternalSRDataElementClass]
Pre-Compile only	true

⌋

[ECUC_Dem_00920] Definition of EcucReferenceDef DemOBDPFCCycleRef ⌈

Parameter Name	DemOBDPFCCycleRef
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Defines the operationCycle, which is relevant for processing the OBDPFCCycle.
Multiplicity	1

▽

△

Type	Reference to DemOperationCycle
Pre-Compile only	true

⌋

[ECUC_Dem_00827] Definition of EcucChoiceReferenceDef DemOBDTimeSinceEngineStart ⌈

Parameter Name	DemOBDTimeSinceEngineStart
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Input variable for the Time Since Engine Start information, which is assigned to a specific data element.
Multiplicity	0..1
Type	Choice reference to [DemExternalCSDataElementClass, DemExternalSRDataElementClass]
Pre-Compile only	true

⌋

[ECUC_Dem_00922] Definition of EcucReferenceDef DemOBDWarmUpCycleRef ⌈

Parameter Name	DemOBDWarmUpCycleRef
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	Defines the operationCycle which denotes the OBD warm-up cycle.
Multiplicity	1
Type	Reference to DemOperationCycle
Pre-Compile only	true

⌋

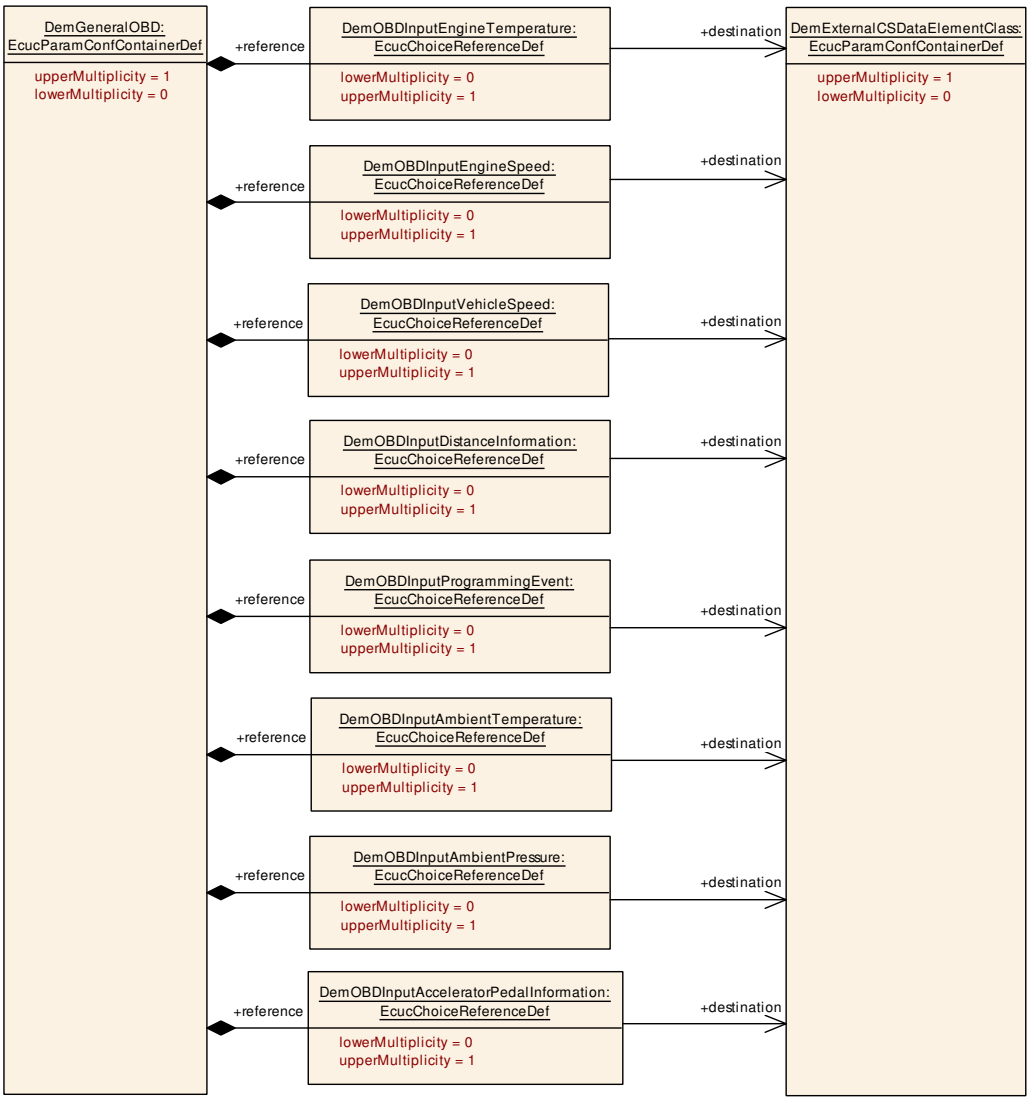


Figure 10.16: DemGeneralOBD configuration overview (part I) for CS data element

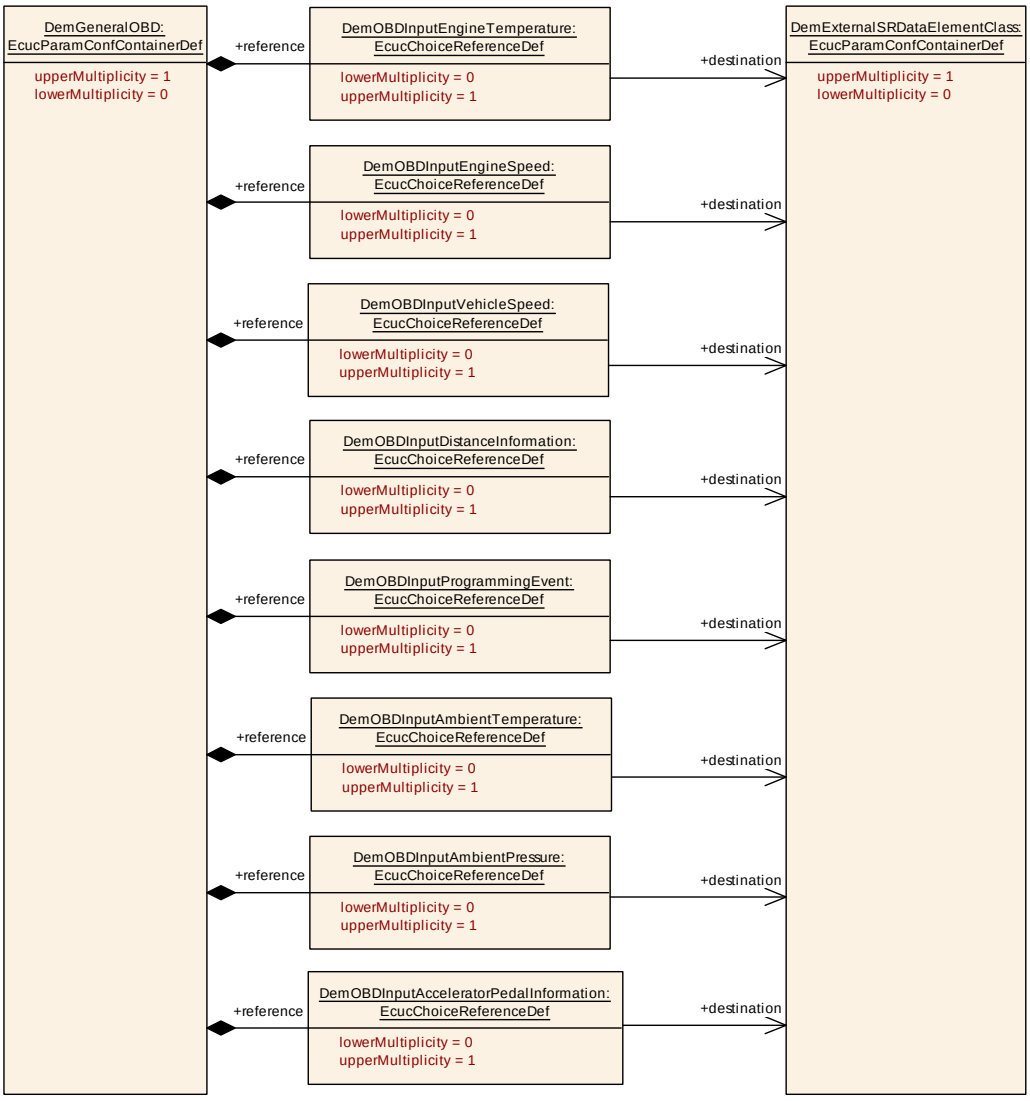


Figure 10.17: DemGeneralOBD configuration overview (part I) for SR data element

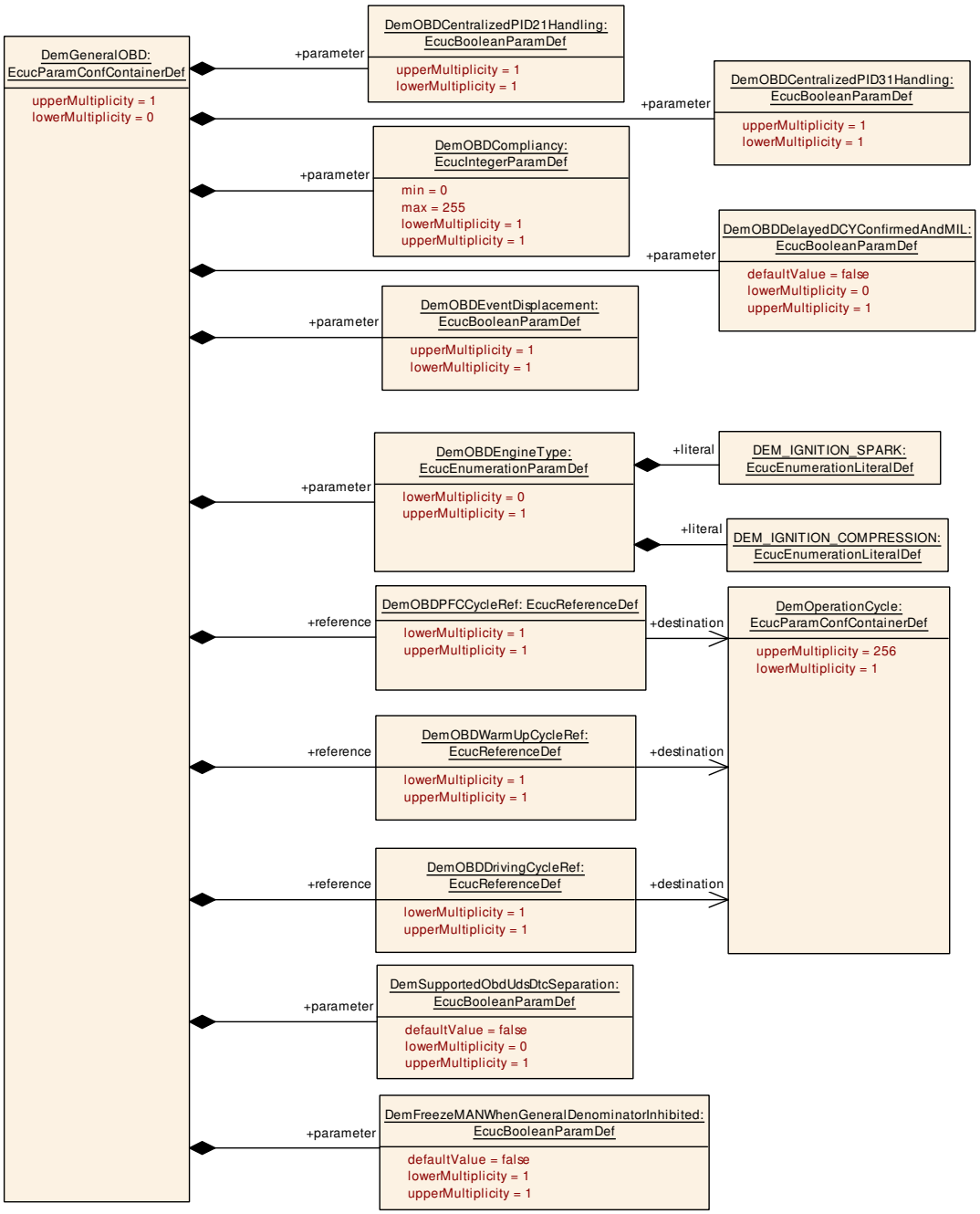


Figure 10.18: DemGeneralOBD configuration overview (part II)

10.2.3.2 DemObdDTC

[ECUC_Dem_00884] Definition of EcucParamConfContainerDef DemObdDTC ⌈

Container Name	DemObdDTC
Parent Container	DemConfigSet
Upstream mapping defined	–
Description	This container contains the configuration (parameters) for DemObdDTC.
Multiplicity	0..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
DemConsiderPtoStatus	1	[ECUC_Dem_00602]
DemDtcValue	0..1	[ECUC_Dem_00885]
DemDtcValue3Byte	0..1	[ECUC_Dem_00988]
DemEventOBDReadinessGroup	0..1	[ECUC_Dem_00755]
DemJ1939DTCValue	0..1	[ECUC_Dem_00892]

No Included Containers

⌋

[ECUC_Dem_00602] Definition of EcucBooleanParamDef DemConsiderPtoStatus ⌈

Parameter Name	DemConsiderPtoStatus	
Parent Container	DemObdDTC	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This parameter is TRUE, when the event is affected by the Dem PTO handling.	
Multiplicity	1	
Type	EcucBooleanParamDef	
Default value	false	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00885] Definition of EcucIntegerParamDef DemDtcValue ⌈

Parameter Name	DemDtcValue	
Parent Container	DemObdDTC	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Unique Diagnostic Trouble Code value for OBD	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	1 .. 65535	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00988] Definition of EcucIntegerParamDef DemDtcValue3Byte ⌈

Parameter Name	DemDtcValue3Byte	
Parent Container	DemObdDTC	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	3 Byte OBD DTC value that is used for the UDS Diagnostic Trouble Code (DTC) if UDS and OBD DTC handling is separated. The value is defining a DTC based on SAE J2012.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	256 .. 16777214	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00755] Definition of EcucEnumerationParamDef DemEventOBDReadinessGroup ⌈

Parameter Name	DemEventOBDReadinessGroup	
Parent Container	DemObdDTC	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This parameter specifies the Event OBD Readiness group as defined by SAE J1979-DA for PID $01 and PID $41 computation. This parameter is only applicable for emission-related ECUs.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	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_CMPRCMPT	Comprehensive component - spark, compr.
	DEM_OBD_RDY_CSER	Cold Start Emission Reduction Strategy Monitoring - spark, compr.
	DEM_OBD_RDY_DOR	Direct Ozone Reduction (DOR) System Monitoring - spark, compr.
	DEM_OBD_RDY_ECS	Engine Cooling System Monitoring - spark, compr.
	DEM_OBD_RDY_EGR	EGR system - spark, compr.
	DEM_OBD_RDY_EGSENS	Exhaust Gas Sensor - compr.
	DEM_OBD_RDY_EVAP	Evaporative system - spark
	DEM_OBD_RDY_FLSYS	Fuel system - spark, compr.
	DEM_OBD_RDY_FLSYS_NONCONT	Non Contious Fuel system - spark, compr
	DEM_OBD_RDY_GPF	Gasoline Particulate Filter - spark.
	DEM_OBD_RDY_HCCAT	Non-Methan HC Catalyst - compr.
	DEM_OBD_RDY_HTCAT	Heated catalyst - spark
	DEM_OBD_RDY_MISF	Misfire - spark, compr.
	DEM_OBD_RDY_NONE	None - spark, compr.
	DEM_OBD_RDY_NOXADSORB	NOx Adsorber Monitoring - 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_OTHER	Other Emission Control or Source System Monitoring - spark, compr.
	DEM_OBD_RDY_PCV	Positive/Crankcase Ventilation (PCV) System Monitoring - spark, compr.
	DEM_OBD_RDY_PMFLT	Particle Matters Filter - compr.
	DEM_OBD_RDY_SECAIR	Secondary air system - spark
	DEM_OBD_RDY_VVT	Variable Valve Timing, Lift, and/or Control (VVT) System Monitoring - spark, compr.
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00892] Definition of EcucIntegerParamDef DemJ1939DTCValue ⌈

Parameter Name	DemJ1939DTCValue	
Parent Container	DemObdDTC	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Unique Diagnostic Trouble Code value for J1939 (consisting of SPN and FMI).	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	1 .. 16777214	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

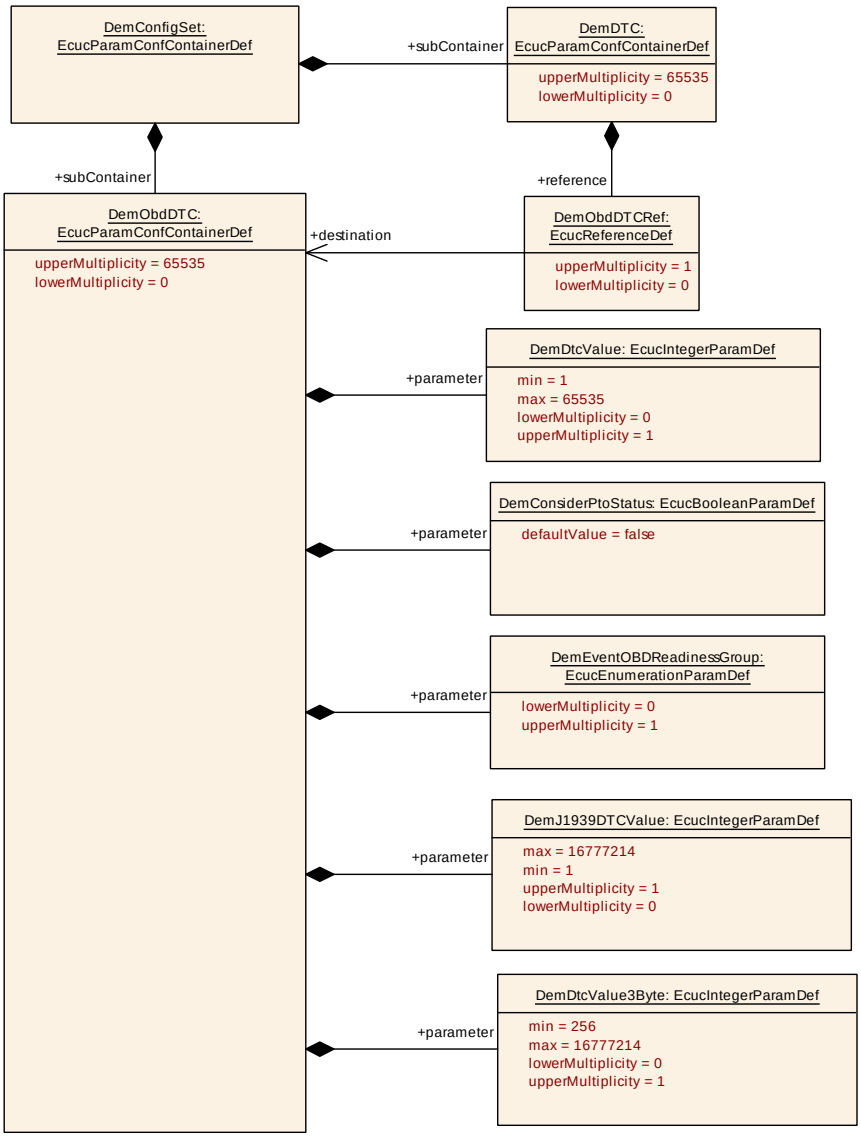


Figure 10.19: DemObdDTC configuration overview

10.2.3.3 DemRatio

[ECUC_Dem_00734] Definition of EcucParamConfContainerDef DemRatio ⌈

Container Name	DemRatio
Parent Container	DemGeneral
Upstream mapping defined	[24, CP_TPS_BSWModuleDescriptionTemplate], [23, CP_TPS_DiagnosticExtract Template]
Description	This container contains the OBD-specific in-use-monitor performance ratio configuration. It is related to a specific event, a FID, and an IUMPR group.
Multiplicity	0..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemIUMPRDenGroup	1	[ECUC_Dem_00838]
DemIUMPRGroup	1	[ECUC_Dem_00737]
DemRatioId	1	[ECUC_Dem_00787]
DemRatioKind	1	[ECUC_Dem_00741]
DemDiagnosticEventRef	1	[ECUC_Dem_00735]
DemFunctionIdRef	0..*	[ECUC_Dem_00736]

No Included Containers

⌋

[ECUC_Dem_00838] Definition of EcucEnumerationParamDef DemIUMPRDenGroup ⌈

Parameter Name	DemIUMPRDenGroup	
Parent Container	DemRatio	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This parameter specifies the assigned denominator type	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_IUMPR_DEN_500MILL	Condition based on definition of 500miles conditions as defined for OBD2.
	DEM_IUMPR_DEN_COLDSTART	Condition based on definition of "cold start" as defined for EU5+
	DEM_IUMPR_DEN_CSERS	Cold start emission reduction strategy" denominator
	DEM_IUMPR_DEN_CSERS_API	Cold start emission reduction strategy" denominator considering additional conditions reported via Dem_ReplUMPRDenRelease.
	DEM_IUMPR_DEN_EVAP	Condition based on definition of "EVAP" conditions as defined for OBD2.
	DEM_IUMPR_DEN_ EVAPPURGEFLOW	Condition based on definition of "EVAP purge flow" conditions
	DEM_IUMPR_DEN_NONE	No further condition. Denominator increments based on GENERAL_INDIVIDUAL_ DENOMINATOR only.
	DEM_IUMPR_DEN_PHYS_API	GENERAL_INDIVIDUAL_DENOMINATOR condition considering additional physical condition (component activity) computed within the SW-C and reported via Dem_ReplUMPRDen Release
Pre-Compile only	true	

⌋

[ECUC_Dem_00737] Definition of EcucEnumerationParamDef DemIUMPRGroup ⌈

Parameter Name	DemIUMPRGroup	
Parent Container	DemRatio	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This parameter specifies the assigned IUMPR group of the ratio Id. For details see SAE J1979-DA	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_IUMPR_AFRI1	Air Fuel Ratio Imbalance Monitor Bank 1
	DEM_IUMPR_AFRI2	Air Fuel Ratio Imbalance Monitor Bank 2
	DEM_IUMPR_BOOSTPRS	Boost Pressure Monitor
	DEM_IUMPR_CAT1	Catalyst Monitor - Bank 1
	DEM_IUMPR_CAT2	Catalyst Monitor - Bank 2
	DEM_IUMPR_EGR	EGR and/or VVT Monitor
	DEM_IUMPR_EGSENSOR	Exhaust Gas Sensor Monitor
	DEM_IUMPR_EVAP	EVAP Monitor
	DEM_IUMPR_FLSYS	Fuel System Monitor
	DEM_IUMPR_NMHCCAT	NMHC Catalyst Monitor
	DEM_IUMPR_NOXADSORB	NOx Adsorber Monitor
	DEM_IUMPR_NOXCAT	NOx/SCR Catalyst Monitor
	DEM_IUMPR_OXS1	Oxygen Sensor Monitor - Bank 1
	DEM_IUMPR_OXS2	Oxygen Sensor Monitor - Bank 2
	DEM_IUMPR_PF1	Particulate Filter Monitor Bank 1
	DEM_IUMPR_PF2	Particulate Filter Monitor Bank 2
	DEM_IUMPR_PMFILTER	PM Filter Monitor
	DEM_IUMPR_PRIVATE	Private group (not reported as InfoType $08/$0B)
	DEM_IUMPR_SAIR	AIR Monitor (Secondary Air)
	DEM_IUMPR_SECOXS1	Secondary Oxygen Sensor Monitor - Bank 1
	DEM_IUMPR_SECOXS2	Secondary Oxygen Sensor Monitor - Bank 2
Pre-Compile only	true	

⌋

[ECUC_Dem_00787] Definition of EcucIntegerParamDef DemRatioId ⌈

Parameter Name	DemRatioId	
Parent Container	DemRatio	
Upstream mapping defined	–	
Description	Defines a unique ratio 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. The ratio Ids should be sequentially ordered beginning with 0 and no gaps in between.	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00741] Definition of EcucEnumerationParamDef DemRatioKind ⌈

Parameter Name	DemRatioKind	
Parent Container	DemRatio	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This parameter defines whether the ratio will be calculated API or observer based.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_RATIO_API	API based ratio Id
	DEM_RATIO_OBSERVER	Observer based ratio Id
Pre-Compile only	true	

⌋

[ECUC_Dem_00735] Definition of EcucReferenceDef DemDiagnosticEventRef ⌈

Parameter Name	DemDiagnosticEventRef
Parent Container	DemRatio
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This reference contains the link to a diagnostic event.
Multiplicity	1
Type	Reference to DemEventParameter
Pre-Compile only	true

⌋

[ECUC_Dem_00736] Definition of EcucReferenceDef DemFunctionIdRef ⌈

Parameter Name	DemFunctionIdRef
Parent Container	DemRatio
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This reference contains the link to a function identifier within the FiM which is used as a primary FID.
Multiplicity	0..*
Type	Symbolic name reference to FiMFID
Pre-Compile only	true

⌋

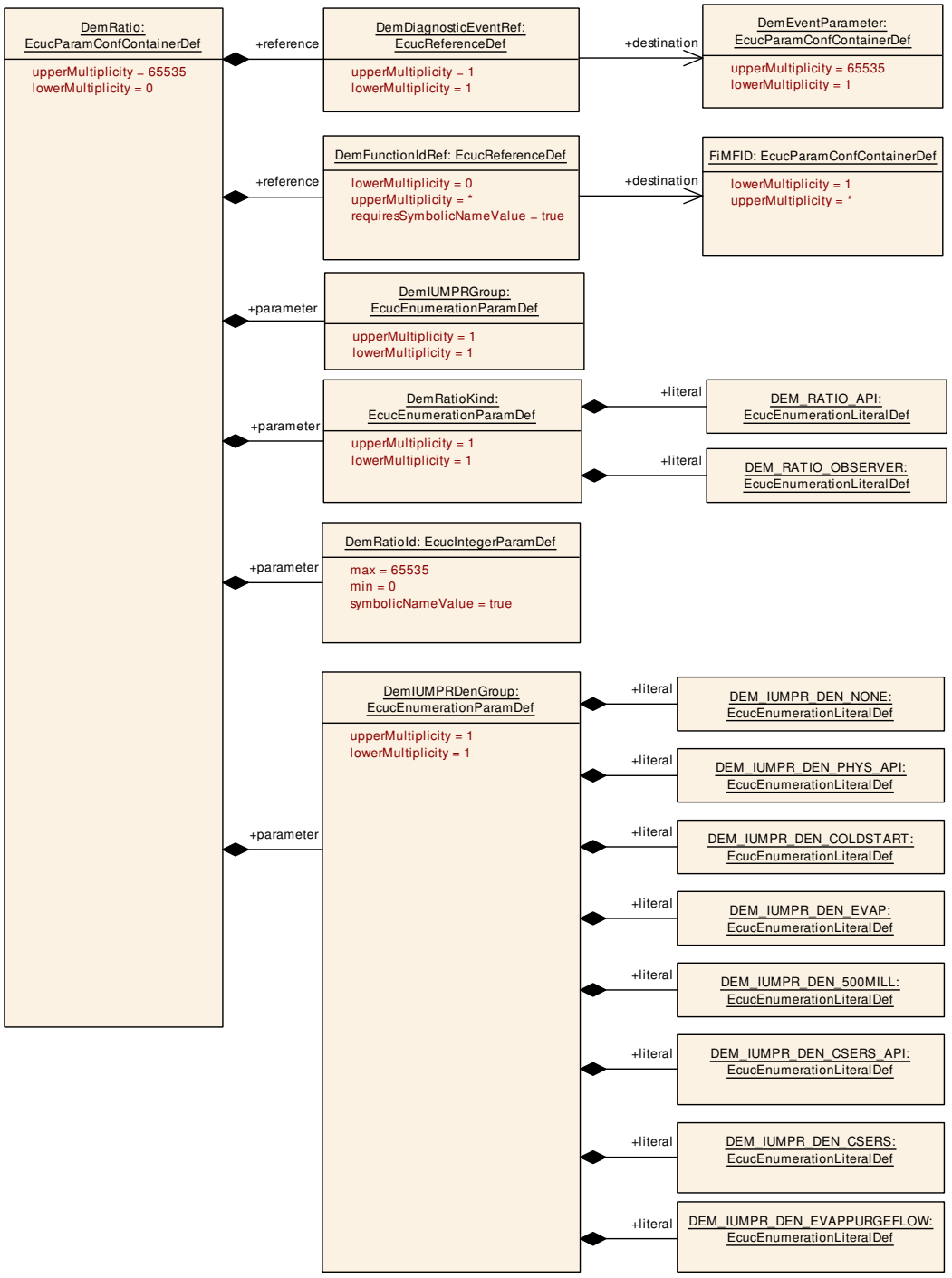


Figure 10.20: DemRatio configuration overview

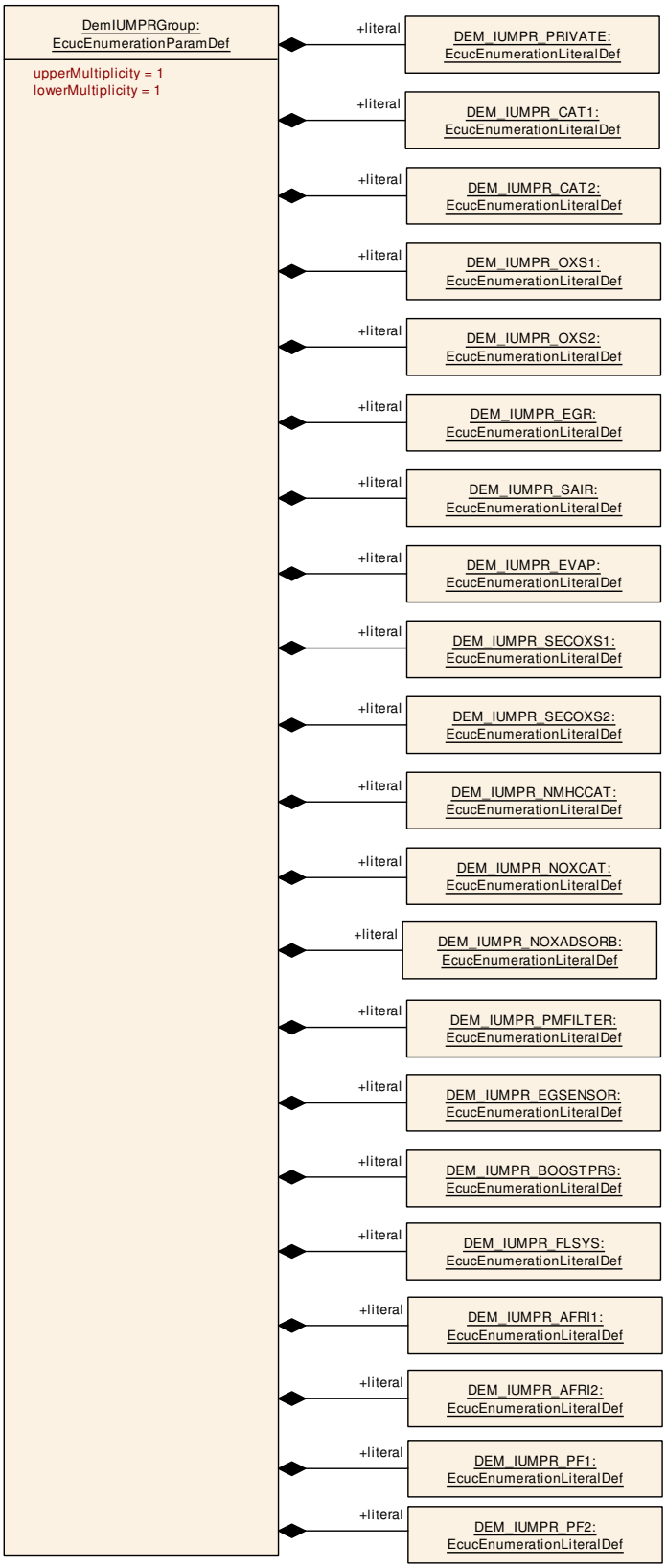


Figure 10.21: DemIUMPRGroup configuration overview

10.2.3.4 DemDtrs

[ECUC_Dem_00913] Definition of EcucParamConfContainerDef DemDtrs ⌈

Container Name	DemDtrs
Parent Container	DemConfigSet
Upstream mapping defined	–
Description	This container holds the configuration of DTRs collection.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

No Included Parameters

Included Containers		
Container Name	**Multiplicity**	**Description**
DemDtr	0..65535	This container holds the configuration of one individual DTR.

⌋

10.2.3.5 DemDtr

[ECUC_Dem_00806] Definition of EcucParamConfContainerDef DemDtr ⌈

Container Name	DemDtr
Parent Container	DemDtrs
Upstream mapping defined	–
Description	This container holds the configuration of one individual DTR.
Multiplicity	0..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDtrCompuDenominator0	1	[ECUC_Dem_00815]
DemDtrCompuNumerator0	1	[ECUC_Dem_00813]
DemDtrCompuNumerator1	1	[ECUC_Dem_00814]
DemDtrId	1	[ECUC_Dem_00807]
DemDtrMid	1	[ECUC_Dem_00809]
DemDtrTid	1	[ECUC_Dem_00810]
DemDtrUasid	1	[ECUC_Dem_00811]
DemDtrUpdateKind	1	[ECUC_Dem_00812]
DemDtrEventRef	0..1	[ECUC_Dem_00808]

No Included Containers

⌋

[ECUC_Dem_00815] Definition of EcucFloatParamDef DemDtrCompuDenominator0 ⌈

Parameter Name	DemDtrCompuDenominator0	
Parent Container	DemDtr	
Upstream mapping defined	–	
Description	Part of the conversion between the binary representation and the physical meaning analogous to the SW-C Template conversion CompuRationalCoeffs with 2 numerator coefficients and 1 denominator coefficient in the direction compuInternalToPhys. This is the only one supported denominator value, a constant divisor. The value 0 is not allowed.	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	[-INF .. INF]	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00813] Definition of EcucFloatParamDef DemDtrCompuNumerator0 ⌈

Parameter Name	DemDtrCompuNumerator0	
Parent Container	DemDtr	
Upstream mapping defined	–	
Description	Part of the conversion between the binary representation and the physical meaning analogous to the SW-C Template conversion CompuRationalCoeffs with 2 numerator coefficients and 1 denominator coefficient in the direction compuInternalToPhys. This is the first numerator value, which is multiplied with x^0, i.e., the offset.	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	[-INF .. INF]	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00814] Definition of EcucFloatParamDef DemDtrCompuNumerator1 ⌈

Parameter Name	DemDtrCompuNumerator1	
Parent Container	DemDtr	
Upstream mapping defined	–	
Description	Part of the conversion between the binary representation and the physical meaning analogous to the SW-C Template conversion CompuRationalCoeffs with 2 numerator coefficients and 1 denominator coefficient in the direction compuInternalToPhys. This is the second numerator value, which is multiplied with x^1, i.e., the factor.	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	[-INF .. INF]	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00807] Definition of EcucIntegerParamDef DemDtrId ⌈

Parameter Name	DemDtrId	
Parent Container	DemDtr	
Upstream mapping defined	–	
Description	The index identifier value assigned to this DTR. The value is generated during the Dem configuration process.	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00809] Definition of EcucIntegerParamDef DemDtrMid ⌈

Parameter Name	DemDtrMid	
Parent Container	DemDtr	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	The OBDMID of the DTR. The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, 0xE0 are reserved.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00810] Definition of EcucIntegerParamDef DemDtrTid ⌈

Parameter Name	DemDtrTid	
Parent Container	DemDtr	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	The OBDTID of the DTR.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00811] Definition of EcucIntegerParamDef DemDtrUasid ⌈

Parameter Name	DemDtrUasid
Parent Container	DemDtr
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]

▽

△

Description	The UaSId the DTR data shall be scaled to, and reported together with the rescaled DTR data.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00812] Definition of EcucEnumerationParamDef DemDtrUpdateKind ⌈

Parameter Name	DemDtrUpdateKind	
Parent Container	DemDtr	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Update conditions applied by the Dem to reports of DTR values. Only supported if a related Event is configured	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_DTR_UPDATE_ALWAYS	Any DTR result reported by the monitor is used by the Dem.
	DEM_DTR_UPDATE_STEADY	The Dem accepts reported DTRs only when the configured debouncing mechanism is stable at the FAIL or PASS limit.
Default value	DEM_DTR_UPDATE_ALWAYS	
Pre-Compile only	true	

⌋

[ECUC_Dem_00808] Definition of EcucReferenceDef DemDtrEventRef ⌈

Parameter Name	DemDtrEventRef
Parent Container	DemDtr
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	Reference to the DemEventParameter this DTR is related to. If the related event is not configured, the Dem cannot ensure consistency between the DTR and the event.
Multiplicity	0..1
Type	Reference to DemEventParameter
Pre-Compile only	true

⌋

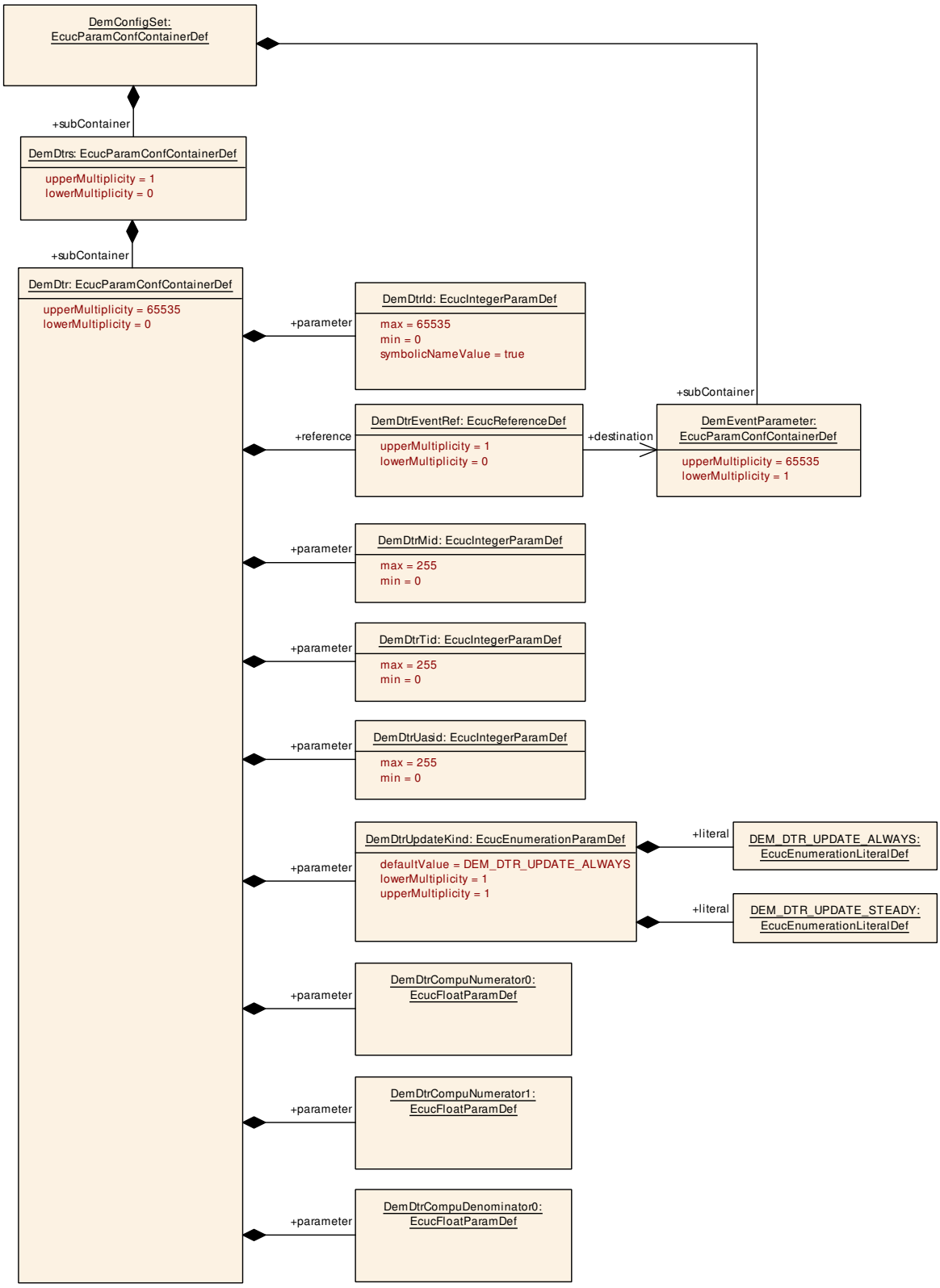


Figure 10.22: DemDtr configuration overview

10.2.3.6 DemPidClass

[ECUC_Dem_00729] Definition of EcucParamConfContainerDef DemPidClass ⌈

Container Name	DemPidClass
Parent Container	DemConfigSet
Upstream mapping defined	–
Description	This container contains the different PIDs for the single global OBD relevant freeze frame class. It is assembled out of one or several data elements.
Multiplicity	0..255
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemPidIdentifier	1	[ECUC_Dem_00705]

Included Containers		
Container Name	**Multiplicity**	**Description**
DemPidDataElement	1..255	This container contains the different data elements contained in the specific PID.

⌋

[ECUC_Dem_00705] Definition of EcucIntegerParamDef DemPidIdentifier ⌈

Parameter Name	DemPidIdentifier	
Parent Container	DemPidClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	identifier of the PID	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

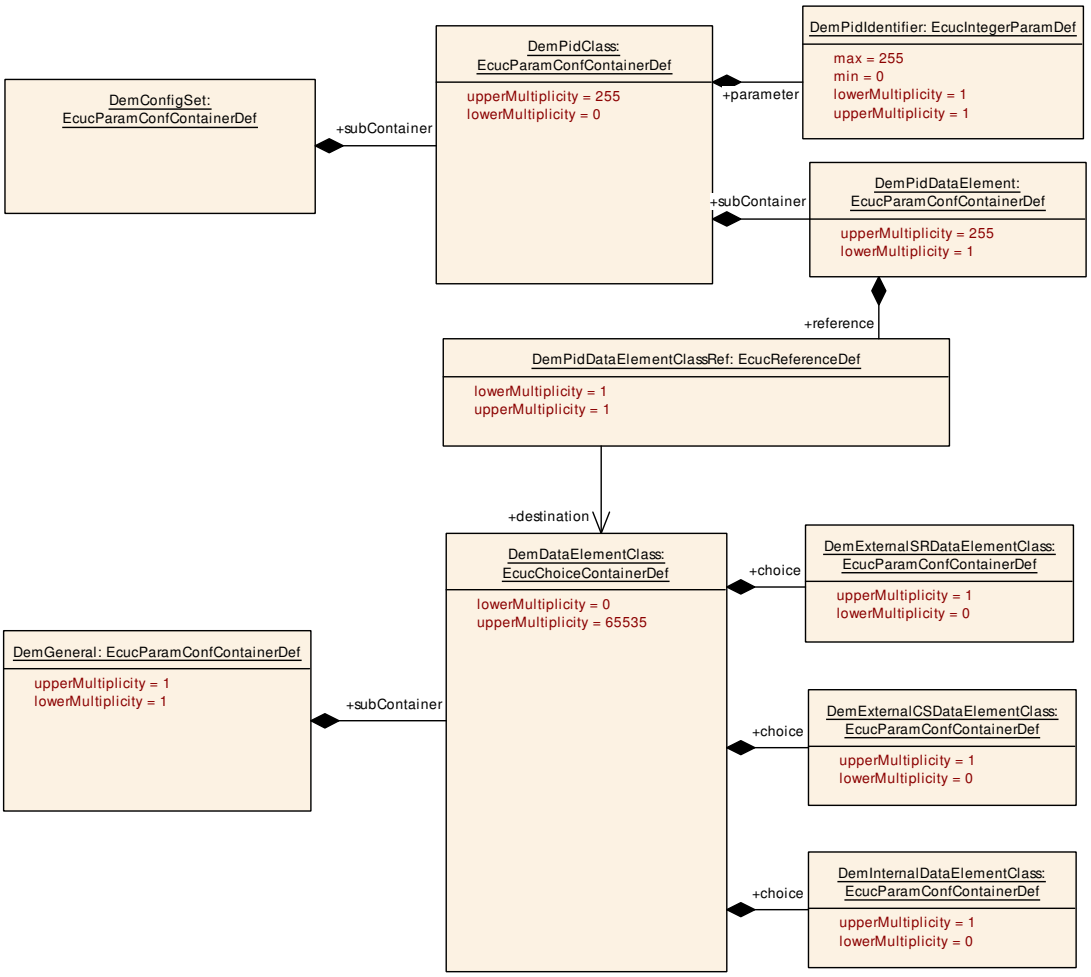


Figure 10.23: DemPidClass configuration overview

10.2.3.7 DemPidDataElement

[ECUC_Dem_00896] Definition of EcucParamConfContainerDef DemPidDataElement ⌈

Container Name	DemPidDataElement
Parent Container	DemPidClass
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This container contains the different data elements contained in the specific PID.
Multiplicity	1..255
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemPidDataElementClassRef	1	[ECUC_Dem_00733]

No Included Containers

⌋

[ECUC_Dem_00733] Definition of EcucReferenceDef DemPidDataElementClassRef ⌈

Parameter Name	DemPidDataElementClassRef	
Parent Container	DemPidDataElement	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This reference contains the link to a data element class.	
Multiplicity	1	
Type	Reference to DemDataElementClass	
Post-Build Value	Loadable	Selectable

⌋

10.2.4 J1939

10.2.4.1 DemGeneralJ19139

[ECUC_Dem_00864] Definition of EcucParamConfContainerDef DemGeneralJ1939 ⌈

Container Name	DemGeneralJ1939
Parent Container	DemGeneral
Upstream mapping defined	–
Description	This container contains the general J1939-specific configuration (parameters) of the Dem module. If the container exists the J1939 support is enabled.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemJ1939ClearDtcSupport	1	[ECUC_Dem_00872]
DemJ1939Dm31Support	1	[ECUC_Dem_00868]
DemJ1939ExpandedFreezeFrameSupport	1	[ECUC_Dem_00873]
DemJ1939FreezeFrameSupport	1	[ECUC_Dem_00866]
DemJ1939RatioSupport	1	[ECUC_Dem_00867]
DemJ1939Readiness1Support	1	[ECUC_Dem_00869]
DemJ1939Readiness2Support	1	[ECUC_Dem_00870]
DemJ1939Readiness3Support	1	[ECUC_Dem_00871]
DemJ1939ReadingDtcSupport	1	[ECUC_Dem_00865]

Included Containers		
Container Name	**Multiplicity**	**Description**
DemCallbackJ1939DTCStatus Changed	0..*	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. In case there is a DemCallbackDTCStatusChangedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackDTCStatusChangedFnc, the Dem will have an R-Port requiring the interface Callback DTCStatusChanged. Status change notifications are supported for DTCs in primary memory only.
DemJ1939FreezeFrameClass	0..255	This container contains the combinations of SPNs s for a J1939 relevant freeze frame.
DemSPNClass	0..*	This container contains the configuration (parameters) for a SPN.

⌋

[ECUC_Dem_00872] Definition of EcucBooleanParamDef DemJ1939ClearDtcSupport ⌈

Parameter Name	DemJ1939ClearDtcSupport
Parent Container	DemGeneralJ1939
Upstream mapping defined	–
Description	This configuration switch defines whether clearing J1939 DTCs (DM03,DM11 and DM22) is supported or not. This switches on and off the APIs Dem_J1939DcmClearDTC and Dem_J1939Dcm ClearSingleDTC .
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00868] Definition of EcucBooleanParamDef DemJ1939Dm31Support ⌈

Parameter Name	DemJ1939Dm31Support
Parent Container	DemGeneralJ1939
Upstream mapping defined	–
Description	This configuration switch defines whether J1939 DM31 is supported or not. This switches on and off the APIs Dem_J1939DcmFirstDTCwithLampStatus and Dem_J1939DcmGetNextDTCwithLampStatus. .
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00873] Definition of EcucBooleanParamDef DemJ1939ExpandedFreezeFrameSupport ⌈

Parameter Name	DemJ1939ExpandedFreezeFrameSupport
Parent Container	DemGeneralJ1939
Upstream mapping defined	–
Description	This configuration switch defines whether J1939 expanded freeze frames are supported or not. This switches on and off the APIs Dem_J1939DcmSetFreezeFrameFilter, Dem_J1939DcmGetNextFreezeFrame and Dem_J1939DcmGetNextSPNInFreezeFrame.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00866] Definition of EcucBooleanParamDef DemJ1939FreezeFrameSupport ⌈

Parameter Name	DemJ1939FreezeFrameSupport
Parent Container	DemGeneralJ1939
Upstream mapping defined	–
Description	This configuration switch defines whether J1939 freeze frames are supported or not. This switches on and off the APIs Dem_J1939DcmSetFreezeFrameFilter and Dem_J1939DcmGetNextFreezeFrame.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00867] Definition of EcucBooleanParamDef DemJ1939RatioSupport ⌈

Parameter Name	DemJ1939RatioSupport
Parent Container	DemGeneralJ1939
Upstream mapping defined	–
Description	This configuration switch defines whether J1939 performance ratios are supported or not. This switches on and off the APIs Dem_J1939DcmSetRatioFilter and Dem_J1939DcmGetNextFilteredRatio.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00869] Definition of EcucBooleanParamDef DemJ1939Readiness1 Support ⌈

Parameter Name	DemJ1939Readiness1Support
Parent Container	DemGeneralJ1939
Upstream mapping defined	–
Description	This configuration switch defines whether J1939 diagnostic readiness 1 is supported or not. This switches on and off the API Dem_J1939DcmReadDiagnosticReadiness1.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00870] Definition of EcucBooleanParamDef DemJ1939Readiness2 Support ⌈

Parameter Name	DemJ1939Readiness2Support
Parent Container	DemGeneralJ1939
Upstream mapping defined	–
Description	This configuration switch defines whether J1939 diagnostic readiness 2 is supported or not. This switches on and off the API Dem_J1939DcmReadDiagnosticReadiness2.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00871] Definition of EcucBooleanParamDef DemJ1939Readiness3 Support ⌈

Parameter Name	DemJ1939Readiness3Support
Parent Container	DemGeneralJ1939
Upstream mapping defined	–
Description	This configuration switch defines whether J1939 diagnostic readiness 3 is supported or not. This switches on and off the API Dem_J1939DcmReadDiagnosticReadiness3.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00865] Definition of EcucBooleanParamDef DemJ1939ReadingDtcSupport ⌈

Parameter Name	DemJ1939ReadingDtcSupport
Parent Container	DemGeneralJ1939
Upstream mapping defined	–
Description	This configuration switch defines whether J1939 DTC readout is supported or not. This switches on and off the APIs Dem_J1939DcmSetDTCFilter, Dem_J1939DcmGetNumberOfFilteredDTC and Dem_J1939DcmGetNextFilteredDTC.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

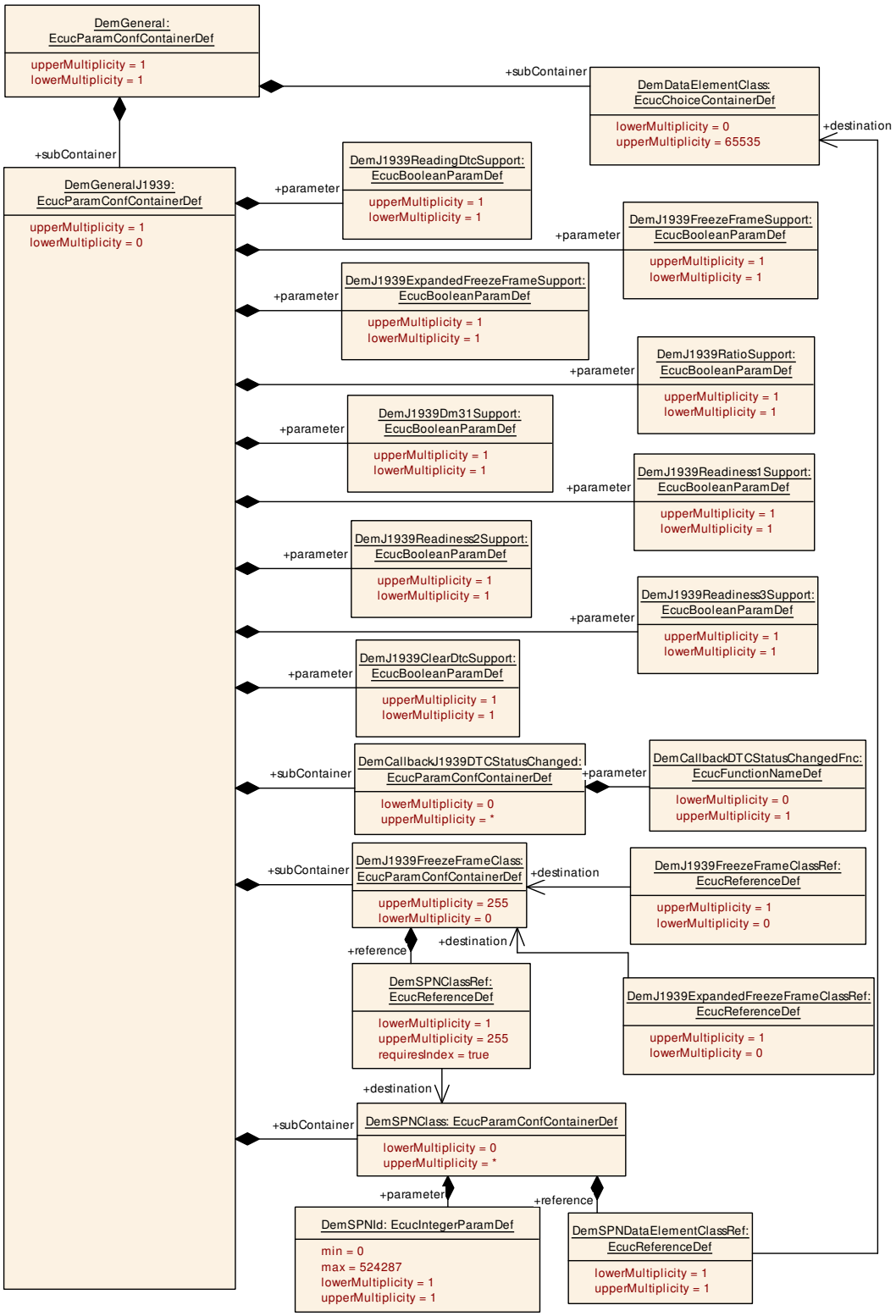


Figure 10.24: DemGeneralJ1939 configuration overview

10.2.4.2 DemJ1939FreezeFrameClass

[ECUC_Dem_00828] Definition of EcucParamConfContainerDef DemJ1939FreezeFrameClass ⌈

Container Name	DemJ1939FreezeFrameClass
Parent Container	DemGeneralJ1939
Upstream mapping defined	–
Description	This container contains the combinations of SPNs s for a J1939 relevant freeze frame.
Multiplicity	0..255
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemSPNClassRef	1..255	[ECUC_Dem_00829]

No Included Containers

⌋

[ECUC_Dem_00829] Definition of EcucReferenceDef DemSPNClassRef ⌈

Parameter Name	DemSPNClassRef
Parent Container	DemJ1939FreezeFrameClass
Upstream mapping defined	–
Description	Reference to an SPN. This reference defines requiresIndex = true since it represents a ordered list of references where the order describes the order of single SPNs in the J1939 Freeze Frame. **Attributes:** requiresIndex=true
Multiplicity	1..255
Type	Reference to DemSPNClass
Pre-Compile only	true

⌋

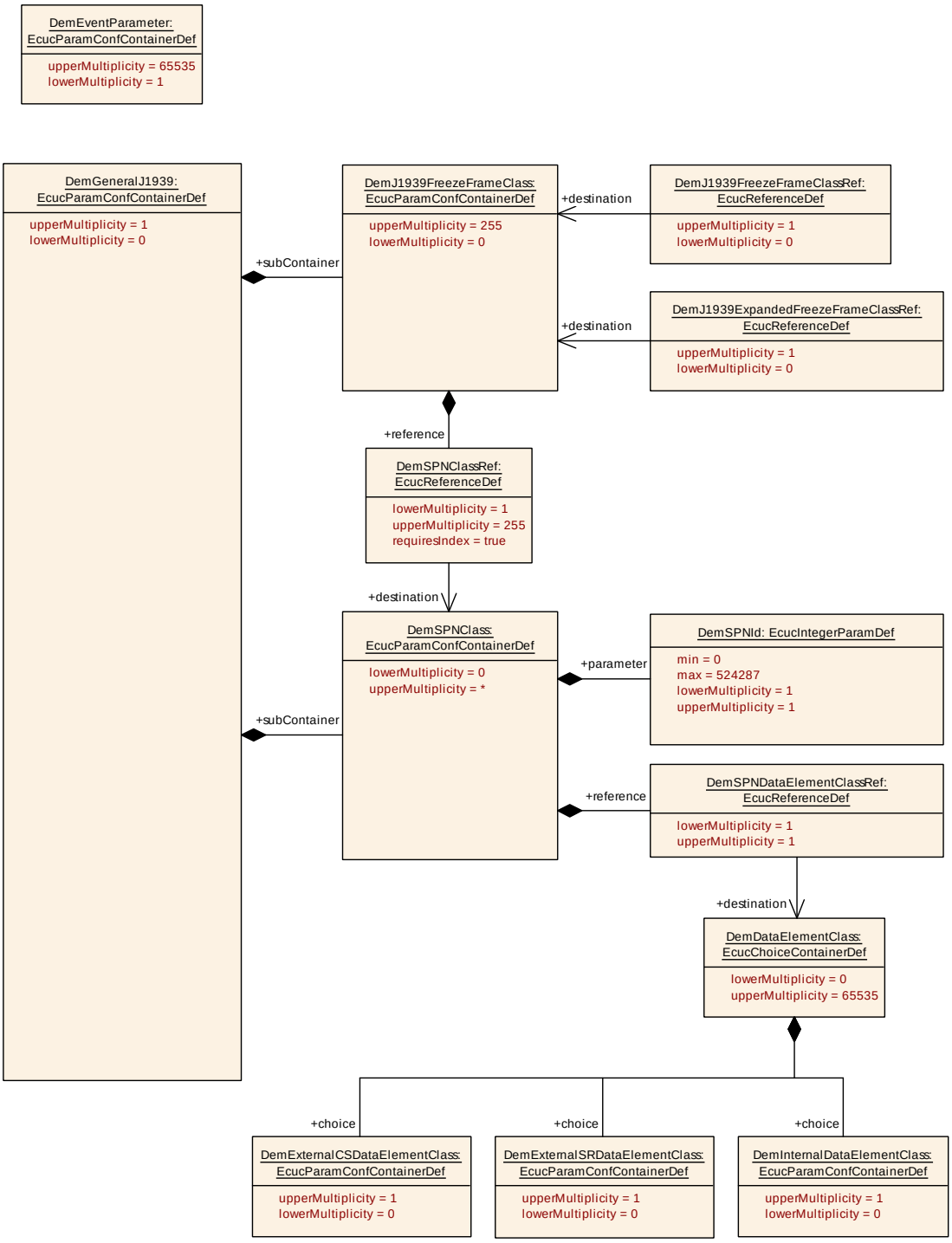


Figure 10.25: DemJ1939FreezeFrameClass configuration overview

10.2.4.3 DemSPNClass

[ECUC_Dem_00830] Definition of EcucParamConfContainerDef DemSPNClass ⌈

Container Name	DemSPNClass
Parent Container	DemGeneralJ1939
Upstream mapping defined	–
Description	This container contains the configuration (parameters) for a SPN.
Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
DemSPNId	1	[ECUC_Dem_00831]
DemSPNDataElementClassRef	1	[ECUC_Dem_00832]

No Included Containers

⌋

[ECUC_Dem_00831] Definition of EcucIntegerParamDef DemSPNId ⌈

Parameter Name	DemSPNId	
Parent Container	DemSPNClass	
Upstream mapping defined	–	
Description	Suspect parameter number	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 524287	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00832] Definition of EcucReferenceDef DemSPNDataElementClassRef ⌈

Parameter Name	DemSPNDataElementClassRef
Parent Container	DemSPNClass
Upstream mapping defined	–
Description	This reference contains the link to a data element class.
Multiplicity	1
Type	Reference to DemDataElementClass
Pre-Compile only	true

⌋

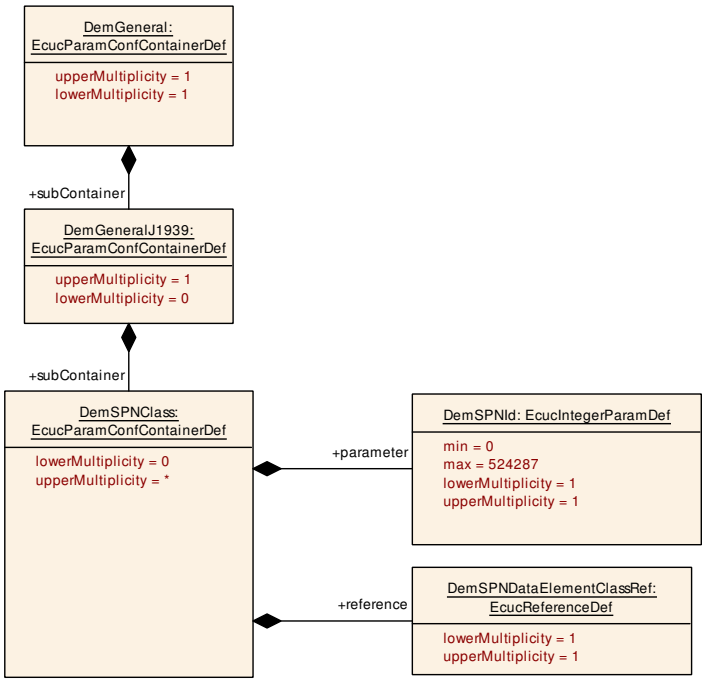


Figure 10.26: DemSPNClass configuration overview

10.2.5 Conditions

10.2.5.1 DemEnableCondition

[ECUC_Dem_00653] Definition of EcucParamConfContainerDef DemEnableCondition ⌈

Container Name	DemEnableCondition
Parent Container	DemGeneral
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This container contains the configuration (parameters) for enable conditions.
Multiplicity	0..255
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemEnableConditionId	1	[ECUC_Dem_00654]
DemEnableConditionStatus	1	[ECUC_Dem_00656]

No Included Containers

⌋

[ECUC_Dem_00654] Definition of EcucIntegerParamDef DemEnableConditionId ⌈

Parameter Name	DemEnableConditionId
Parent Container	DemEnableCondition
Upstream mapping defined	–
Description	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. The enable conditions should be sequentially ordered beginning with 0 and no gaps in between.
Multiplicity	1
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)
Range	0 .. 255
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00656] Definition of EcucBooleanParamDef DemEnableCondition Status ⌈

Parameter Name	DemEnableConditionStatus
Parent Container	DemEnableCondition
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
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). true: acceptance of a diagnostic event enabled false: acceptance of a diagnostic event disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

10.2.5.2 DemEnableConditionGroup

[ECUC_Dem_00745] Definition of EcucParamConfContainerDef DemEnableConditionGroup ⌈

Container Name	DemEnableConditionGroup
Parent Container	DemGeneral
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This container contains the configuration (parameters) for enable condition groups.
Multiplicity	0..255
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemEnableConditionRef	1..255	[ECUC_Dem_00655]

No Included Containers

⌋

[ECUC_Dem_00655] Definition of EcucReferenceDef DemEnableConditionRef ⌈

Parameter Name	DemEnableConditionRef
Parent Container	DemEnableConditionGroup
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	References an enable condition.
Multiplicity	1..255
Type	Reference to DemEnableCondition
Pre-Compile only	true

⌋

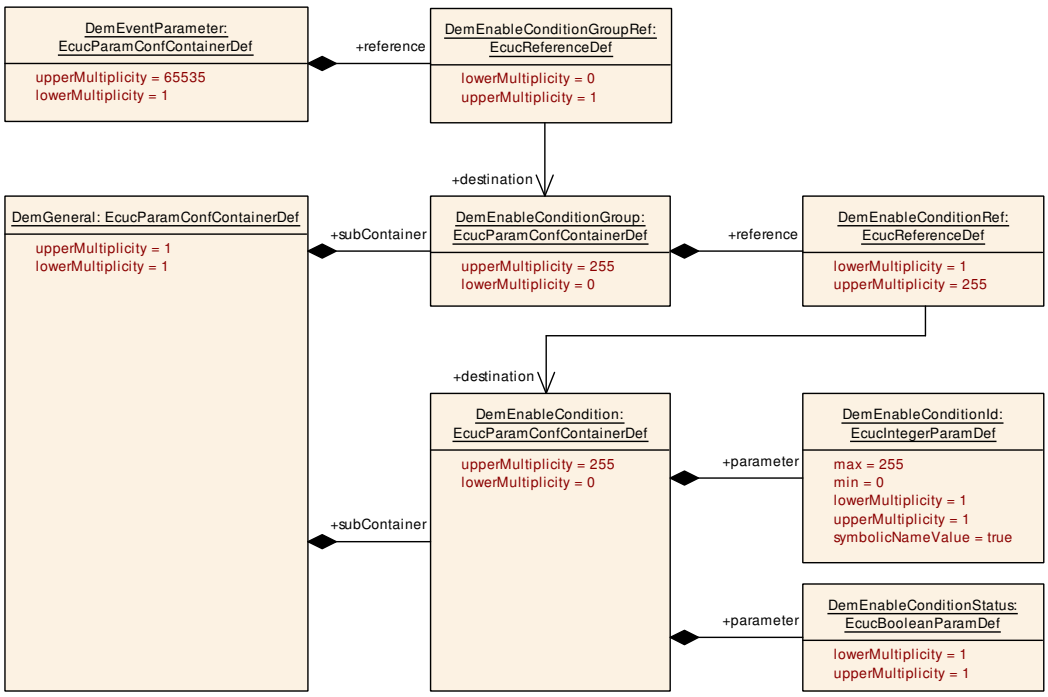


Figure 10.27: DemEnableCondition configuration overview

10.2.5.3 DemStorageCondition

[ECUC_Dem_00728] Definition of EcucParamConfContainerDef DemStorageCondition ⌈

Container Name	DemStorageCondition
Parent Container	DemGeneral
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This container contains the configuration (parameters) for storage conditions.
Multiplicity	0..255
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemStorageConditionId	1	[ECUC_Dem_00730]
DemStorageConditionStatus	1	[ECUC_Dem_00731]
DemStorageConditionReplacementEventRef	0..1	[ECUC_Dem_00893]

No Included Containers

⌋

[ECUC_Dem_00730] Definition of EcucIntegerParamDef DemStorageConditionId ⌈

Parameter Name	DemStorageConditionId
Parent Container	DemStorageCondition
Upstream mapping defined	–
Description	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. The storage conditions should be sequentially ordered beginning with 0 and no gaps in between.
Multiplicity	1
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)
Range	0 .. 255
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00731] Definition of EcucBooleanParamDef DemStorageCondition Status ⌈

Parameter Name	DemStorageConditionStatus
Parent Container	DemStorageCondition
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	Defines the initial status for enable or disable of storage of a diagnostic event. The value is the initialization after power up (before this condition is reported the first time). true: storage of a diagnostic event enabled false: storage of a diagnostic event disabled
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–

▽

△

Pre-Compile only	true

⌋

[ECUC_Dem_00893] Definition of EcucReferenceDef DemStorageConditionReplacementEventRef ⌈

Parameter Name	DemStorageConditionReplacementEventRef
Parent Container	DemStorageCondition
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	Specifies the reference to an event which is stored to event memory and supports failure analysis.
Multiplicity	0..1
Type	Reference to DemEventParameter
Pre-Compile only	true

⌋

10.2.5.4 DemStorageConditionGroup

[ECUC_Dem_00773] Definition of EcucParamConfContainerDef DemStorageConditionGroup ⌈

Container Name	DemStorageConditionGroup
Parent Container	DemGeneral
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This container contains the configuration (parameters) for storage condition groups.
Multiplicity	0..255
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemStorageConditionRef	1..255	[ECUC_Dem_00768]

No Included Containers

⌋

[ECUC_Dem_00768] Definition of EcucReferenceDef DemStorageConditionRef ⌈

Parameter Name	DemStorageConditionRef
Parent Container	DemStorageConditionGroup
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	References an enable condition.

▽

△

Multiplicity	1..255
Type	Reference to DemStorageCondition
Pre-Compile only	true

⌋

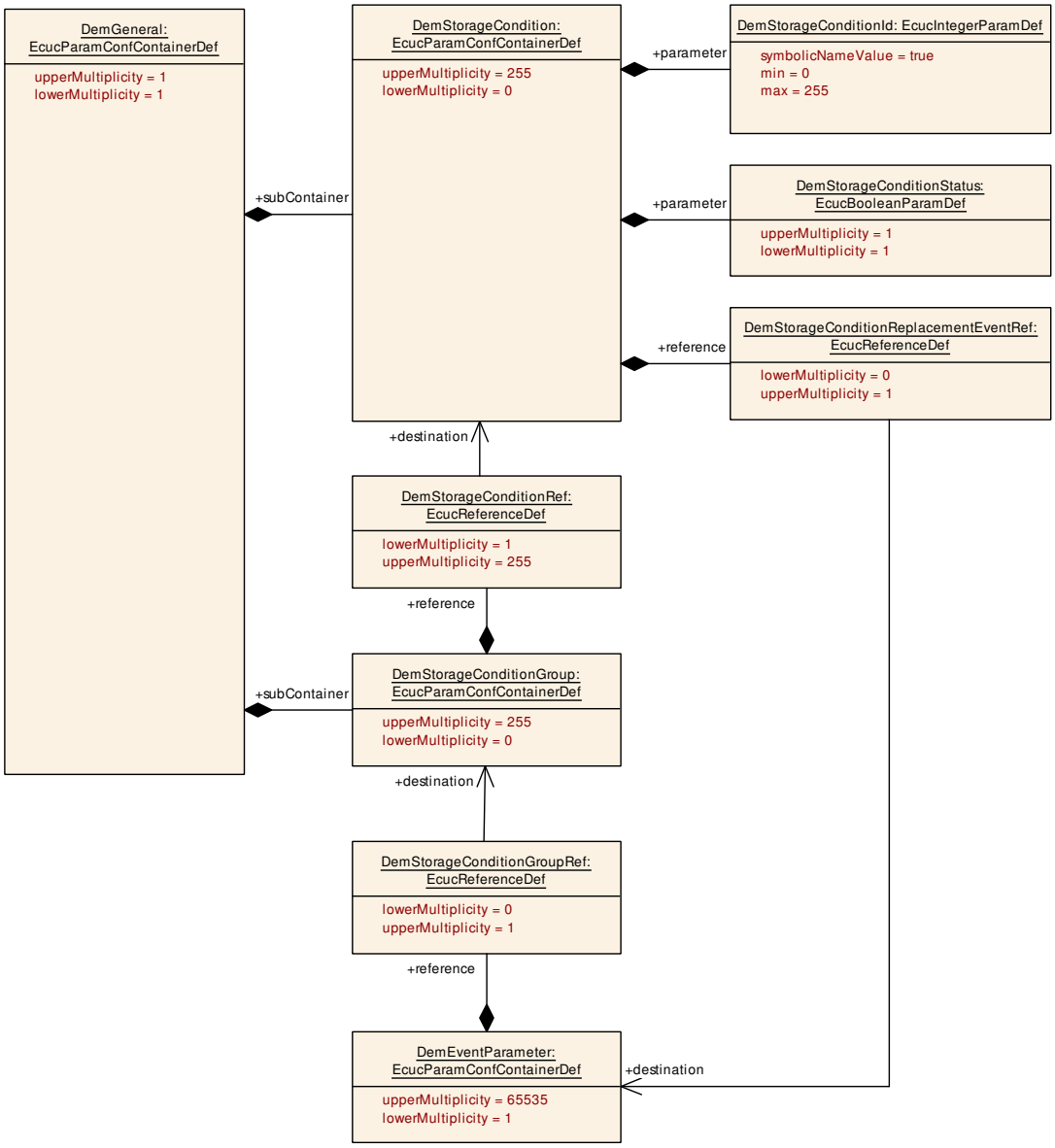


Figure 10.28: DemStorageCondition configuration overview

10.2.6 Event Memory

10.2.6.1 DemEventMemorySet

[ECUC_Dem_00939] Definition of EcucParamConfContainerDef DemEventMemorySet ⌈

Container Name	DemEventMemorySet
Parent Container	DemGeneral
Upstream mapping defined	–
Description	This container is a collection of referenced event memories and related information for a Dem client.
Multiplicity	1..255
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemMaxNumberEventEntryPermanent	1	[ECUC_Dem_00689]
DemTypeOfDTCSupported	1	[ECUC_Dem_00720]
DemAmberWarningLampIndicatorRef	0..1	[ECUC_Dem_00821]
DemMILIndicatorRef	0..1	[ECUC_Dem_00723]
DemProtectLampIndicatorRef	0..1	[ECUC_Dem_00822]
DemRedStopLampIndicatorRef	0..1	[ECUC_Dem_00820]

Included Containers		
Container Name	**Multiplicity**	**Description**
DemClearDTCNotification	0..255	Contains callback function definition which are called during clear DTC operations.
DemIndicator	0..255	This container contains the configuration (parameters) for Indicators.
DemPrimaryMemory	1	This container contains the primary event memory specific parameters of the Dem module.
DemUserDefinedMemory	0..30	This container contains the user defined event memory specific parameters of the Dem module.

⌋

[ECUC_Dem_00689] Definition of EcucIntegerParamDef DemMaxNumberEventEntryPermanent ⌈

Parameter Name	DemMaxNumberEventEntryPermanent	
Parent Container	DemEventMemorySet	
Upstream mapping defined	–	
Description	Maximum number of events which can be stored in the permanent memory. The assignment of an event to this memory type is dynamic and used for emission-related events only.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00720] Definition of EcucEnumerationParamDef DemTypeOfDTC-Supported ⌈

Parameter Name	DemTypeOfDTCSupported	
Parent Container	DemEventMemorySet	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This parameter defines the format returned by Dem_GetTranslationType and does not relate to/influence the supported Dem functionality.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_DTC_TRANSLATION_ISO11992_4	ISO11992-4 DTC format
	DEM_DTC_TRANSLATION_ISO14229_1	ISO14229-1 DTC format (3 byte format)
	DEM_DTC_TRANSLATION_ISO15031_6	ISO15031-6 DTC format = SAE_J2012-DA_DTCFormat_00 (2 byte format)
	DEM_DTC_TRANSLATION_SAEJ1939_73	SAEJ1939-73 DTC format
	DEM_DTC_TRANSLATION_SAE_J2012_DA_DTCFORMAT_04	SAE_J2012-DA_DTCFormat_00 (3 byte format)
Pre-Compile only	true	

⌋

[ECUC_Dem_00821] Definition of EcucReferenceDef DemAmberWarningLamp IndicatorRef ⌈

Parameter Name	DemAmberWarningLampIndicatorRef
Parent Container	DemEventMemorySet
Upstream mapping defined	–
Description	This parameter defines the indicator representing the AmberWarningLamp . This parameter may be used for ECUs supporting J1939.
Multiplicity	0..1
Type	Reference to DemIndicator
Pre-Compile only	true

⌋

[ECUC_Dem_00723] Definition of EcucReferenceDef DemMILIndicatorRef ⌈

Parameter Name	DemMILIndicatorRef
Parent Container	DemEventMemorySet
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This parameter defines the indicator representing the MIL. This parameter is mandatory for ECUs supporting OBD (refer to DemOBDSupport).
Multiplicity	0..1
Type	Reference to DemIndicator
Pre-Compile only	true

⌋

[ECUC_Dem_00822] Definition of EcucReferenceDef DemProtectLampIndicatorRef ⌈

Parameter Name	DemProtectLampIndicatorRef
Parent Container	DemEventMemorySet
Upstream mapping defined	–
Description	This parameter defines the indicator representing the ProtectLamp. This parameter may be used for ECUs supporting J1939.
Multiplicity	0..1
Type	Reference to DemIndicator
Pre-Compile only	true

⌋

[ECUC_Dem_00820] Definition of EcucReferenceDef DemRedStopLampIndicatorRef ⌈

Parameter Name	DemRedStopLampIndicatorRef
Parent Container	DemEventMemorySet
Upstream mapping defined	–
Description	This parameter defines the indicator representing the RedStopLamp. This parameter may be used for ECUs supporting J1939.
Multiplicity	0..1
Type	Reference to DemIndicator
Pre-Compile only	true

⌋

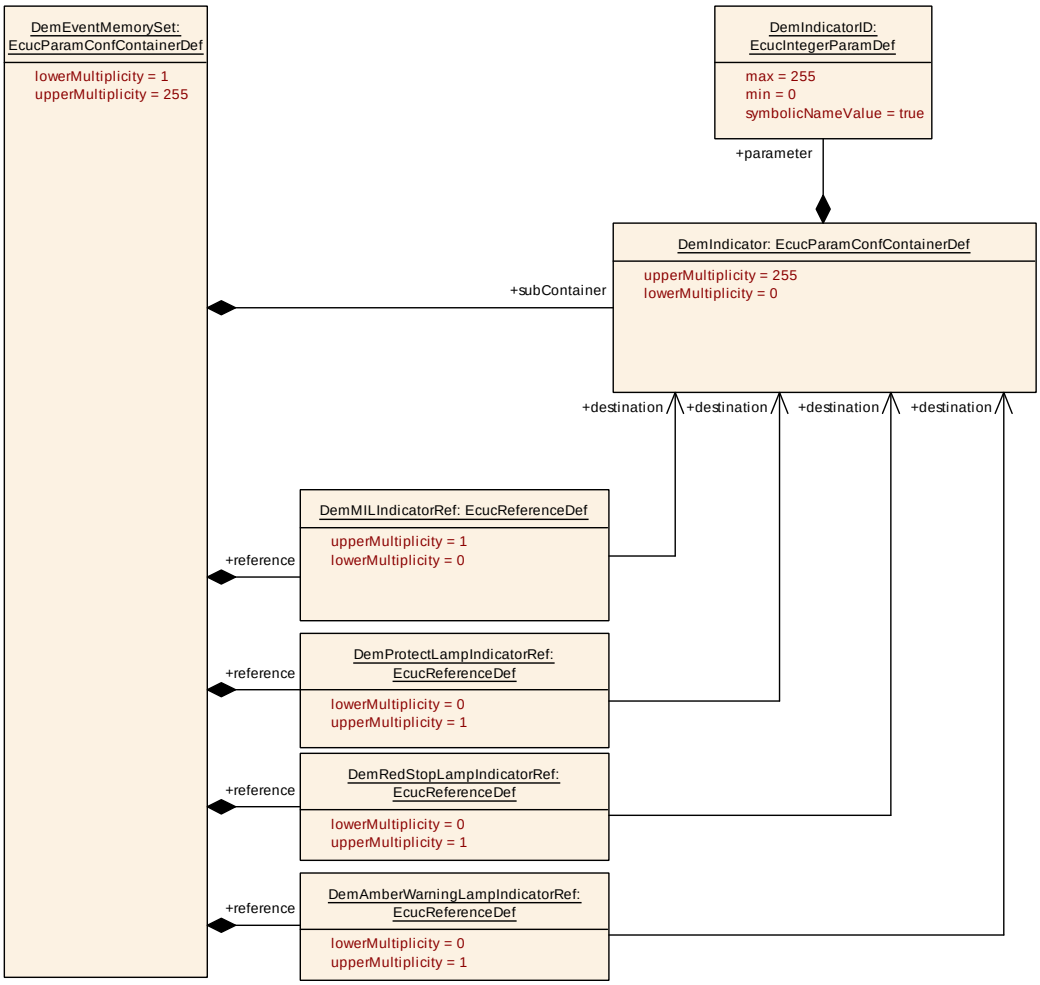


Figure 10.29: DemEventMemorySet configuration overview

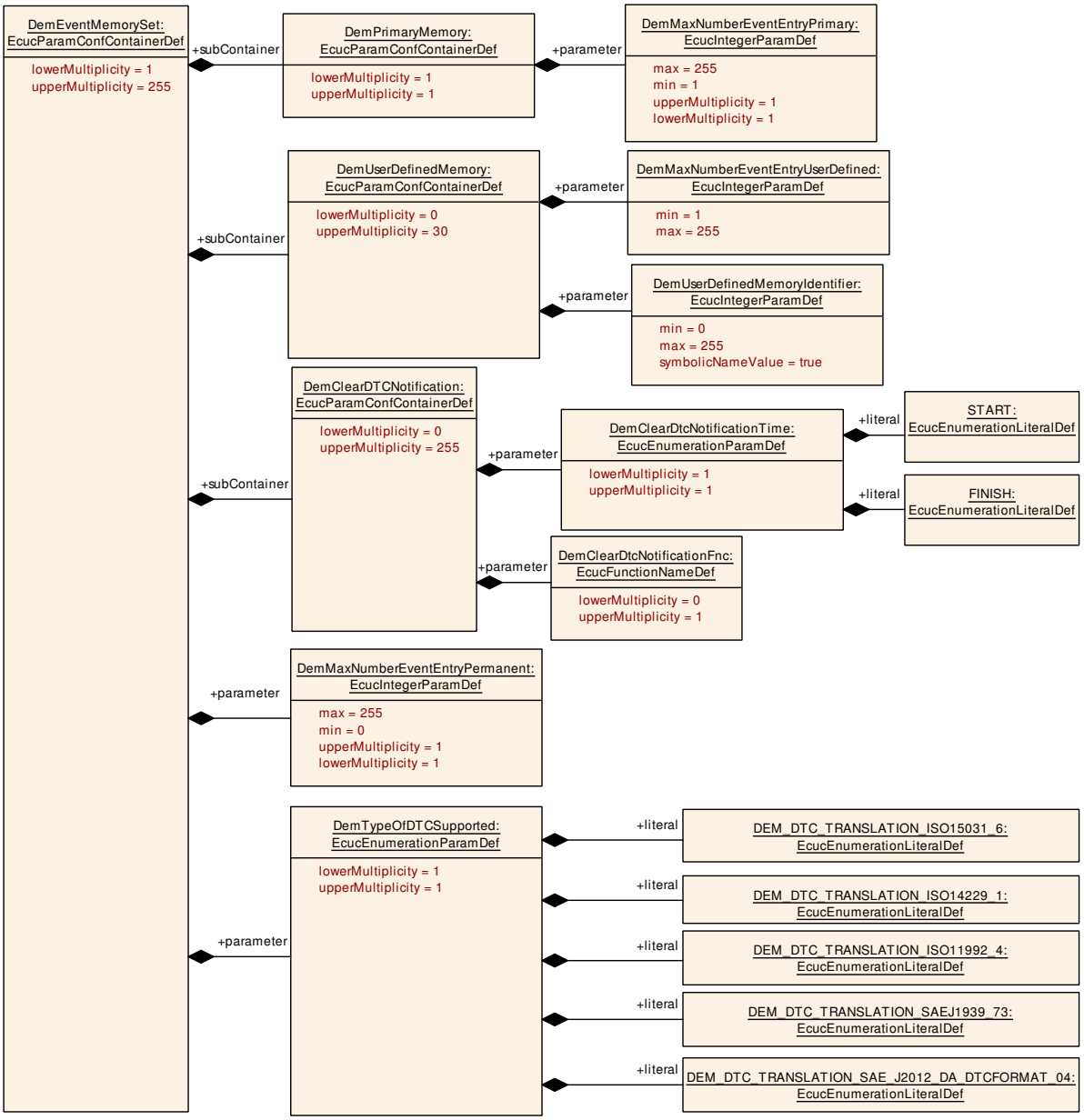


Figure 10.30: DemEventMemorySet included containers overview

10.2.6.2 DemPrimaryMemory

[ECUC_Dem_00901] Definition of EcucParamConfContainerDef DemPrimary Memory ⌈

Container Name	DemPrimaryMemory
Parent Container	DemEventMemorySet
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]

▽

△

Description	This container contains the primary event memory specific parameters of the Dem module.
Multiplicity	1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDtcStatusAvailabilityMask	1	[ECUC_Dem_00978]
DemEventDisplacementStrategy	1	[ECUC_Dem_00980]
DemEventMemoryEntryStorageTrigger	1	[ECUC_Dem_00982]
DemMaxNumberEventEntryPrimary	1	[ECUC_Dem_00690]
DemOccurrenceCounterProcessing	1	[ECUC_Dem_00984]
DemTypeOfFreezeFrameRecordNumeration	1	[ECUC_Dem_00986]

Included Containers		
Container Name	**Multiplicity**	**Description**
DemGroupOfDTC	0..65535	This container contains the configuration (parameters) for DTC groups.

⌋

[ECUC_Dem_00978] Definition of EcucIntegerParamDef DemDtcStatusAvailabilityMask ⌈

Parameter Name	DemDtcStatusAvailabilityMask
Parent Container	DemPrimaryMemory, DemUserDefinedMemory
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	Mask for the supported DTC status bits by the Dem. This mask is used in the positive response of UDS service 0x19.
Multiplicity	1
Type	EcucIntegerParamDef
Range	0 .. 255
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00980] Definition of EcucEnumerationParamDef DemEventDisplacementStrategy ⌈

Parameter Name	DemEventDisplacementStrategy
Parent Container	DemPrimaryMemory, DemUserDefinedMemory
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This configuration switch defines, whether support for event displacement is enabled or not, and which displacement strategy is followed.
Multiplicity	1
Type	EcucEnumerationParamDef

▽

△

Range	DEM_DISPLACEMENT_FULL	Event memory entry displacement is enabled, by consideration of priority active/passive status, and occurrence.
	DEM_DISPLACEMENT_NONE	Event memory entry displacement is disabled.
	DEM_DISPLACEMENT_PRIO_OCC	Event memory entry displacement is enabled, by consideration of priority and occurrence (but without active/passive status).
Pre-Compile only	true	

⌋

[ECUC_Dem_00982] Definition of EcucEnumerationParamDef DemEventMemoryEntryStorageTrigger ⌈

Parameter Name	DemEventMemoryEntryStorageTrigger	
Parent Container	DemPrimaryMemory, DemUserDefinedMemory	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Configures the primary trigger to allocate an event memory entry.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_TRIGGER_ON_CONFIRMED	Event Memory entries are triggered if the UDS status bit 3 (confirmedDTC) changes from 0 to 1.
	DEM_TRIGGER_ON_FDC_THRESHOLD	Event Memory entries are triggered when the FDC threshold is reached.
	DEM_TRIGGER_ON_TEST_FAILED	Event Memory entries are triggered if the UDS status bit 0 (testFailed) changes from 0 to 1.
Default value	DEM_TRIGGER_ON_TEST_FAILED	
Pre-Compile only	true	

⌋

[ECUC_Dem_00690] Definition of EcucIntegerParamDef DemMaxNumberEventEntryPrimary ⌈

Parameter Name	DemMaxNumberEventEntryPrimary	
Parent Container	DemPrimaryMemory	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Maximum number of events which can be stored in the primary memory	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	1 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00984] Definition of EcucEnumerationParamDef DemOccurrenceCounterProcessing ⌈

Parameter Name	DemOccurrenceCounterProcessing	
Parent Container	DemPrimaryMemory, DemUserDefinedMemory	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	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 Dem OBDSupport) configuration switch shall always set to DEM_PROCESS_OCCCTR_TF.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_PROCESS_OCCCTR_CDTC	the occurrence counter is triggered by the Test Failed bit if the fault confirmation was successful (ConfirmedDTC bit is set)
	DEM_PROCESS_OCCCTR_TF	the occurrence counter is only triggered by the TestFailed bit (and the fault confirmation is not considered) This parameter is mandatory in case of J1939.
Pre-Compile only	true	

⌋

[ECUC_Dem_00986] Definition of EcucEnumerationParamDef DemTypeOfFreezeFrameRecordNumeration ⌈

Parameter Name	DemTypeOfFreezeFrameRecordNumeration	
Parent Container	DemPrimaryMemory, DemUserDefinedMemory	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This parameter defines the type of assigning freeze frame record numbers for event-specific freeze frame records.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_FF_RECNUM_CALCULATED	freeze frame records will be numbered consecutive starting by 1 in their chronological order
	DEM_FF_RECNUM_CONFIGURED	freeze frame records will be numbered based on the given configuration in their chronological order
Pre-Compile only	true	

⌋

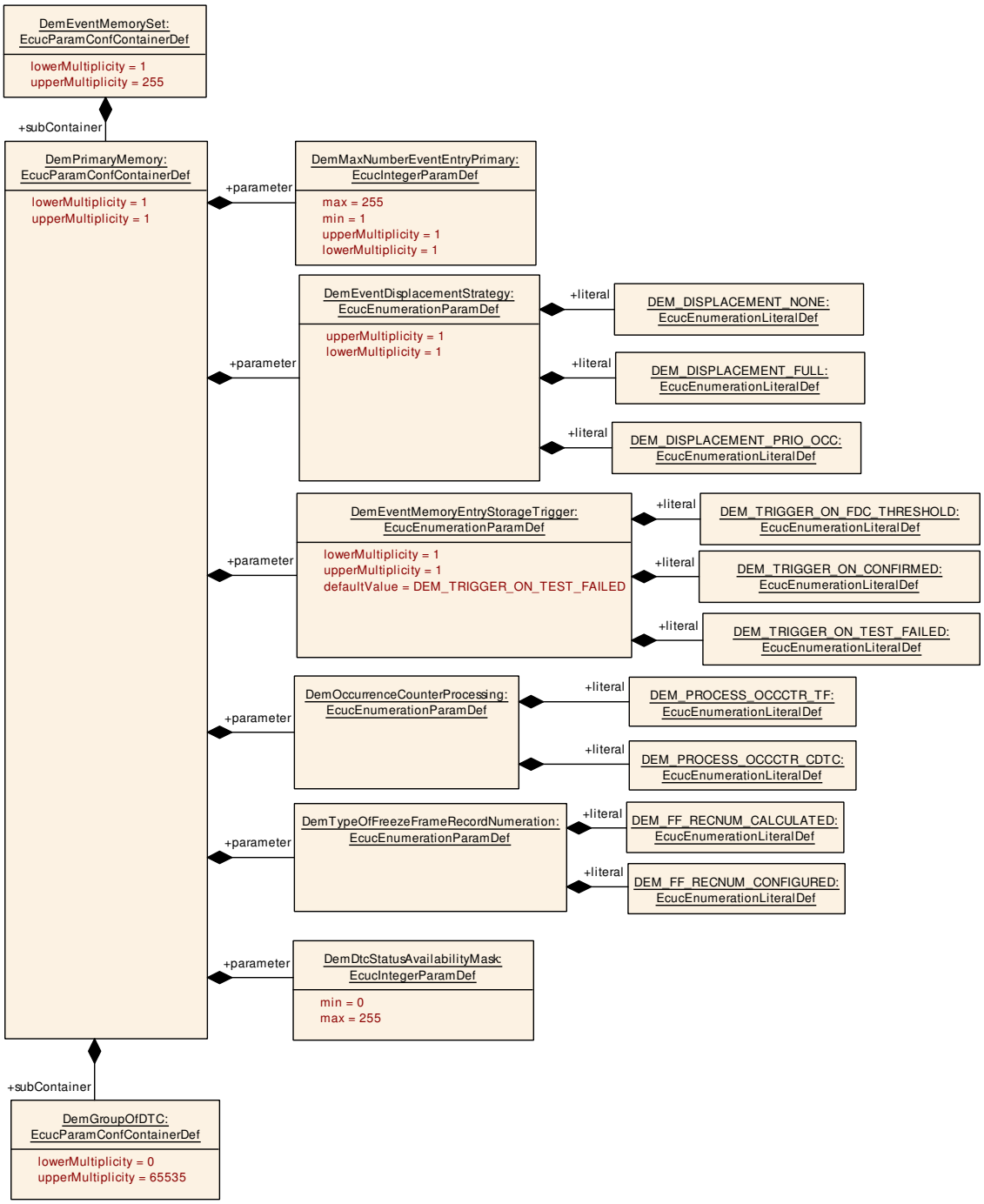


Figure 10.31: DemPrimaryMemory configuration overview

10.2.6.3 DemUserDefinedMemory

[ECUC_Dem_00910] Definition of EcucParamConfContainerDef DemUserDefined Memory ⌈

Container Name	DemUserDefinedMemory
Parent Container	DemEventMemorySet
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This container contains the user defined event memory specific parameters of the Dem module.
Multiplicity	0..30
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
DemAffectedByControlDTCSetting	0..1	[ECUC_Dem_00993]
DemDtcStatusAvailabilityMask	1	[ECUC_Dem_00978]
DemEventDisplacementStrategy	1	[ECUC_Dem_00980]
DemEventMemoryEntryStorageTrigger	1	[ECUC_Dem_00982]
DemMaxNumberEventEntryUserDefined	1	[ECUC_Dem_00691]
DemOccurrenceCounterProcessing	1	[ECUC_Dem_00984]
DemTypeOfFreezeFrameRecordNumeration	1	[ECUC_Dem_00986]
DemUserDefinedMemoryIdentifier	1	[ECUC_Dem_00903]

Included Containers		
Container Name	Multiplicity	Description
DemGroupOfDTC	0..65535	This container contains the configuration (parameters) for DTC groups.

⌋

[ECUC_Dem_00993] Definition of EcucBooleanParamDef DemAffectedByControlDTCSetting ⌈

Parameter Name	DemAffectedByControlDTCSetting
Parent Container	DemUserDefinedMemory
Upstream mapping defined	–
Description	This attribute configures how the content of the memory is affected by an active ControlDTCSetting or not. If the attribute is set to true, the user-defined fault memory is not updated if ControlDTCSetting is off. If the attribute is set to false, the user defined fault memory is updated even if ControlDTCSetting is off.
Multiplicity	0..1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

For parameter table [ECUC_Dem_00978] DemDtcStatusAvailabilityMask, see definition below container DemPrimaryMemory.

For parameter table [ECUC_Dem_00980] DemEventDisplacementStrategy, see definition below container DemPrimaryMemory.

For parameter table [ECUC_Dem_00982] DemEventMemoryEntryStorageTrigger, see definition below container DemPrimaryMemory.

[ECUC_Dem_00691] Definition of EcucIntegerParamDef DemMaxNumberEvent EntryUserDefined ⌈

Parameter Name	DemMaxNumberEventEntryUserDefined	
Parent Container	DemUserDefinedMemory	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Maximum number of events which can be stored in the user defined memory.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	1 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

For parameter table [ECUC_Dem_00984] DemOccurrenceCounterProcessing, see definition below container DemPrimaryMemory.

For parameter table [ECUC_Dem_00986] DemTypeOfFreezeFrameRecordNumeration, see definition below container DemPrimaryMemory.

[ECUC_Dem_00903] Definition of EcucIntegerParamDef DemUserDefinedMemoryIdentifier ⌈

Parameter Name	DemUserDefinedMemoryIdentifier	
Parent Container	DemUserDefinedMemory	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Identifier used by external tester to identify the User defined event memory.	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

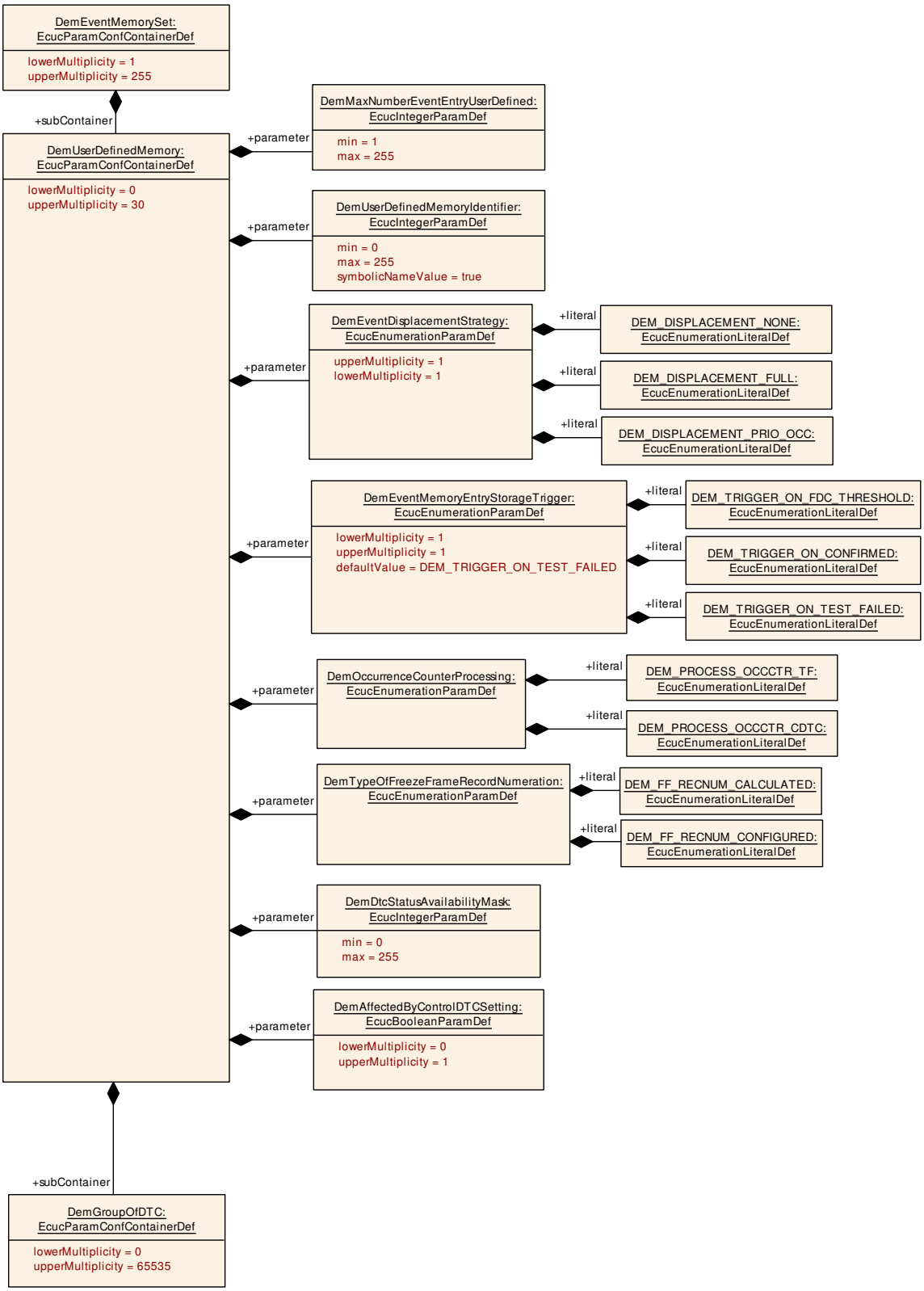


Figure 10.32: DemUserDefinedMemory configuration overview

10.2.7 Debouncing

10.2.7.1 DemDebounceAlgorithmClass

[ECUC_Dem_00604] Definition of EcucChoiceContainerDef DemDebounceAlgorithmClass ⌈

Choice Container Name	DemDebounceAlgorithmClass
Parent Container	DemEventParameter
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate], [22, CP_TPS_SoftwareComponent Template]
Description	Debounce algorithm class: counter based, time based, or monitor internal.
Multiplicity	1
Pre-Compile only	true

No Included Parameters

Container Choices		
Container Name	**Multiplicity**	**Description**
DemDebounceCounterBased	0..1	This container contains the configuration (parameters) for counter based debouncing.
DemDebounceMonitorInternal	0..1	This container contains the configuration (parameters) for monitor internal debouncing.
DemDebounceTimeBased	0..1	This container contains the configuration (parameters) for time based debouncing.

⌋

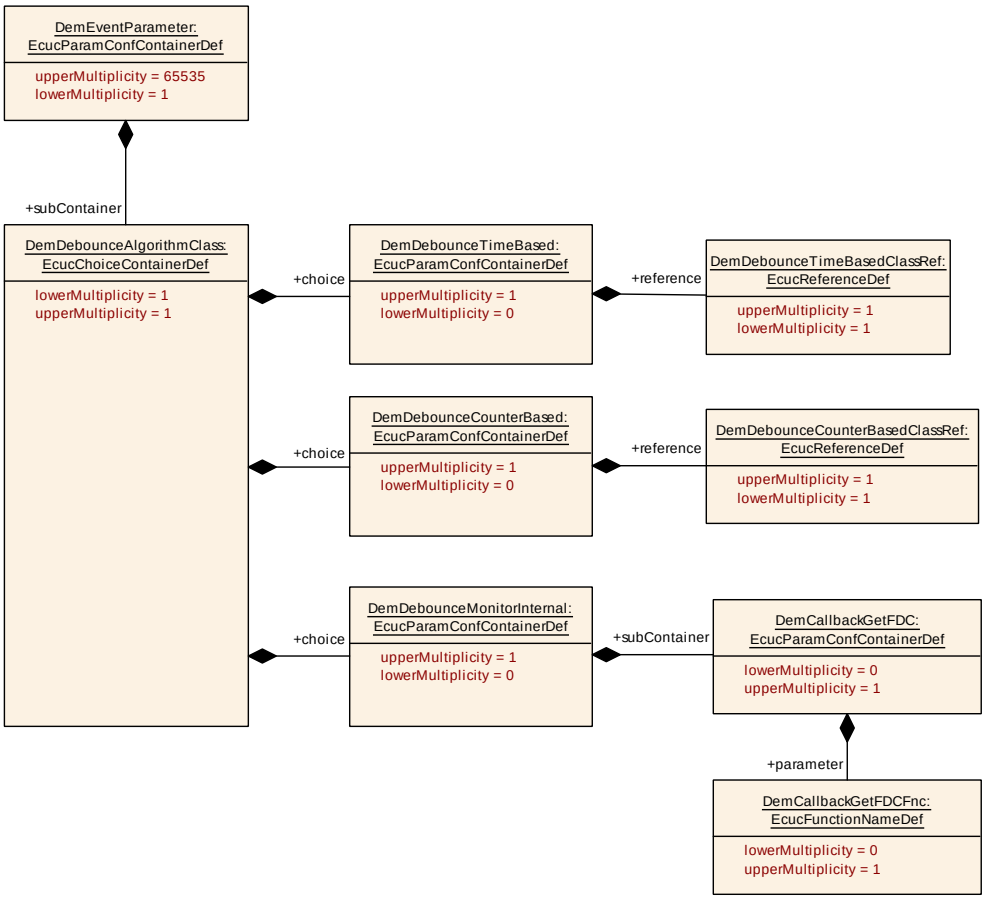


Figure 10.33: DemDebounceAlgorithmClass configuration overview

10.2.7.2 DemDebounceCounterBased

[ECUC_Dem_00711] Definition of EcucParamConfContainerDef DemDebounceCounterBased ⌈

Container Name	DemDebounceCounterBased
Parent Container	DemDebounceAlgorithmClass
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate], [22, CP_TPS_SoftwareComponent Template]
Description	This container contains the configuration (parameters) for counter based debouncing.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDebounceCounterBasedClassRef	1	[ECUC_Dem_00883]

No Included Containers

⌋

[ECUC_Dem_00883] Definition of EcucReferenceDef DemDebounceCounterBasedClassRef ⌈

Parameter Name	DemDebounceCounterBasedClassRef	
Parent Container	DemDebounceCounterBased	
Upstream mapping defined	–	
Description	This reference selects the DemDebounceCounterBasedClass applied for the debouncing of the DemEventParameter.	
Multiplicity	1	
Type	Reference to DemDebounceCounterBasedClass	
Post-Build Value	Loadable	Selectable

⌋

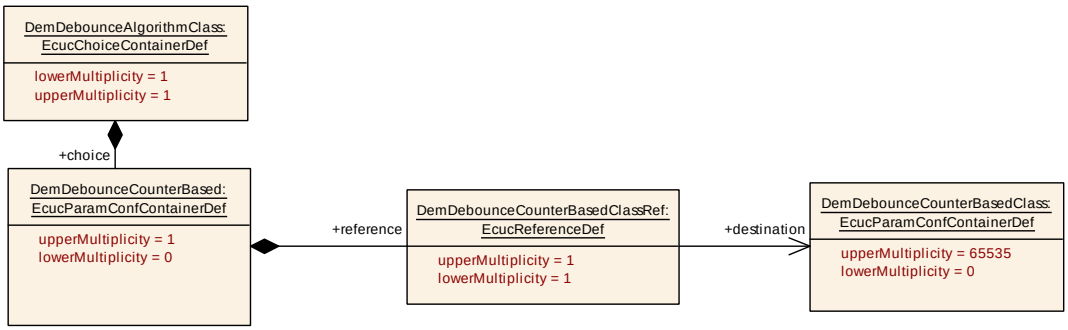


Figure 10.34: DemDebounceCounterBased configuration overview

10.2.7.3 DemDebounceCounterBasedClass

[ECUC_Dem_00881] Definition of EcucParamConfContainerDef DemDebounceCounterBasedClass ⌈

Container Name	DemDebounceCounterBasedClass
Parent Container	DemConfigSet
Upstream mapping defined	[22, CP_TPS_SoftwareComponentTemplate]
Description	This container contains the configuration of Debounce Counter Based Class
Multiplicity	0..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemCounterBasedFdcThresholdStorageValue	0..1	[ECUC_Dem_00914]
DemDebounceBehavior	1	[ECUC_Dem_00786]
DemDebounceCounterDecrementStepSize	1	[ECUC_Dem_00635]
DemDebounceCounterFailedThreshold	1	[ECUC_Dem_00618]
DemDebounceCounterIncrementStepSize	1	[ECUC_Dem_00637]
DemDebounceCounterJumpDown	1	[ECUC_Dem_00685]

▽

△

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDebounceCounterJumpDownValue	1	[ECUC_Dem_00638]
DemDebounceCounterJumpUp	1	[ECUC_Dem_00686]
DemDebounceCounterJumpUpValue	1	[ECUC_Dem_00639]
DemDebounceCounterPassedThreshold	1	[ECUC_Dem_00636]
DemDebounceCounterStorage	1	[ECUC_Dem_00791]

No Included Containers

⌋

[ECUC_Dem_00914] Definition of EcucIntegerParamDef DemCounterBasedFdcThresholdStorageValue ⌈

Parameter Name	DemCounterBasedFdcThresholdStorageValue	
Parent Container	DemDebounceCounterBasedClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Threshold to allocate an event memory entry and to capture the Freeze Frame.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	1 .. 32767	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00786] Definition of EcucEnumerationParamDef DemDebounceBehavior ⌈

Parameter Name	DemDebounceBehavior	
Parent Container	DemDebounceCounterBasedClass, DemDebounceTimeBasedClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	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.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_DEBOUNCE_FREEZE	The event debounce counter/timer will be frozen with the current value and will not change while a related enable condition is not fulfilled or Control DTCSetting 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. Set EventStatus).

▽

△

	DEM_DEBOUNCE_RESET	The event debounce counter/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.
Default value	DEM_DEBOUNCE_RESET	
Pre-Compile only	true	

⌋

[ECUC_Dem_00635] Definition of EcucIntegerParamDef DemDebounceCounterDecrementStepSize ⌈

Parameter Name	DemDebounceCounterDecrementStepSize	
Parent Container	DemDebounceCounterBasedClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Defines the step size for decrementation of the internal debounce counter (PREPASSED).	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	1 .. 32768	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00618] Definition of EcucIntegerParamDef DemDebounceCounterFailedThreshold ⌈

Parameter Name	DemDebounceCounterFailedThreshold	
Parent Container	DemDebounceCounterBasedClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Defines the value of the internal debounce counter, which indicates the failed status.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	1 .. 32767	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00637] Definition of EcucIntegerParamDef DemDebounceCounterIncrementStepSize ⌈

Parameter Name	DemDebounceCounterIncrementStepSize
Parent Container	DemDebounceCounterBasedClass
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	Defines the step size for incrementation of the internal debounce counter (PREFAILED).
Multiplicity	1

▽

△

Type	EcucIntegerParamDef	
Range	1 .. 32767	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00685] Definition of EcucBooleanParamDef DemDebounceCounterJumpDown ⌈

Parameter Name	DemDebounceCounterJumpDown	
Parent Container	DemDebounceCounterBasedClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate], [22, CP_TPS_SoftwareComponentTemplate]	
Description	Switch for the activation of Jump-Down. true: Jump-Down activated false: Jump-Down deactivated	
Multiplicity	1	
Type	EcucBooleanParamDef	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00638] Definition of EcucIntegerParamDef DemDebounceCounterJumpDownValue ⌈

Parameter Name	DemDebounceCounterJumpDownValue	
Parent Container	DemDebounceCounterBasedClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Jump-Down value of the internal debounce counter which is taken as initialization value for the counter when the respective step-down occurs.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	-32768 .. 32767	
Default value	0	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00686] Definition of EcucBooleanParamDef DemDebounceCounterJumpUp ⌈

Parameter Name	DemDebounceCounterJumpUp
Parent Container	DemDebounceCounterBasedClass
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	Switch for the activation of Jump-Up. true: Jump-Up activated false: Jump-Up deactivated
Multiplicity	1
Type	EcucBooleanParamDef

▽

△

Default value	–	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00639] Definition of EcucIntegerParamDef DemDebounceCounterJumpUpValue ⌈

Parameter Name	DemDebounceCounterJumpUpValue	
Parent Container	DemDebounceCounterBasedClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Jump-Up value of the internal debounce counter which is taken as initialization value for the counter when the respective step-up occurs.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	-32768 .. 32767	
Default value	0	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00636] Definition of EcucIntegerParamDef DemDebounceCounterPassedThreshold ⌈

Parameter Name	DemDebounceCounterPassedThreshold	
Parent Container	DemDebounceCounterBasedClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate], [22, CP_TPS_SoftwareComponentTemplate]	
Description	Defines the value of the internal debounce counter, which indicates the passed status.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	-32768 .. -1	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00791] Definition of EcucBooleanParamDef DemDebounceCounterStorage ⌈

Parameter Name	DemDebounceCounterStorage
Parent Container	DemDebounceCounterBasedClass
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	Switch to store the debounce counter value non-volatile or not. true: debounce counter value shall be stored non-volatile false: debounce counter value is volatile
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false

▽

△

Pre-Compile only	true

⌋

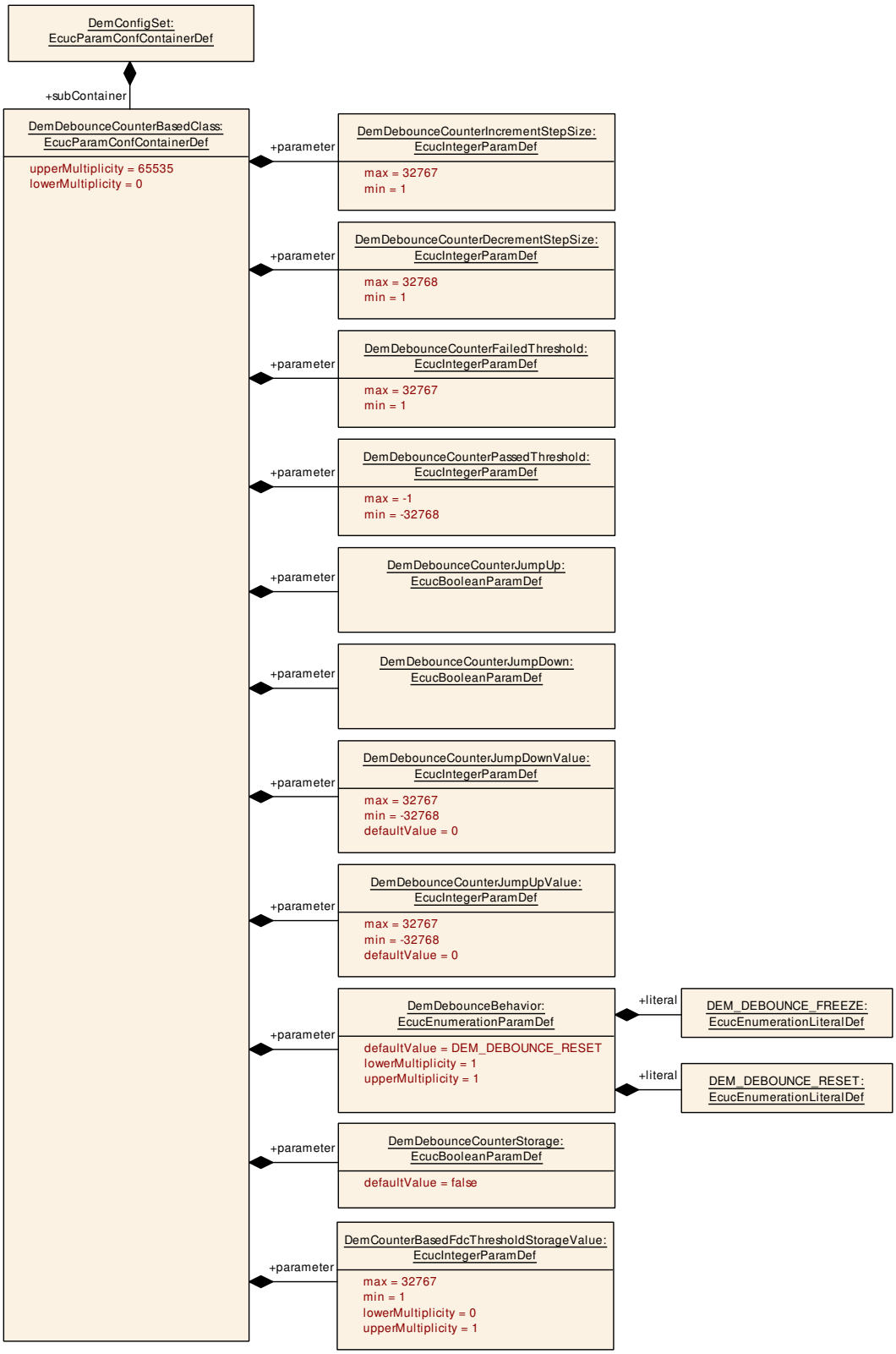


Figure 10.35: DemDebounceCounterBasedClass configuration overview

10.2.7.4 DemDebounceTimeBased

[ECUC_Dem_00713] Definition of EcucParamConfContainerDef DemDebounceTimeBased ⌈

Container Name	DemDebounceTimeBased
Parent Container	DemDebounceAlgorithmClass
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate], [22, CP_TPS_SoftwareComponent Template]
Description	This container contains the configuration (parameters) for time based debouncing.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDebounceTimeBasedClassRef	1	[ECUC_Dem_00891]

No Included Containers

⌋

[ECUC_Dem_00891] Definition of EcucReferenceDef DemDebounceTimeBased ClassRef ⌈

Parameter Name	DemDebounceTimeBasedClassRef	
Parent Container	DemDebounceTimeBased	
Upstream mapping defined	–	
Description	This reference selects the DemDebounceTimeBasedClass applied for the debouncing of the DemEventParameter.	
Multiplicity	1	
Type	Reference to DemDebounceTimeBasedClass	
Post-Build Value	Loadable	Selectable

⌋

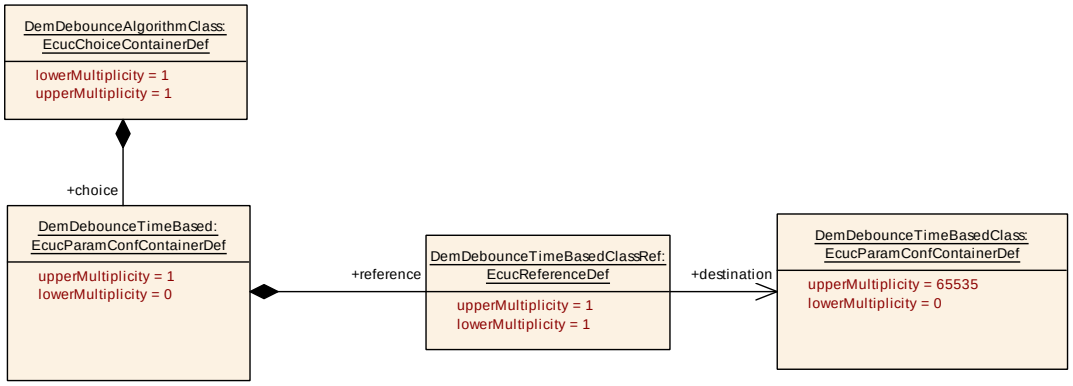


Figure 10.36: DemDebounceTimeBased configuration overview

10.2.7.5 DemDebounceTimeBasedClass

[ECUC_Dem_00882] Definition of EcucParamConfContainerDef DemDebounceTimeBasedClass ⌈

Container Name	DemDebounceTimeBasedClass
Parent Container	DemConfigSet
Upstream mapping defined	–
Description	This container contains the configuration of Debounce Time Based Class.
Multiplicity	0..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDebounceBehavior	1	[ECUC_Dem_00786]
DemDebounceTimeFailedThreshold	1	[ECUC_Dem_00716]
DemDebounceTimePassedThreshold	1	[ECUC_Dem_00717]
DemTimeBasedFdcThresholdStorageValue	0..1	[ECUC_Dem_00915]

No Included Containers

⌋

For parameter table [ECUC_Dem_00786] DemDebounceBehavior, see definition below container DemDebounceCounterBasedClass.

[ECUC_Dem_00716] Definition of EcucFloatParamDef DemDebounceTimeFailedThreshold ⌈

Parameter Name	DemDebounceTimeFailedThreshold	
Parent Container	DemDebounceTimeBasedClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
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. Dem configuration tools must convert this float value to the appropriate value format for the use in the software implementation of Dem.	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	[0.001 .. 3600]	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00717] Definition of EcucFloatParamDef DemDebounceTimePassedThreshold ⌈

Parameter Name	DemDebounceTimePassedThreshold	
Parent Container	DemDebounceTimeBasedClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
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. Dem configuration tools must convert this float value to the appropriate value format for the use in the software implementation of Dem.	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	[0.001 .. 3600]	
Default value	–	
Post-Build Value	Loadable	Selectable

⌋

[ECUC_Dem_00915] Definition of EcucFloatParamDef DemTimeBasedFdcThresholdStorageValue ⌈

Parameter Name	DemTimeBasedFdcThresholdStorageValue	
Parent Container	DemDebounceTimeBasedClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Threshold to allocate an event memory entry and to capture the Freeze Frame.	
Multiplicity	0..1	
Type	EcucFloatParamDef	
Range	[0.001 .. 3600]	
Default value	–	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

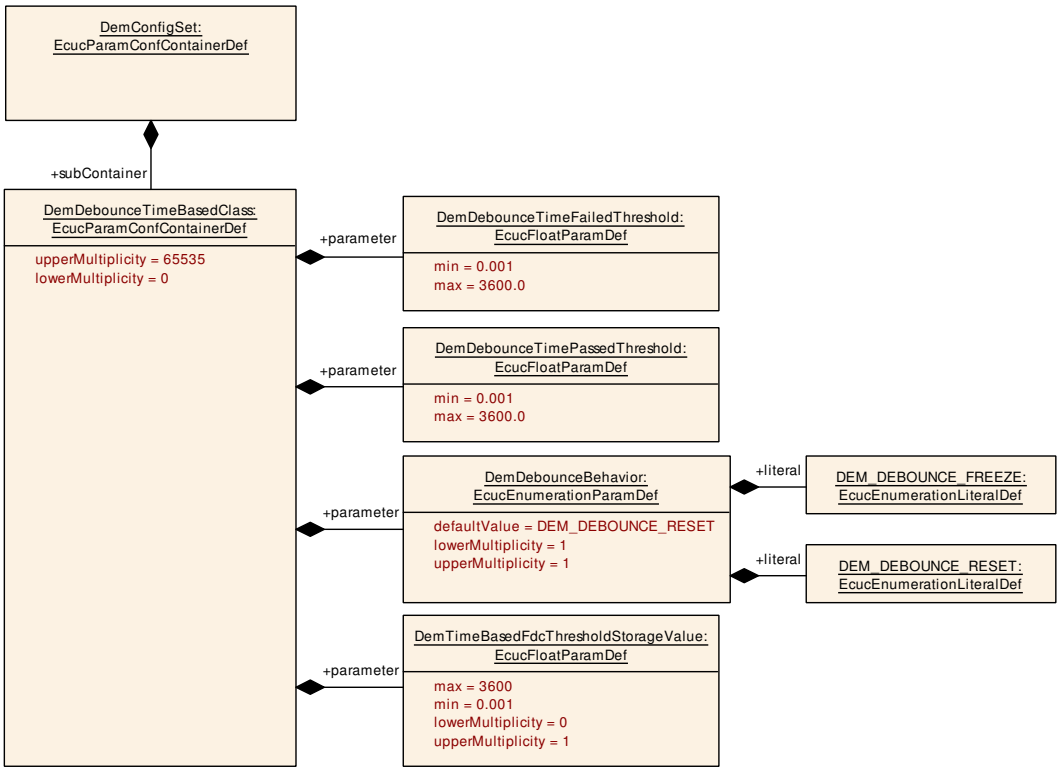


Figure 10.37: DemDebounceTimeBasedClass configuration overview

10.2.7.6 DemDebounceMonitorInternal

[ECUC_Dem_00712] Definition of EcucParamConfContainerDef DemDebounceMonitorInternal ⌈

Container Name	DemDebounceMonitorInternal
Parent Container	DemDebounceAlgorithmClass
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate], [22, CP_TPS_SoftwareComponent Template]
Description	This container contains the configuration (parameters) for monitor internal debouncing.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

No Included Parameters

Included Containers		
Container Name	**Multiplicity**	**Description**
DemCallbackGetFDC	0..1	DemCallbackGetFDC specifies the callback (parameter Dem CallbackGetFDCFnc is present) or R-Port (no parameter Dem CallbackGetFDCFnc is present) to retrieve the fault detection counter value. In case no container is configured, no fault detection counter will be available.

⌋

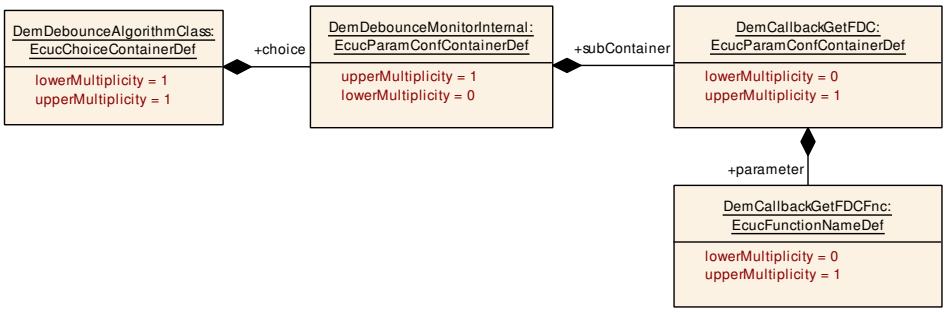


Figure 10.38: DemDebounceMonitorInternal configuration overview

10.2.8 Callbacks

10.2.8.1 DemCallbackClearEventAllowed

[ECUC_Dem_00607] Definition of EcucParamConfContainerDef DemCallbackClearEventAllowed ⌈

Container Name	DemCallbackClearEventAllowed
Parent Container	DemEventParameter
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	The presence of this container indicates that the Dem has access to a "ClearEvent Allowed" callback. In case there is a DemCallbackClearEventAllowedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackClearEventAllowedFnc, the Dem will have an R-Port requiring the interface CallbackClearEventAllowed whose name is generated by using the unique callback-prefix followed by the event name.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemCallbackClearEventAllowedFnc	0..1	[ECUC_Dem_00609]
DemClearEventAllowedBehavior	1	[ECUC_Dem_00788]

No Included Containers

⌋

[ECUC_Dem_00609] Definition of EcucFunctionNameDef DemCallbackClearEventAllowedFnc ⌈

Parameter Name	DemCallbackClearEventAllowedFnc
Parent Container	DemCallbackClearEventAllowed
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]

▽

△

Description	Function name of prototype "ClearEventAllowed".
Multiplicity	0..1
Type	EcucFunctionNameDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

[ECUC_Dem_00788] Definition of EcucEnumerationParamDef DemClearEventAllowedBehavior ⌈

Parameter Name	DemClearEventAllowedBehavior	
Parent Container	DemCallbackClearEventAllowed	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Defines the resulting UDS status byte for the related event, which must not be cleared according to the ClearEventAllowed callback.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_NO_STATUS_BYTE_CHANGE	The UDS status byte keeps unchanged.
	DEM_ONLY_THIS_CYCLE_AND_READINESS	The <...>ThisOperationCycle and readiness bits of the UDS status byte are reset.
Default value	DEM_NO_STATUS_BYTE_CHANGE	
Pre-Compile only	true	

⌋

10.2.8.2 DemCallbackDTCStatusChanged

[ECUC_Dem_00626] Definition of EcucParamConfContainerDef DemCallbackDTCStatusChanged ⌈

Container Name	DemCallbackDTCStatusChanged
Parent Container	DemClient
Upstream mapping defined	–
Description	The presence of this container indicates, that the Dem has access to a "DTCStatus Changed" callback, which the Dem will call to notify other components about the change in the status of a DTC. In case there is a DemCallbackDTCStatusChangedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackDTCStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackDTCStatusChanged. Status change notifications are supported for DTCs in primary memory only.
Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemCallbackDTCStatusChangedFnc	0..1	[ECUC_Dem_00627]

No Included Containers

⌋

[ECUC_Dem_00627] Definition of EcucFunctionNameDef DemCallbackDTCStatusChangedFnc ⌈

Parameter Name	DemCallbackDTCStatusChangedFnc
Parent Container	DemCallbackDTCStatusChanged
Upstream mapping defined	–
Description	Function name of prototype "DTCStatusChanged".
Multiplicity	0..1
Type	EcucFunctionNameDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

10.2.8.3 DemCallbackJ1939DTCStatusChanged

[ECUC_Dem_00823] Definition of EcucParamConfContainerDef DemCallbackJ1939DTCStatusChanged ⌈

Container Name	DemCallbackJ1939DTCStatusChanged
Parent Container	DemGeneralJ1939
Upstream mapping defined	–
Description	The presence of this container indicates, that the Dem has access to a "DTCStatus Changed" callback, which the Dem will call to notify other components about the change in the status of a DTC. In case there is a DemCallbackDTCStatusChangedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackDTCStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackDTCStatusChanged. Status change notifications are supported for DTCs in primary memory only.
Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemCallbackDTCStatusChangedFnc	0..1	[ECUC_Dem_00824]

No Included Containers

⌋

[ECUC_Dem_00824] Definition of EcucFunctionNameDef DemCallbackDTCStatusChangedFnc ⌈

Parameter Name	DemCallbackDTCStatusChangedFnc
Parent Container	DemCallbackJ1939DTCStatusChanged
Upstream mapping defined	–
Description	Function name of prototype "DTCStatusChanged".
Multiplicity	0..1
Type	EcucFunctionNameDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

10.2.8.4 DemCallbackOBDDTCStatusChanged

[ECUC_Dem_00825] Definition of EcucParamConfContainerDef DemCallbackOBDDTCStatusChanged ⌈

Container Name	DemCallbackOBDDTCStatusChanged
Parent Container	DemGeneralOBD
Upstream mapping defined	–
Description	The presence of this container indicates, that the Dem has access to a "DTCStatus Changed" callback, which the Dem will call to notify other components about the change in the status of a DTC. In case there is a DemCallbackDTCStatusChangedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackDTCStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackDTCStatusChanged. Status change notifications are supported for DTCs in primary memory only.
Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemCallbackDTCStatusChangedFnc	0..1	[ECUC_Dem_00826]

No Included Containers

⌋

[ECUC_Dem_00826] Definition of EcucFunctionNameDef DemCallbackDTCStatusChangedFnc ⌈

Parameter Name	DemCallbackDTCStatusChangedFnc
Parent Container	DemCallbackOBDDTCStatusChanged
Upstream mapping defined	–
Description	Function name of prototype "DTCStatusChanged".
Multiplicity	0..1
Type	EcucFunctionNameDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

For details on how this is reflected in the model refer to figure 7.25.

10.2.8.5 DemCallbackGetFDC

[ECUC_Dem_00630] Definition of EcucParamConfContainerDef DemCallbackGetFDC ⌈

Container Name	DemCallbackGetFDC
Parent Container	DemDebounceMonitorInternal
Upstream mapping defined	–
Description	DemCallbackGetFDC specifies the callback (parameter DemCallbackGetFDCFnc is present) or R-Port (no parameter DemCallbackGetFDCFnc is present) to retrieve the fault detection counter value. In case no container is configured, no fault detection counter will be available.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemCallbackGetFDCFnc	0..1	[ECUC_Dem_00631]

No Included Containers

⌋

[ECUC_Dem_00631] Definition of EcucFunctionNameDef DemCallbackGetFDCFnc ⌈

Parameter Name	DemCallbackGetFDCFnc
Parent Container	DemCallbackGetFDC
Upstream mapping defined	–

▽

△

Description	This parameter defines the name of the function that the Dem will call to retrieve the fault-detection counter value from a complex driver.
Multiplicity	0..1
Type	EcucFunctionNameDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

10.2.8.6 DemCallbackEventDataChanged

[ECUC_Dem_00606] Definition of EcucParamConfContainerDef DemCallbackEventDataChanged ⌈

Container Name	DemCallbackEventDataChanged
Parent Container	DemEventParameter
Upstream mapping defined	–
Description	The presence of this container indicates that the Dem has access to an "EventDataChanged" callback. In case there is a DemCallbackEventDataChangedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackEventDataChangedFnc, the Dem will have an R-Port requiring the interface CallbackEventDataChanged whose name is generated by using the unique callback-prefix followed by the event name.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemCallbackEventDataChangedFnc	0..1	[ECUC_Dem_00608]

No Included Containers

⌋

[ECUC_Dem_00608] Definition of EcucFunctionNameDef DemCallbackEventDataChangedFnc ⌈

Parameter Name	DemCallbackEventDataChangedFnc
Parent Container	DemCallbackEventDataChanged
Upstream mapping defined	–
Description	Function name of prototype "EventDataChanged"
Multiplicity	0..1
Type	EcucFunctionNameDef

▽

△

Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

10.2.8.7 DemCallbackEventUdsStatusChanged

[ECUC_Dem_00628] Definition of EcucParamConfContainerDef DemCallbackEventUdsStatusChanged ⌈

Container Name	DemCallbackEventUdsStatusChanged
Parent Container	DemEventParameter
Upstream mapping defined	–
Description	The presence of this container indicates, that the Dem has access to an "EventUds StatusChanged" callback, which the Dem will call to notify other components about the change in the status of an event. In case there is a DemCallbackEventUdsStatusChangedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackEventUdsStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackEventUdsStatusChanged, whose name is generated by using the unique callback-prefix followed by the event name.
Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemCallbackEventUdsStatusChangedFnc	0..1	[ECUC_Dem_00629]

No Included Containers

⌋

[ECUC_Dem_00629] Definition of EcucFunctionNameDef DemCallbackEventUdsStatusChangedFnc ⌈

Parameter Name	DemCallbackEventUdsStatusChangedFnc
Parent Container	DemCallbackEventUdsStatusChanged
Upstream mapping defined	–
Description	Function name of prototype "Dem_CBEventUdsStatusChanged"
Multiplicity	0..1
Type	EcucFunctionNameDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

10.2.8.8 DemCallbackMonitorStatusChanged

[ECUC_Dem_00936] Definition of EcucParamConfContainerDef DemCallbackMonitorStatusChanged ⌈

Container Name	DemCallbackMonitorStatusChanged
Parent Container	DemEventParameter
Upstream mapping defined	–
Description	The presence of this container indicates, that the Dem has access to an "MonitorStatus Changed" callback, which the Dem will call to notify other components about the change in the status of an event. In case there is a DemCallbackMonitorStatusChangedFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackMonitorStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackMonitorStatusChanged, whose name is generated by using the unique callback-prefix followed by the event name.
Multiplicity	0..*
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemCallbackMonitorStatusChangedFnc	0..1	[ECUC_Dem_00937]

No Included Containers

⌋

[ECUC_Dem_00937] Definition of EcucFunctionNameDef DemCallbackMonitorStatusChangedFnc ⌈

Parameter Name	DemCallbackMonitorStatusChangedFnc
Parent Container	DemCallbackMonitorStatusChanged
Upstream mapping defined	–
Description	Function name of prototype "<Module>_DemTriggerOnMonitorStatus"
Multiplicity	0..1
Type	EcucFunctionNameDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

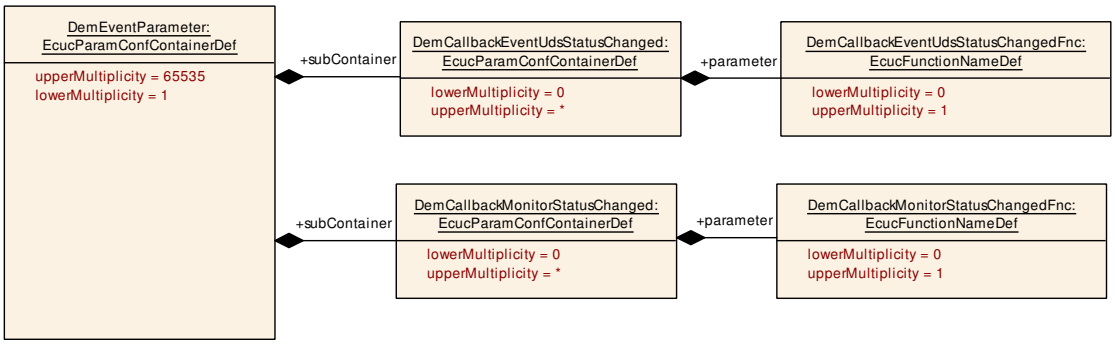


Figure 10.39: Notification of event status changed configuration overview

10.2.8.9 DemCallbackInitMForE

[ECUC_Dem_00632] Definition of EcucParamConfContainerDef DemCallbackInitMForE ⌈

Container Name	DemCallbackInitMForE
Parent Container	DemEventParameter
Upstream mapping defined	–
Description	The presence of this container indicates, that the Dem has access to an "InitMonitorForEvent" callback, which the Dem will call to initialize a monitor. In case the container has a DemCallbackInitMForEFnc, this parameter defines the name of the function that the Dem will call. In case there is no DemCallbackInitMForEFnc, the Dem will have an R-Port requiring the interface CallbackInitMonitorForEvent, whose name is generated by using the unique callback-prefix followed by the event name.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemCallbackInitMForEFnc	0..1	[ECUC_Dem_00601]

No Included Containers

⌋

[ECUC_Dem_00601] Definition of EcucFunctionNameDef DemCallbackInitMForEFnc ⌈

Parameter Name	DemCallbackInitMForEFnc
Parent Container	DemCallbackInitMForE
Upstream mapping defined	–
Description	Function name of prototype "InitMonitorForEvent".
Multiplicity	0..1

▽

△

Type	EcucFunctionNameDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

10.2.8.10 DemClearDTCNotification

[ECUC_Dem_00925] Definition of EcucParamConfContainerDef DemClearDTCNotification ⌈

Container Name	DemClearDTCNotification
Parent Container	DemEventMemorySet
Upstream mapping defined	[22, CP_TPS_SoftwareComponentTemplate]
Description	Contains callback function definition which are called during clear DTC operations.
Multiplicity	0..255
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemClearDtcNotificationFnc	0..1	[ECUC_Dem_00926]
DemClearDtcNotificationTime	1	[ECUC_Dem_00927]

No Included Containers

⌋

[ECUC_Dem_00926] Definition of EcucFunctionNameDef DemClearDtcNotificationFnc ⌈

Parameter Name	DemClearDtcNotificationFnc
Parent Container	DemClearDTCNotification
Upstream mapping defined	–
Description	Notification callback function name which is called on a clear DTC operation (refer to <Module>_ClearDtcNotification).
Multiplicity	0..1
Type	EcucFunctionNameDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

[ECUC_Dem_00927] Definition of EcucEnumerationParamDef DemClearDtcNotificationTime ⌈

Parameter Name	DemClearDtcNotificationTime	
Parent Container	DemClearDTCNotification	
Upstream mapping defined	[22, CP_TPS_SoftwareComponentTemplate]	
Description	Configure, whether the callback shall be called on start of a clear or after finishing a clear DTC operation (refer to <Module>_ClearDtcNotification)	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	FINISH	Call callbacks after finishing DTC clear operation
	START	Call callbacks at start of DTC clear operation
Pre-Compile only	true	

⌋

10.2.9 Event related data

See Figure 7.38 for Event related data configuration overview.

10.2.9.1 DemFreezeFrameClass

[ECUC_Dem_00673] Definition of EcucParamConfContainerDef DemFreezeFrameClass ⌈

Container Name	DemFreezeFrameClass
Parent Container	DemConfigSet
Upstream mapping defined	–
Description	This container contains the combinations of DIDs for freeze frames of UDS and emission related diagnostics over UDS.
Multiplicity	0..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDidClassRef	1..255	[ECUC_Dem_00707]

No Included Containers

⌋

[ECUC_Dem_00707] Definition of EcucReferenceDef DemDidClassRef ⌈

Parameter Name	DemDidClassRef	
Parent Container	DemFreezeFrameClass	
Upstream mapping defined	–	
Description	Reference to the DID elements which shall be contained in the freeze frame. **Attributes:** requiresIndex=true	
Multiplicity	1..255	
Type	Reference to DemDidClass	
Post-Build Multiplicity	Loadable	Selectable
Post-Build Value	Loadable	Selectable

⌋

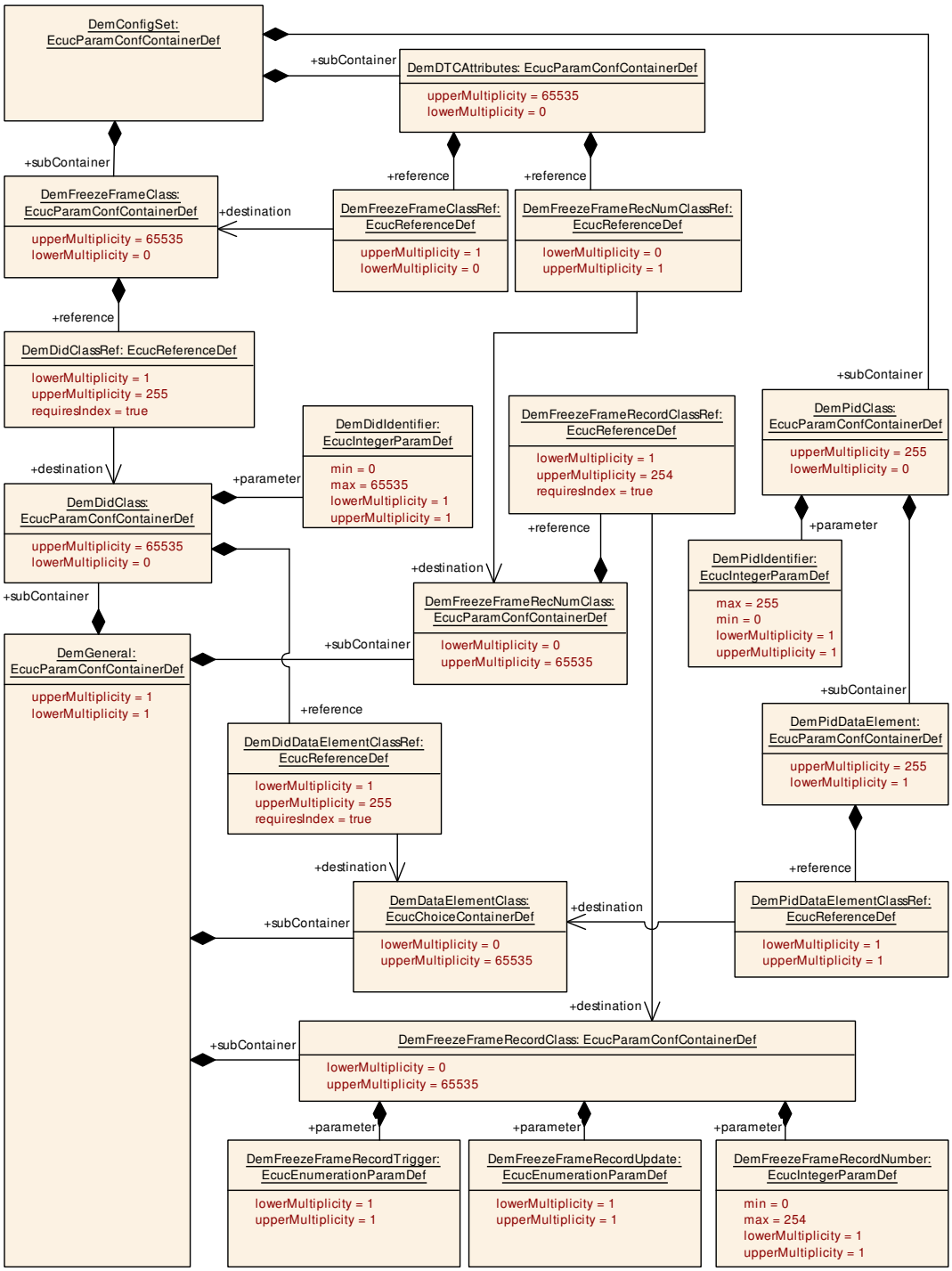


Figure 10.40: DemFreezeFrameClass configuration overview

10.2.9.2 DemFreezeFrameRecordClass

[ECUC_Dem_00801] Definition of EcucParamConfContainerDef DemFreeze FrameRecordClass ⌈

Container Name	DemFreezeFrameRecordClass
Parent Container	DemGeneral
Upstream mapping defined	–
Description	This container contains a list of dedicated, different freeze frame record numbers.
Multiplicity	0..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
DemFreezeFrameRecordNumber	1	[ECUC_Dem_00777]
DemFreezeFrameRecordTrigger	1	[ECUC_Dem_00803]
DemFreezeFrameRecordUpdate	1	[ECUC_Dem_00802]

No Included Containers

⌋

[ECUC_Dem_00777] Definition of EcucIntegerParamDef DemFreezeFrameRecordNumber ⌈

Parameter Name	DemFreezeFrameRecordNumber
Parent Container	DemFreezeFrameRecordClass
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This parameter defines the snapshot record number for a freeze frame. This record number is unique per freeze frame record number class. The number needs to be unique within a fault memory.
Multiplicity	1
Type	EcucIntegerParamDef
Range	0 .. 254
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00803] Definition of EcucEnumerationParamDef DemFreezeFrameRecordTrigger

Status: DRAFT

⌈

Parameter Name	DemFreezeFrameRecordTrigger
Parent Container	DemFreezeFrameRecordClass
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	Defines the trigger to store the FreezeFrameRecord. **Tags:** atp.Status=draft
Multiplicity	1
Type	EcucEnumerationParamDef

▽

△

Range	DEM_TRIGGER_ON_ CONFIRMED	Event Memory entries are triggered if the UDS status bit 3 (confirmedDTC) changes from 0 to 1.
	DEM_TRIGGER_ON_EVERY_ TEST_FAILED	EventMemory entries are triggered every time the event is reported as FAILED or PRE_FAILED results in failed.
	DEM_TRIGGER_ON_FDC_ THRESHOLD	Event Memory entries are triggered when the FDC threshold is reached.
	DEM_TRIGGER_ON_PENDING	Event Memory entries are triggered if the UDS status bit 2 (pendingDTC) changes from 0 to 1.
	DEM_TRIGGER_ON_TEST_ FAILED	Event Memory entries are triggered if the UDS status bit 0 (testFailed) changes from 0 to 1.
	DEM_TRIGGER_ON_TEST_ FAILED_THIS_OPERATION_ CYCLE	Event Memory entries are triggered if the UDS status bit 1 (testFailedThisOperationCycle) changes from 0 to 1.
Pre-Compile only	true	

⌋

[ECUC_Dem_00802] Definition of EcucEnumerationParamDef DemFreezeFrame RecordUpdate ⌈

Parameter Name	DemFreezeFrameRecordUpdate	
Parent Container	DemFreezeFrameRecordClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This parameter defines the case, when the freeze frame record is stored/updated.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_UPDATE_RECORD_NO	This record is only captured for new event memory entries.
	DEM_UPDATE_RECORD_YES	This record is captured every time.
Pre-Compile only	true	

⌋

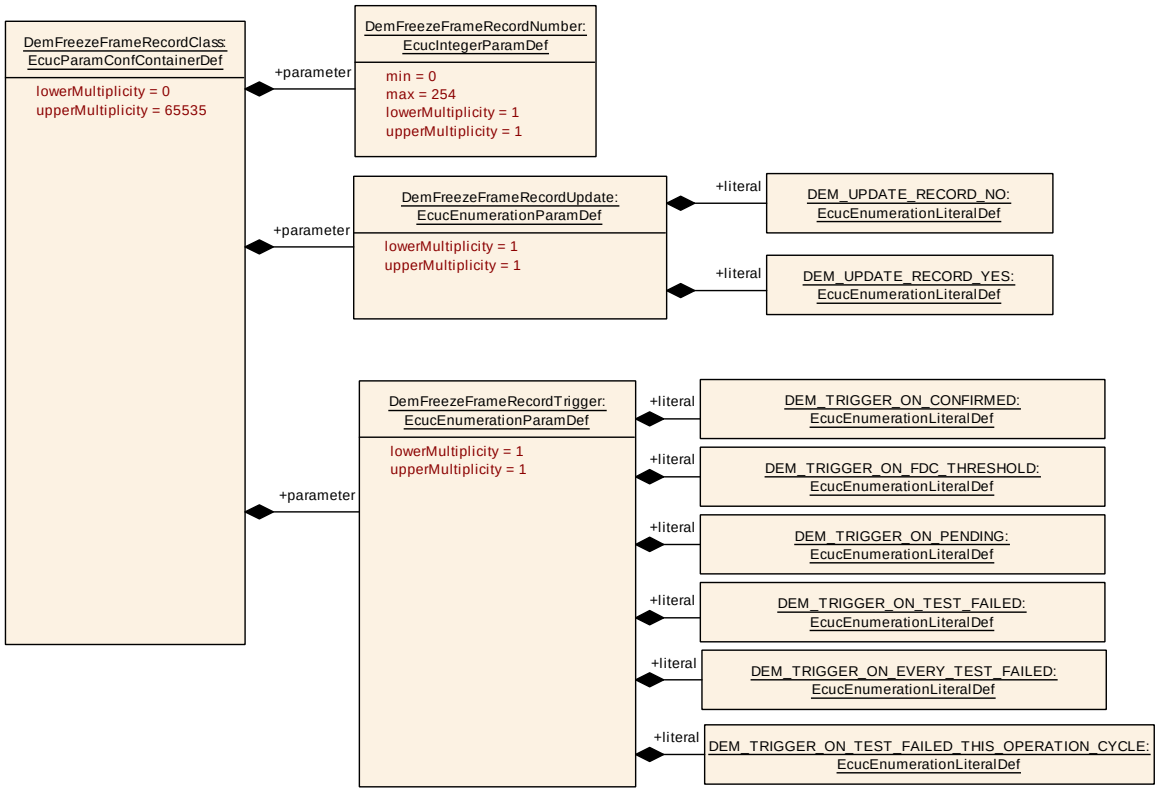


Figure 10.41: DemFreezeFrameRecordClass configuration overview

10.2.9.3 DemFreezeFrameRecNumClass

[ECUC_Dem_00775] Definition of EcucParamConfContainerDef DemFreeze FrameRecNumClass ⌈

Container Name	DemFreezeFrameRecNumClass
Parent Container	DemGeneral
Upstream mapping defined	–
Description	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 according freeze frame records.
Multiplicity	0..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemFreezeFrameRecordClassRef	1..254	[ECUC_Dem_00800]

No Included Containers

⌋

[ECUC_Dem_00800] Definition of EcucReferenceDef DemFreezeFrameRecordClassRef ⌈

Parameter Name	DemFreezeFrameRecordClassRef
Parent Container	DemFreezeFrameRecNumClass
Upstream mapping defined	–
Description	This parameter references record number(s) for a freeze frame record. **Attributes:** requiresIndex=true
Multiplicity	1..254
Type	Reference to DemFreezeFrameRecordClass
Pre-Compile only	true

⌋

10.2.9.4 DemExtendedDataClass

[ECUC_Dem_00664] Definition of EcucParamConfContainerDef DemExtendedDataClass ⌈

Container Name	DemExtendedDataClass
Parent Container	DemGeneral
Upstream mapping defined	–
Description	This class contains the combinations of extended data records for an extended data class.
Multiplicity	0..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemExtendedDataRecordClassRef	1..253	[ECUC_Dem_00774]

No Included Containers

⌋

[ECUC_Dem_00774] Definition of EcucReferenceDef DemExtendedDataRecordClassRef ⌈

Parameter Name	DemExtendedDataRecordClassRef
Parent Container	DemExtendedDataClass
Upstream mapping defined	–
Description	This reference contains the link to an extended data class record. **Attributes:** requiresIndex=true
Multiplicity	1..253
Type	Reference to DemExtendedDataRecordClass
Pre-Compile only	true

⌋

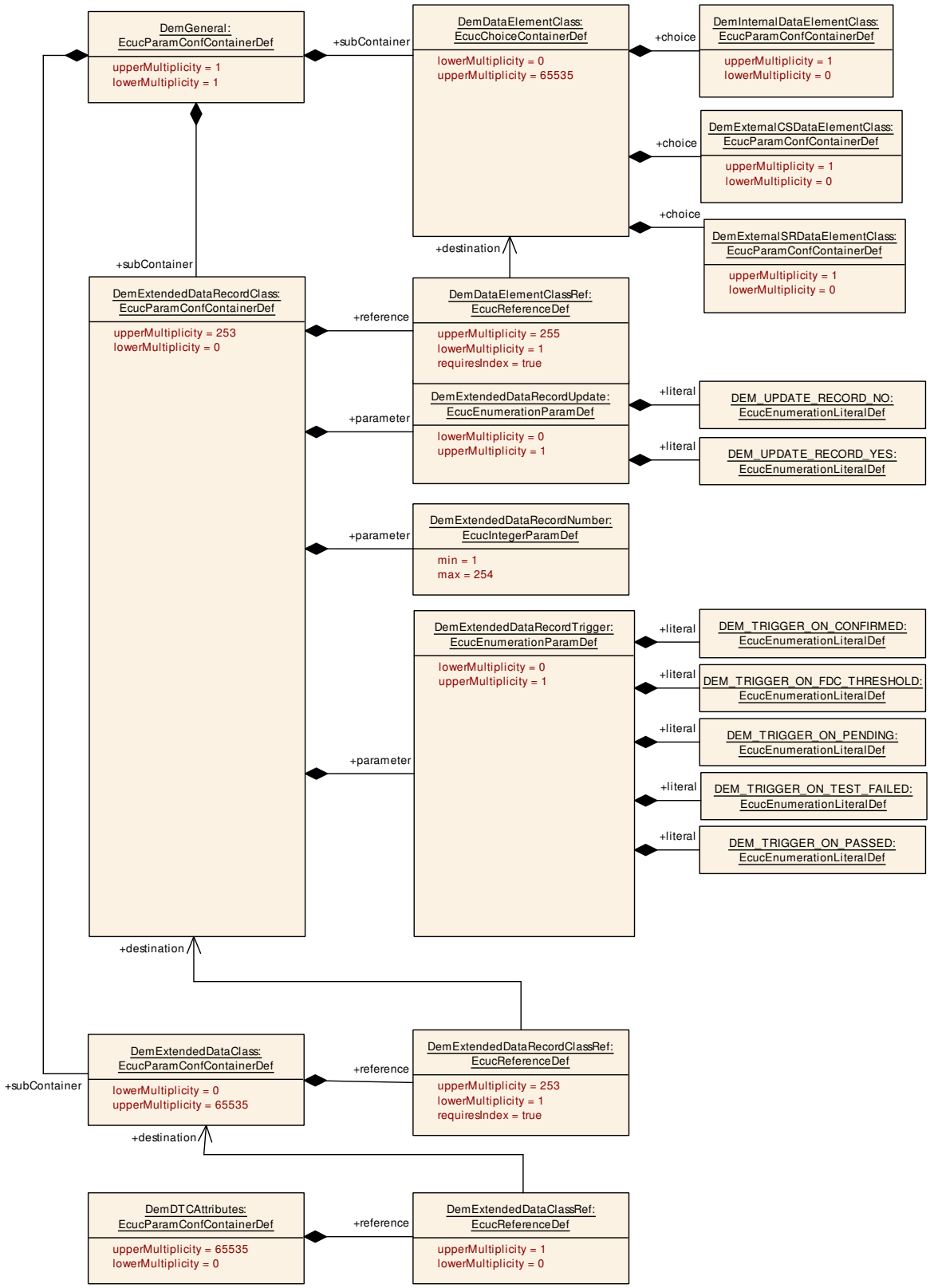


Figure 10.42: DemExtendedDataClass configuration overview

10.2.9.5 DemExtendedDataRecordClass

[ECUC_Dem_00665] Definition of EcucParamConfContainerDef DemExtendedDataRecordClass ⌈

Container Name	DemExtendedDataRecordClass
Parent Container	DemGeneral
Upstream mapping defined	–
Description	This container contains the configuration (parameters) for an extended data record class. It is assembled out of one or several data elements.
Multiplicity	0..253
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemExtendedDataRecordNumber	1	[ECUC_Dem_00666]
DemExtendedDataRecordTrigger	0..1	[ECUC_Dem_00804]
DemExtendedDataRecordUpdate	0..1	[ECUC_Dem_00621]
DemDataElementClassRef	1..255	[ECUC_Dem_00771]

No Included Containers

⌋

[ECUC_Dem_00666] Definition of EcucIntegerParamDef DemExtendedDataRecordNumber ⌈

Parameter Name	DemExtendedDataRecordNumber
Parent Container	DemExtendedDataRecordClass
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This configuration parameter specifies a number for an extended data record. The value is unique within a fault memory. One or more extended data records can be assigned to one diagnostic event/DTC. For primary fault memory, the value 0x00 and the range from 0xF0 to 0xFF are either reserved or predefined. For user defined fault memories, only the value 0x00 is reserved and 0xFF is predefined.
Multiplicity	1
Type	EcucIntegerParamDef
Range	1 .. 254
Default value	–
Pre-Compile only	true

⌋

[ECUC_Dem_00804] Definition of EcucEnumerationParamDef DemExtendedDataRecordTrigger ⌈

Parameter Name	DemExtendedDataRecordTrigger	
Parent Container	DemExtendedDataRecordClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Defines the trigger to store the ExtendedDataRecord.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	DEM_TRIGGER_ON_CONFIRMED	ExtendedDataRecord will be stored when the UDS status confirmed bit changes from 0 to 1.
	DEM_TRIGGER_ON_FDC_THRESHOLD	ExtendedDataRecord will be stored when the FDC reaches its threshold.
	DEM_TRIGGER_ON_PASSED	ExtendedDataRecord will be stored when the event is reported as passed (testFailed bit transition 1 --> 0).
	DEM_TRIGGER_ON_PENDING	ExtendedDataRecord will be stored when the UDS status pending bit changes from 0 to 1.
	DEM_TRIGGER_ON_TEST_FAILED	ExtendedDataRecord will be stored when the UDS status test failed bit changes from 0 to 1.
Pre-Compile only	true	

⌋

[ECUC_Dem_00621] Definition of EcucEnumerationParamDef DemExtendedDataRecordUpdate ⌈

Parameter Name	DemExtendedDataRecordUpdate	
Parent Container	DemExtendedDataRecordClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This extended data record is captured if the configured trigger condition in "Dem ExtendedDataRecordTrigger" is fulfilled.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	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.
Pre-Compile only	true	

⌋

[ECUC_Dem_00771] Definition of EcucReferenceDef DemDataElementClassRef ⌈

Parameter Name	DemDataElementClassRef
Parent Container	DemExtendedDataRecordClass
Upstream mapping defined	–
Description	This reference contains the link to a data element class. **Attributes:** requiresIndex=true
Multiplicity	1..255
Type	Reference to DemDataElementClass

▽

△

Pre-Compile only	true

⌋

10.2.10 Data elements

10.2.10.1 DemDataElementClass

[ECUC_Dem_00610] Definition of EcucChoiceContainerDef DemDataElement Class ⌈

Choice Container Name	DemDataElementClass
Parent Container	DemGeneral
Upstream mapping defined	–
Description	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.
Multiplicity	0..65535
Pre-Compile only	true

No Included Parameters

Container Choices		
Container Name	**Multiplicity**	**Description**
DemExternalCSDataElementClass	0..1	This container contains the configuration (parameters) for an external client/server based data element class. 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 port or a C function-call is used, is defined by DemDataElementUsePort.
DemExternalSRDataElementClass	0..1	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.
DemInternalDataElementClass	0..1	This container contains the configuration (parameters) for an internal data element class.

⌋

See Figure 7.39 for `DemDataElementClass` configuration overview.

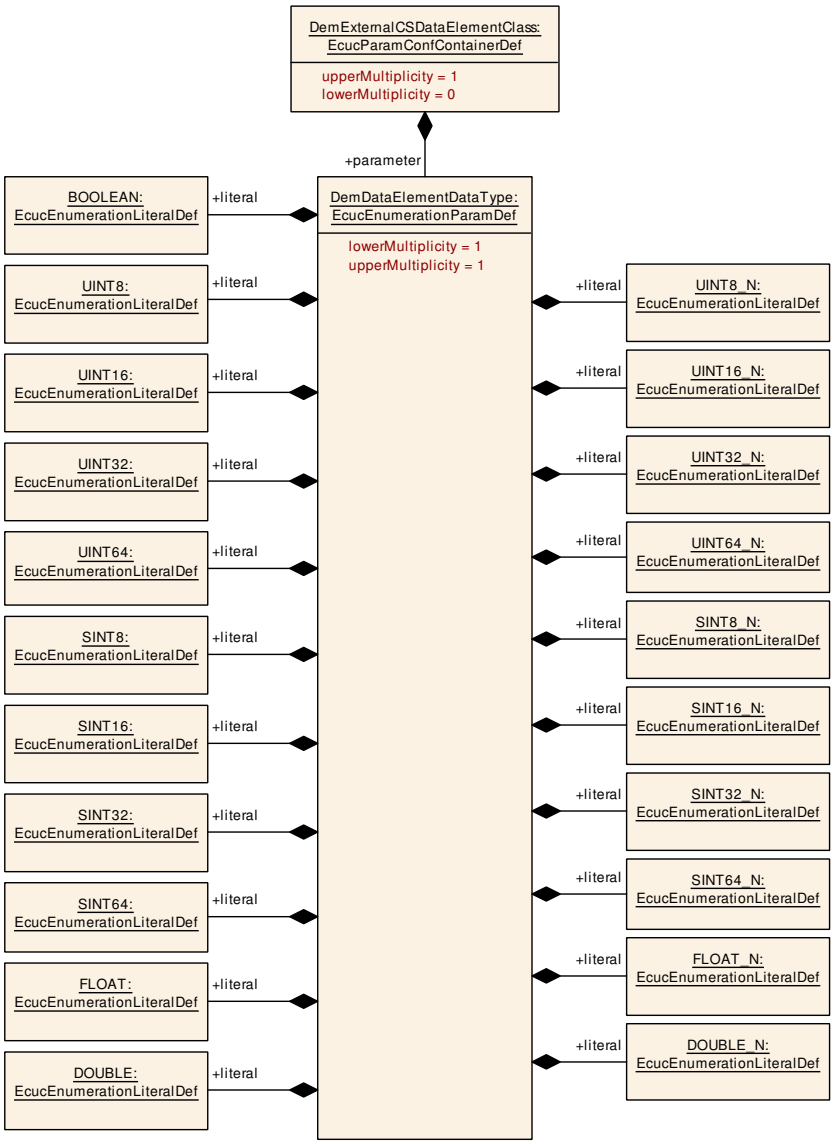


Figure 10.43: DemDataElementDataType configuration overview

10.2.10.2 DemDataElementInstance

[ECUC_Dem_00875] Definition of EcucParamConfContainerDef DemDataElementInstance ⌈

Container Name	DemDataElementInstance
Parent Container	DemSRDataElementClass
Upstream mapping defined	–
Description	Instance Reference to the primitive data in a port where the data element is typed with an ApplicationPrimitveDataType or an ImplementationDataType.
Multiplicity	0..1

▽

△

Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDataElementInstanceRef	1	[ECUC_Dem_00861]

No Included Containers

⌋

[ECUC_Dem_00861] Definition of EcucInstanceReferenceDef DemDataElement InstanceRef ⌈

Parameter Name	DemDataElementInstanceRef
Parent Container	DemDataElementInstance
Upstream mapping defined	–
Description	Instance Reference to the primitive data which shall be read or written. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationPrimitiveDataType of category VALUE or BOOLEAN or if the AutosarDataPrototype is typed with a ImplementationDataType of category VALUE or TYPE_REFERENCE that in turn boils down to VALUE
Multiplicity	1
Type	Instance reference to AUTOSAR-DATA-PROTOTYPE context: ROOT-SW-COMPOSITION-PROTOTYPE SW-COMPONENT-PROTOTYPE PORT-PROTOTYPE
Pre-Compile only	true

⌋

10.2.10.3 DemDidClass

[ECUC_Dem_00706] Definition of EcucParamConfContainerDef DemDidClass ⌈

Container Name	DemDidClass
Parent Container	DemGeneral
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]
Description	This container contains the configuration (parameters) for a data Id class. It is assembled out of one or several data elements.
Multiplicity	0..65535
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDidIdentifier	1	[ECUC_Dem_00650]
DemDidDataElementClassRef	1..255	[ECUC_Dem_00617]

No Included Containers

⌋

[ECUC_Dem_00650] Definition of EcucIntegerParamDef DemDidIdentifier ⌈

Parameter Name	DemDidIdentifier	
Parent Container	DemDidClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	Identifier of the Data ID.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 65535	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00617] Definition of EcucReferenceDef DemDidDataElementClassRef ⌈

Parameter Name	DemDidDataElementClassRef
Parent Container	DemDidClass
Upstream mapping defined	–
Description	This reference contains the link to a data element class. **Attributes:** requiresIndex=true
Multiplicity	1..255
Type	Reference to DemDataElementClass
Pre-Compile only	true

⌋

10.2.10.4 DemInternalDataElementClass

[ECUC_Dem_00684] Definition of EcucParamConfContainerDef DemInternalDataElementClass ⌈

Container Name	DemInternalDataElementClass
Parent Container	DemDataElementClass
Upstream mapping defined	–
Description	This container contains the configuration (parameters) for an internal data element class.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDataElementDataSize	0..1	[ECUC_Dem_00614]
DemInternalDataElement	1	[ECUC_Dem_00616]

No Included Containers

⌋

[ECUC_Dem_00614] Definition of EcucIntegerParamDef DemDataElementDataSize ⌈

Parameter Name	DemDataElementDataSize	
Parent Container	DemInternalDataElementClass	
Upstream mapping defined	–	
Description	Defines the size of the data element in bytes.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	1 .. 255	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00616] Definition of EcucEnumerationParamDef DemInternalDataElement ⌈

Parameter Name	DemInternalDataElement	
Parent Container	DemInternalDataElementClass	
Upstream mapping defined	[23, CP_TPS_DiagnosticExtractTemplate]	
Description	This parameter defines the Dem-internal data value, which is mapped to the data element.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DEM_AGINGCTR_DOWNCNT	map down-counting Dem-internal aging counter, max. range: 1 byte
	DEM_AGINGCTR_UPCNT	map up-counting Dem-internal aging counter, max. range: 1 byte
	DEM_AGINGCTR_UPCNT_FIRST_ACTIVE	Map up-counting Dem-internal aging counter, max. range: 1 byte, starting at 1 when aging starts.
	DEM_CLR_DIST	Distance since diagnostic trouble codes cleared (PID31)
	DEM_CLR_TIME	Time since diagnostic trouble codes cleared (PID4E)
	DEM_CURRENT_FDC	map Dem-internal fault detection counter, max. range: 1 byte
	DEM_CYCLES_SINCE_FIRST_FAILED	map Dem-internal Operation Cycle Counter - Cycles since first failed, max. range: 1 byte
	DEM_CYCLES_SINCE_LAST_FAILED	map Dem-internal Operation Cycle Counter - Cycles since last failed, max. range: 1 byte

▽

△

	DEM_CYCLES_TESTED_SINCE_FIRST_FAILED	map Dem-internal Operation Cycle Counter - Tested cycles since first failed excluding test not complete, max. range: 1 byte
	DEM_CYCLES_TESTED_SINCE_LAST_FAILED	map Dem-internal Operation Cycle Counter - Tested cycles since last failed excluding test not complete, max. range: 1 byte
	DEM_DTC_PRIORITY	DTC priority statically assigned to this DTC via DemDTCPriority. The length is one byte.
	DEM_DTR	Maps the Dem internal DTR test values in SAE J1979-2 format
	DEM_EVENT_ASSOCIATED_IDENTIFICATION	Represents the static value associated to an event by DemEventAssociatedIdentification.
	DEM_FAILED_CYCLES	map Dem-internal Operation Cycle Counter - Failed cycles, max. range: 1 byte
	DEM_IUMPR	Maps the Dem internal IUMPR ratio in SAE J1979-2 format
	DEM_J1939LAMP_STATUS	Current state of applicable lamp(s) for this DTC in SAE J1939 DM31 format.
	DEM_MAX_FDC_DURING_CURRENT_CYCLE	map Dem-internal DTC Fault Detection Counter maximum value during current operation cycle, max. range: 1 byte
	DEM_MAX_FDC_SINCE_LAST_CLEAR	map Dem-internal DTC Fault Detection Counter maximum value since last clear, max. range: 1 byte
	DEM_MIL_DIST	Distance Travelled While MIL is Activated (PID21)
	DEM_MIL_TIME	Time run by the engine while MIL is activated (PID4D)
	DEM_MONITOR_ACTIVITY_DATA	Maps the Dem internal Monitor Activity Data value in SAE J1979-2 format
	DEM_MONITOR_DATA_0	The monitorData0 value provided along with Dem_SetEventStatusWithMonitorData.
	DEM_MONITOR_DATA_1	The monitorData1 value provided along with Dem_SetEventStatusWithMonitorData.
	DEM_OBDDTC_2BYTE	Map OBD number statically assigned to DemObdDTC.DemDtcValue max. range: 2 bytes.
	DEM_OBDDTC_3BYTE	Map OBD number statically assigned to DemObdDTC.DemDtcValue3Byte max. range: 3 bytes.
	DEM_OCCCTR	map Dem-internal occurrence counter, max. range: 1 byte
	DEM_OVFLIND	map Dem-internal overflow indication, max. range: 1 byte (0 = False, 1 = True)
	DEM_SIGNIFICANCE	map (static) Dem-internal event significance (refer to DemDTCSignificance), max. range: 1 byte (0 = OCCURRENCE, 1 = FAULT)
	DEM_WARM_UPS	Number of warm-ups since diagnostic trouble codes cleared (PID30)
Pre-Compile only	true	

⌋

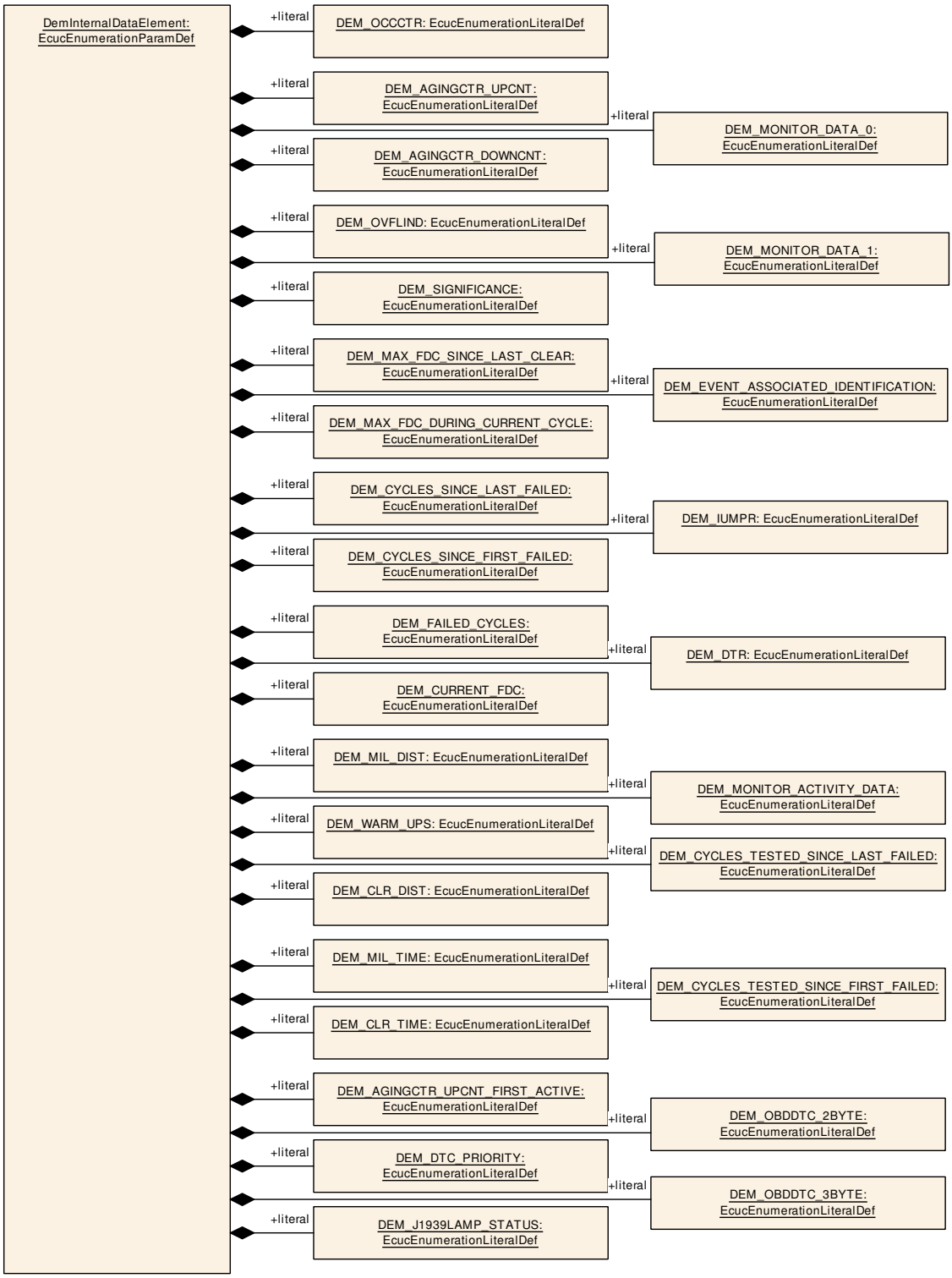


Figure 10.44: DemInternalDataElementClass configuration overview

10.2.10.5 DemExternalCSDataElementClass

[ECUC_Dem_00668] Definition of EcucParamConfContainerDef DemExternalCS-DataElementClass ⌈

Container Name	DemExternalCSDataElementClass
Parent Container	DemDataElementClass
Upstream mapping defined	–
Description	This container contains the configuration (parameters) for an external client/server based data element class. 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 port or a C function-call is used, is defined by DemDataElementUsePort.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDataElementArraySize	0..1	[ECUC_Dem_00949]
DemDataElementDataType	1	[ECUC_Dem_00950]
DemDataElementFncSignature	0..1	[ECUC_Dem_00990]
DemDataElementProvideMonitorData	1	[ECUC_Dem_00951]
DemDataElementReadFnc	0..1	[ECUC_Dem_00619]
DemDataElementUsePort	1	[ECUC_Dem_00647]

No Included Containers

⌋

[ECUC_Dem_00949] Definition of EcucIntegerParamDef DemDataElementArray Size ⌈

Parameter Name	DemDataElementArraySize	
Parent Container	DemExternalCSDataElementClass	
Upstream mapping defined	–	
Description	If the data element is an array data element, it defines the number of elements in the array. Note that this is not the size in bytes.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	1 .. 4294967295	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00950] Definition of EcucEnumerationParamDef DemDataElement DataType ⌈

Parameter Name	DemDataElementDataType	
Parent Container	DemExternalCSDataElementClass, DemExternalSRDataElementClass	
Upstream mapping defined	–	
Description	Provide the implementation data type of this external data element.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BOOLEAN	Type of the data is boolean.
	DOUBLE	Type of the data is double (64 bit float).
	DOUBLE_N	Type of the data is double (64 bit float) array.
	FLOAT	Type of the data is float.
	FLOAT_N	Type of the data is float array.
	SINT16	Type of the data is sint16.
	SINT16_N	Type of the data is sint16 array.
	SINT32	Type of the data is sint32.
	SINT32_N	Type of the data is sint32 array.
	SINT64	Type of the data is sint64.
	SINT64_N	Type of the data is sint64 array.
	SINT8	Type of the data is sint8.
	SINT8_N	Type of the data is sint8 array.
	UINT16	Type of the data is uint16.
	UINT16_N	Type of the data is uint16 array.
	UINT32	Type of the data is uint32.
	UINT32_N	Type of the data is uint32 array.
	UINT64	Type of the data is uint64.
	UINT64_N	Type of the data is uint64 array.
	UINT8	Type of the data is uint8.
	UINT8_N	Type of the data is uint8 array.
Pre-Compile only	true	

⌋

[ECUC_Dem_00990] Definition of EcucEnumerationParamDef DemDataElement FncSignature ⌈

Parameter Name	DemDataElementFncSignature
Parent Container	DemExternalCSDataElementClass
Upstream mapping defined	–
Description	If the parameter is set to DATA_FNC_NORMAL the function omits DataLength and Dcm_NegativeResponseCodeType parameters. If the parameter is set to DATA_FNC_PROXY the generated function call to retrieve the data element uses the signature for Dcm proxy with DataLength and Dcm_NegativeResponseCodeType parameters. Please note that the Dcm proxy is commonly handling the diagnostic data connections for the Dcm and Dem module.
Multiplicity	0..1
Type	EcucEnumerationParamDef

▽

△

Range	DATA_FNC_NORMAL	Function call used for Sw Cluster internal usage.
	DATA_FNC_PROXY	Function call used with generic set of arguments used for Dcm low proxy.
Default value	DATA_FNC_NORMAL	
Pre-Compile only	true	

⌋

[ECUC_Dem_00951] Definition of EcucBooleanParamDef DemDataElementProvideMonitorData ⌈

Parameter Name	DemDataElementProvideMonitorData
Parent Container	DemExternalCSDataElementClass
Upstream mapping defined	[22, CP_TPS_SoftwareComponentTemplate]
Description	If the parameter is set to true, the generated function call to retrieve the data element has the monitorData0 as additional first parameter.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_Dem_00619] Definition of EcucFunctionNameDef DemDataElementReadFnc ⌈

Parameter Name	DemDataElementReadFnc
Parent Container	DemExternalCSDataElementClass
Upstream mapping defined	–
Description	In case of DemDataElementUsePort is false, this parameter defines the prototype of the C function "ReadDataElement" used to get the according value.
Multiplicity	0..1
Type	EcucFunctionNameDef
Default value	–
Regular Expression	–
Pre-Compile only	true

⌋

[ECUC_Dem_00647] Definition of EcucBooleanParamDef DemDataElementUsePort ⌈

Parameter Name	DemDataElementUsePort
Parent Container	DemExternalCSDataElementClass
Upstream mapping defined	–
Description	If the parameter is set to True, a R-Port is generated, to obtain the data element (interface DataServices_{Data}). If the parameter is set to False, the information is obtained by C function-call on another BSW module specified by the parameter Dem DataElementReadFnc.
Multiplicity	1

▽

△

Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

10.2.10.6 DemExternalSRDataElementClass

[ECUC_Dem_00669] Definition of EcucParamConfContainerDef DemExternalSR-DataElementClass ⌈

Container Name	DemExternalSRDataElementClass
Parent Container	DemDataElementClass
Upstream mapping defined	–
Description	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.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDataElementArraySize	0..1	[ECUC_Dem_00967]
DemDataElementDataType	1	[ECUC_Dem_00950]
DemDataElementEndianness	0..1	[ECUC_Dem_00841]

Included Containers		
Container Name	**Multiplicity**	**Description**
DemDiagnosisScaling	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation. Out if this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.
DemSRDataElementClass	0..1	This container defines the source of data in a provided port which shall be read for a external data element This container shall contain either one DemSubElementInData ElementInstance OR DemDataElementInstance reference.

⌋

[ECUC_Dem_00967] Definition of EcucIntegerParamDef DemDataElementArray Size ⌈

Parameter Name	DemDataElementArraySize
Parent Container	DemExternalSRDataElementClass
Upstream mapping defined	–
Description	If the data element is an array data element, it defines the number of elements in the array. Note that this is not the size in bytes.

▽

△

Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	1 .. 4294967295	
Default value	–	
Pre-Compile only	true	

⌋

For parameter table [ECUC_Dem_00950] DemDataElementDataType, see definition below container DemExternalCSDataElementClass.

[ECUC_Dem_00841] Definition of EcucEnumerationParamDef DemDataElement Endianness ⌈

Parameter Name	DemDataElementEndianness	
Parent Container	DemExternalSRDataElementClass	
Upstream mapping defined	–	
Description	Defines the endianness of the data belonging to an external data. If no DemDataElementEndianness is defined the value of DemDataElementDefault Endianness is applicable.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	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
Pre-Compile only	true	

⌋

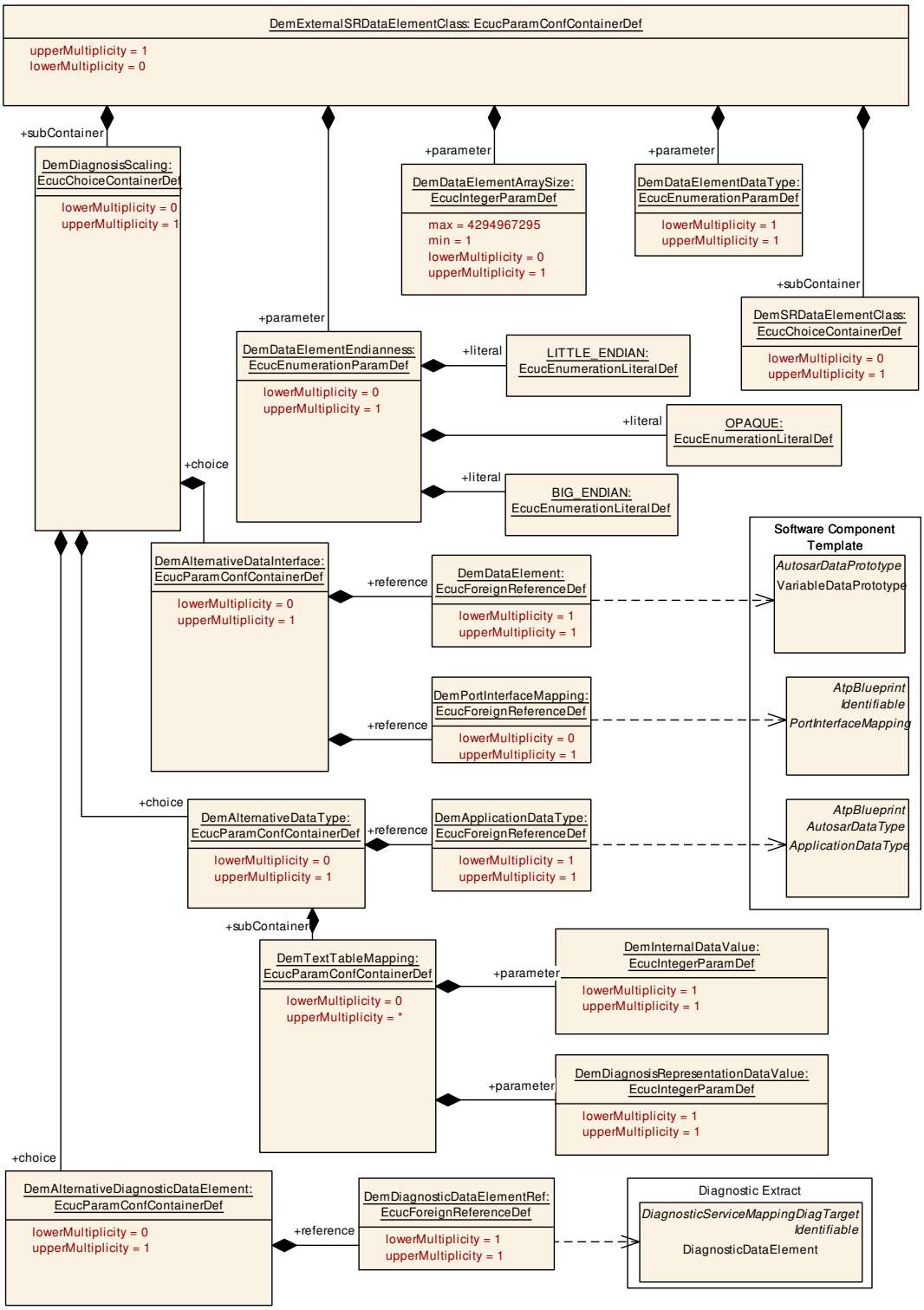


Figure 10.45: DemExternalSRDataElementClass

10.2.10.7 DemSRDataElementClass

[ECUC_Dem_00842] Definition of EcucChoiceContainerDef DemSRDataElement Class ⌈

Choice Container Name	DemSRDataElementClass
Parent Container	DemExternalSRDataElementClass
Upstream mapping defined	–
Description	This container defines the source of data in a provided port which shall be read for a external data element This container shall contain either one DemSubElementInDataElementInstance OR DemDataElementInstance reference.
Multiplicity	0..1
Pre-Compile only	true

No Included Parameters

Container Choices		
Container Name	**Multiplicity**	**Description**
DemDataElementInstance	0..1	Instance Reference to the primitive data in a port where the data element is typed with an ApplicationPrimitveDataType or an ImplementationDataType.
DemSubElementInDataElement Instance	0..1	Instance Reference to the primitve sub-element (at any level) of composite data in a port where the data element is typed with an ApplicationCompositeDataType.

⌋

10.2.10.8 DemSubElementInDataElementInstance

[ECUC_Dem_00874] Definition of EcucParamConfContainerDef DemSubElement InDataElementInstance ⌈

Container Name	DemSubElementInDataElementInstance
Parent Container	DemSRDataElementClass
Upstream mapping defined	–
Description	Instance Reference to the primitve sub-element (at any level) of composite data in a port where the data element is typed with an ApplicationCompositeDataType.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemSubElementInDataElementInstanceRef	1	[ECUC_Dem_00860]

No Included Containers

⌋

[ECUC_Dem_00860] Definition of EcucInstanceReferenceDef DemSubElementInDataElementInstanceRef ⌈

Parameter Name	DemSubElementInDataElementInstanceRef
Parent Container	DemSubElementInDataElementInstance
Upstream mapping defined	–
Description	Instance Reference to the primitve sub-element (at any level) of composite data in a port which shall be read or written. Supported are VariableDataPrototypes in Sender ReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in Parameter Interfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationCompositeDataType.
Multiplicity	1
Type	Instance reference to APPLICATION-COMPOSITE-ELEMENT-DATA-PROTOTYPE context: ROOT-SW-COMPOSITION-PROTOTYPE SW-COMPONENT-PROTOTYPE PORT-PROTOTYPE AUTOSAR-DATA-PROTOTYPE APPLICATION-COMPOSITE-ELEMENT-DATA-PROTOTYPE*
Pre-Compile only	true

⌋

10.2.10.9 DemDiagnosisScaling

[ECUC_Dem_00843] Definition of EcucChoiceContainerDef DemDiagnosisScaling ⌈

Choice Container Name	DemDiagnosisScaling
Parent Container	DemExternalSRDataElementClass
Upstream mapping defined	–
Description	This container contains the configuration (parameters) of an alternative Diagnosis Representation. Out if this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.
Multiplicity	0..1
Pre-Compile only	true

No Included Parameters

Container Choices		
Container Name	**Multiplicity**	**Description**
DemAlternativeDataInterface	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of a Variable DataPrototype in a DataInterface. Additionally a reference to PortInterfaceMapping can be defined which provide already the mapping rules between the Variable DataPrototype in a DataInterface used by the software component (DemSRDataElementClass) and the intended Diagnosis Representation defined by DemExternalSRData ElementClass.

▽

△

Container Choices		
Container Name	**Multiplicity**	**Description**
DemAlternativeDataType	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of an ApplicationDataType. Additionally the definition of a text table mapping can be a defined for ApplicationDataTypes that refers to a CompuMethod of category TEXTTABLE and SCALE_LINEAR_AND_TEXTTABLE.
DemAlternativeDiagnosticDataElement	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of Diagnostic Extract.

⌋

10.2.10.10 DemAlternativeDataInterface

[ECUC_Dem_00844] Definition of EcucParamConfContainerDef DemAlternativeDataInterface ⌈

Container Name	DemAlternativeDataInterface
Parent Container	DemDiagnosisScaling
Upstream mapping defined	–
Description	This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of a VariableDataPrototoype in a DataInterface. Additionally a reference to PortInterfaceMapping can be defined which provide already the mapping rules between the VariableDataPrototoype in a DataInterface used by the software component (DemSRDataElementClass) and the intended Diagnosis Representation defined by DemExternalSRDataElementClass.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDataElement	1	[ECUC_Dem_00845]
DemPortInterfaceMapping	0..1	[ECUC_Dem_00846]

No Included Containers

⌋

[ECUC_Dem_00845] Definition of EcucForeignReferenceDef DemDataElement ⌈

Parameter Name	DemDataElement
Parent Container	DemAlternativeDataInterface
Upstream mapping defined	[22, CP_TPS_SoftwareComponentTemplate]

▽

△

Description	Alternative Diagnosis Representation for the data defined by the means of a Variable DataPrototype in a DataInterface. The CompuMethod of the data type of the referenced VariableDataPrototype will be applied to the data type of the VariableDataPrototype in the interface used by the Dem.
Multiplicity	1
Type	Foreign reference to VARIABLE-DATA-PROTOTYPE
Pre-Compile only	true

⌋

[ECUC_Dem_00846] Definition of EcucForeignReferenceDef DemPortInterface Mapping ⌈

Parameter Name	DemPortInterfaceMapping
Parent Container	DemAlternativeDataInterface
Upstream mapping defined	[22, CP_TPS_SoftwareComponentTemplate]
Description	Optional reference to PortInterfaceMapping which defines the mapping rules. The PortInterfaceMapping is used to get the DataPrototypeMapping that describes a conversion between the data prototype referenced by DemDataElement and the data prototype referenced from DcmDspExternalSRDataElementClass.
Multiplicity	0..1
Type	Foreign reference to PORT-INTERFACE-MAPPING
Pre-Compile only	true

⌋

10.2.10.11 DemAlternativeDiagnosticDataElement

[ECUC_Dem_00934] Definition of EcucParamConfContainerDef DemAlternative DiagnosticDataElement ⌈

Container Name	DemAlternativeDiagnosticDataElement
Parent Container	DemDiagnosisScaling
Upstream mapping defined	–
Description	This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of Diagnostic Extract.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDiagnosticDataElementRef	1	[ECUC_Dem_00935]

No Included Containers

⌋

[ECUC_Dem_00935] Definition of EcucForeignReferenceDef DemDiagnosticDataElementRef ⌈

Parameter Name	DemDiagnosticDataElementRef
Parent Container	DemAlternativeDiagnosticDataElement
Upstream mapping defined	–
Description	Alternative Diagnosis Representation for the data defined by the means of a Diagnostic DataElement in the Diagnostic Extract. This EcucForeignReference enables the access to all SwDataDefProps, in particular BaseType, CompuMethod and DataConstr The CompuMethod and DataConstr that applies to the referenced DiagnosticData Element will be applied to the data type of the VariableDataPrototype in the interface used by the Dem. The mapped ImplementationDataType needs to match the given BaseType.
Multiplicity	1
Type	Foreign reference to DIAGNOSTIC-DATA-ELEMENT
Pre-Compile only	true

⌋

10.2.10.12 DemAlternativeDataType

[ECUC_Dem_00847] Definition of EcucParamConfContainerDef DemAlternativeDataType ⌈

Container Name	DemAlternativeDataType
Parent Container	DemDiagnosisScaling
Upstream mapping defined	–
Description	This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of an ApplicationDataType. Additionally the definition of a text table mapping can be a defined for ApplicationData Types that refers to a CompuMethod of category TEXTTABLE and SCALE_LINEAR_AND_TEXTTABLE.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemApplicationDataType	1	[ECUC_Dem_00848]

Included Containers		
Container Name	**Multiplicity**	**Description**
DemTextTableMapping	0..*	The purpose of the DemDspTextTableMapping is to associate a texttable value defined in the context of the Dem to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DemDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiver Interface for the PortPrototype on the Dem ServiceComponent does not yet exist.

⌋

[ECUC_Dem_00848] Definition of EcucForeignReferenceDef DemApplication DataType ⌈

Parameter Name	DemApplicationDataType
Parent Container	DemAlternativeDataType
Upstream mapping defined	[22, CP_TPS_SoftwareComponentTemplate]
Description	Alternative Diagnosis Representation for the data defined by the means of a ApplicationDataType of category VALUE, BOOLEAN or ARRAY. The CompuMethod that applies to the referenced ApplicationDataType in case of category VALUE or BOOLEAN will be applied to the data type of the VariableData Prototype in the interface used by the Dem.
Multiplicity	1
Type	Foreign reference to APPLICATION-DATA-TYPE
Pre-Compile only	true

⌋

10.2.10.13 DemTextTableMapping

[ECUC_Dem_00849] Definition of EcucParamConfContainerDef DemTextTable Mapping ⌈

Container Name	DemTextTableMapping
Parent Container	DemAlternativeDataType
Upstream mapping defined	–
Description	The purpose of the DemDspTextTableMapping is to associate a texttable value defined in the context of the Dem to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a Sender ReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DemDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dem ServiceComponent does not yet exist.
Multiplicity	0..*

▽

△

Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
DemDiagnosisRepresentationDataValue	1	[ECUC_Dem_00851]
DemInternalDataValue	1	[ECUC_Dem_00850]

No Included Containers

⌋

[ECUC_Dem_00851] Definition of EcucIntegerParamDef DemDiagnosisRepresentationDataValue ⌈

Parameter Name	DemDiagnosisRepresentationDataValue	
Parent Container	DemTextTableMapping	
Upstream mapping defined	–	
Description	The data value in the diagnosis representation.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 18446744073709551615	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_Dem_00850] Definition of EcucIntegerParamDef DemInternalDataValue ⌈

Parameter Name	DemInternalDataValue	
Parent Container	DemTextTableMapping	
Upstream mapping defined	–	
Description	The ECU internal data value.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 18446744073709551615	
Default value	–	
Pre-Compile only	true	

⌋

10.3 Published information

For details refer to [6] Chapter 10.3 *“Published Information”*.

A Not Applicable Requirements

[SWS_Dem_NA_00999]

Upstream requirements: SRS_BSW_00005, SRS_BSW_00161, SRS_BSW_00162, SRS_BSW_00164, SRS_BSW_00168, SRS_BSW_00170, SRS_BSW_00327, SRS_BSW_00331, SRS_BSW_00341, SRS_BSW_00347, SRS_BSW_00348, SRS_BSW_00350, SRS_BSW_00353, SRS_BSW_00357, SRS_BSW_00359, SRS_BSW_00360, SRS_BSW_00374, SRS_BSW_00375, SRS_BSW_00379, SRS_BSW_00433, RS_Diag_04005, RS_Diag_04006, RS_Diag_04015, RS_Diag_04016, RS_Diag_04019, RS_Diag_04020, RS_Diag_04021, RS_Diag_04032, RS_Diag_04033, RS_Diag_04064, RS_Diag_04091, RS_Diag_04097, RS_Diag_04098, RS_Diag_04119, RS_Diag_04120, RS_Diag_04121, RS_Diag_04135, RS_Diag_04136, RS_Diag_04139

⌈These requirements are not applicable to this specification.⌋

B Change history of AUTOSAR traceable items

Please note that the lists in this chapter also include traceable items that have been removed from the specification in a later version. These items do not appear as hyperlinks in the document.

B.1 Traceable item history of this document according to AUTOSAR Release R26-11

B.1.1 Added Specification Items in R26-11

Number	Heading
[ECUC_Dem_00996]	Definition of EcucFunctionNameDef DemGeneralCallbackEventUdsStatus ChangedFnc
[SWS_Dem_01426]	General Event UDS Status Changed notification
[SWS_Dem_01427]	Mapping of 3-byte OBD DTC to internal data
[SWS_Dem_01428]	Mapping of 2-byte OBD DTC to internal data
[SWS_Dem_01430]	Clearing an individual J1939 DTC
[SWS_Dem_01431]	Initial PTO state for PID $1E
[SWS_Dem_01432]	Searching for the next filtered J1939 DTC with severity
[SWS_Dem_01433]	Returning a matching J1939 DTC, occurrence counter, and severity
[SWS_Dem_01434]	Returning configured severity for a J1939 DTC
[SWS_Dem_01435]	Returning no-severity when no J1939 DTC severity is configured
[SWS_Dem_91202]	Definition of API function Dem_J1939DcmClearSingleDTC
[SWS_Dem_91203]	Definition of API function Dem_DcmReadDataOfPID1E
[SWS_Dem_91204]	Definition of datatype Dem_J1939DcmDTCSeverityType
[SWS_Dem_91205]	Definition of API function Dem_J1939DcmGetNextFilteredDTCWithSeverity

Table B.1: Added Specification Items in R26-11

B.1.2 Changed Specification Items in R26-11

Number	Heading
[ECUC_Dem_00127]	Definition of EcucEnumerationParamDef DemNvStorageStrategy
[ECUC_Dem_00601]	Definition of EcucFunctionNameDef DemCallbackInitMForEFnc
[ECUC_Dem_00602]	Definition of EcucBooleanParamDef DemConsiderPtoStatus
[ECUC_Dem_00604]	Definition of EcucChoiceContainerDef DemDebounceAlgorithmClass
[ECUC_Dem_00605]	Definition of EcucIntegerParamDef DemMaxNumberFreezeFrameRecords
[ECUC_Dem_00606]	Definition of EcucParamConfContainerDef DemCallbackEventDataChanged

▽

△

Number	Heading
[ECUC_Dem_00607]	Definition of EcucParamConfContainerDef DemCallbackClearEventAllowed
[ECUC_Dem_00608]	Definition of EcucFunctionNameDef DemCallbackEventDataChangedFnc
[ECUC_Dem_00609]	Definition of EcucFunctionNameDef DemCallbackClearEventAllowedFnc
[ECUC_Dem_00610]	Definition of EcucChoiceContainerDef DemDataElementClass
[ECUC_Dem_00614]	Definition of EcucIntegerParamDef DemDataElementDataSize
[ECUC_Dem_00616]	Definition of EcucEnumerationParamDef DemInternalDataElement
[ECUC_Dem_00617]	Definition of EcucReferenceDef DemDidDataElementClassRef
[ECUC_Dem_00618]	Definition of EcucIntegerParamDef DemDebounceCounterFailedThreshold
[ECUC_Dem_00619]	Definition of EcucFunctionNameDef DemDataElementReadFnc
[ECUC_Dem_00621]	Definition of EcucEnumerationParamDef DemExtendedDataRecordUpdate
[ECUC_Dem_00623]	Definition of EcucIntegerParamDef DemAgingCycleCounterThreshold
[ECUC_Dem_00624]	Definition of EcucReferenceDef DemAgingCycleRef
[ECUC_Dem_00625]	Definition of EcucIntegerParamDef DemBswErrorBufferSize
[ECUC_Dem_00626]	Definition of EcucParamConfContainerDef DemCallbackDTCStatusChanged
[ECUC_Dem_00627]	Definition of EcucFunctionNameDef DemCallbackDTCStatusChangedFnc
[ECUC_Dem_00628]	Definition of EcucParamConfContainerDef DemCallbackEventUdsStatus Changed
[ECUC_Dem_00629]	Definition of EcucFunctionNameDef DemCallbackEventUdsStatusChanged Fnc
[ECUC_Dem_00630]	Definition of EcucParamConfContainerDef DemCallbackGetFDC
[ECUC_Dem_00631]	Definition of EcucFunctionNameDef DemCallbackGetFDCFnc
[ECUC_Dem_00632]	Definition of EcucParamConfContainerDef DemCallbackInitMForE
[ECUC_Dem_00634]	Definition of EcucParamConfContainerDef DemConfigSet
[ECUC_Dem_00635]	Definition of EcucIntegerParamDef DemDebounceCounterDecrementStep Size
[ECUC_Dem_00636]	Definition of EcucIntegerParamDef DemDebounceCounterPassedThreshold
[ECUC_Dem_00637]	Definition of EcucIntegerParamDef DemDebounceCounterIncrementStep Size
[ECUC_Dem_00638]	Definition of EcucIntegerParamDef DemDebounceCounterJumpDownValue
[ECUC_Dem_00639]	Definition of EcucIntegerParamDef DemDebounceCounterJumpUpValue
[ECUC_Dem_00641]	Definition of EcucParamConfContainerDef DemDTCAttributes
[ECUC_Dem_00642]	Definition of EcucReferenceDef DemDTCAttributesRef
[ECUC_Dem_00643]	Definition of EcucIntegerParamDef DemDTCFunctionalUnit
[ECUC_Dem_00645]	Definition of EcucEnumerationParamDef DemDTCSeverity
[ECUC_Dem_00647]	Definition of EcucBooleanParamDef DemDataElementUsePort
[ECUC_Dem_00648]	Definition of EcucBooleanParamDef DemDevErrorDetect
[ECUC_Dem_00650]	Definition of EcucIntegerParamDef DemDidIdentifier
[ECUC_Dem_00653]	Definition of EcucParamConfContainerDef DemEnableCondition
[ECUC_Dem_00654]	Definition of EcucIntegerParamDef DemEnableConditionId

▽

△

Number	Heading
[ECUC_Dem_00655]	Definition of EcucReferenceDef DemEnableConditionRef
[ECUC_Dem_00656]	Definition of EcucBooleanParamDef DemEnableConditionStatus
[ECUC_Dem_00659]	Definition of EcucIntegerParamDef DemEventId
[ECUC_Dem_00660]	Definition of EcucEnumerationParamDef DemEventKind
[ECUC_Dem_00661]	Definition of EcucParamConfContainerDef DemEventParameter
[ECUC_Dem_00662]	Definition of EcucIntegerParamDef DemDTCPriority
[ECUC_Dem_00664]	Definition of EcucParamConfContainerDef DemExtendedDataClass
[ECUC_Dem_00665]	Definition of EcucParamConfContainerDef DemExtendedDataRecordClass
[ECUC_Dem_00666]	Definition of EcucIntegerParamDef DemExtendedDataRecordNumber
[ECUC_Dem_00667]	Definition of EcucReferenceDef DemExtendedDataClassRef
[ECUC_Dem_00668]	Definition of EcucParamConfContainerDef DemExternalCSDataElement Class
[ECUC_Dem_00669]	Definition of EcucParamConfContainerDef DemExternalSRDataElement Class
[ECUC_Dem_00671]	Definition of EcucBooleanParamDef DemFFPrestorageSupported
[ECUC_Dem_00673]	Definition of EcucParamConfContainerDef DemFreezeFrameClass
[ECUC_Dem_00674]	Definition of EcucReferenceDef DemFreezeFrameClassRef
[ECUC_Dem_00677]	Definition of EcucParamConfContainerDef DemGeneral
[ECUC_Dem_00678]	Definition of EcucIntegerParamDef DemGroupDTCs
[ECUC_Dem_00679]	Definition of EcucParamConfContainerDef DemGroupOfDTC
[ECUC_Dem_00680]	Definition of EcucParamConfContainerDef DemIndicator
[ECUC_Dem_00681]	Definition of EcucParamConfContainerDef DemIndicatorAttribute
[ECUC_Dem_00682]	Definition of EcucEnumerationParamDef DemIndicatorBehaviour
[ECUC_Dem_00683]	Definition of EcucIntegerParamDef DemIndicatorID
[ECUC_Dem_00684]	Definition of EcucParamConfContainerDef DemInternalDataElementClass
[ECUC_Dem_00685]	Definition of EcucBooleanParamDef DemDebounceCounterJumpDown
[ECUC_Dem_00686]	Definition of EcucBooleanParamDef DemDebounceCounterJumpUp
[ECUC_Dem_00687]	Definition of EcucReferenceDef DemIndicatorRef
[ECUC_Dem_00689]	Definition of EcucIntegerParamDef DemMaxNumberEventEntryPermanent
[ECUC_Dem_00690]	Definition of EcucIntegerParamDef DemMaxNumberEventEntryPrimary
[ECUC_Dem_00691]	Definition of EcucIntegerParamDef DemMaxNumberEventEntryUserDefined
[ECUC_Dem_00692]	Definition of EcucIntegerParamDef DemMaxNumberPrestoredFF
[ECUC_Dem_00696]	Definition of EcucParamConfContainerDef DemNvRamBlockId
[ECUC_Dem_00697]	Definition of EcucReferenceDef DemNvRamBlockIdRef
[ECUC_Dem_00698]	Definition of EcucEnumerationParamDef DemOBDSupport
[ECUC_Dem_00701]	Definition of EcucParamConfContainerDef DemOperationCycle
[ECUC_Dem_00702]	Definition of EcucReferenceDef DemOperationCycleRef
[ECUC_Dem_00704]	Definition of EcucBooleanParamDef DemPTOSupport
[ECUC_Dem_00705]	Definition of EcucIntegerParamDef DemPidIdentifier

▽

△

Number	Heading
[ECUC_Dem_00706]	Definition of EcucParamConfContainerDef DemDidClass
[ECUC_Dem_00707]	Definition of EcucReferenceDef DemDidClassRef
[ECUC_Dem_00711]	Definition of EcucParamConfContainerDef DemDebounceCounterBased
[ECUC_Dem_00712]	Definition of EcucParamConfContainerDef DemDebounceMonitorInternal
[ECUC_Dem_00713]	Definition of EcucParamConfContainerDef DemDebounceTimeBased
[ECUC_Dem_00715]	Definition of EcucFloatParamDef DemTaskTime
[ECUC_Dem_00716]	Definition of EcucFloatParamDef DemDebounceTimeFailedThreshold
[ECUC_Dem_00717]	Definition of EcucFloatParamDef DemDebounceTimePassedThreshold
[ECUC_Dem_00719]	Definition of EcucBooleanParamDef DemTriggerFiMReports
[ECUC_Dem_00720]	Definition of EcucEnumerationParamDef DemTypeOfDTCSupported
[ECUC_Dem_00723]	Definition of EcucReferenceDef DemMILIndicatorRef
[ECUC_Dem_00724]	Definition of EcucBooleanParamDef DemDebounceCounterBasedSupport
[ECUC_Dem_00725]	Definition of EcucBooleanParamDef DemDebounceTimeBasedSupport
[ECUC_Dem_00728]	Definition of EcucParamConfContainerDef DemStorageCondition
[ECUC_Dem_00729]	Definition of EcucParamConfContainerDef DemPidClass
[ECUC_Dem_00730]	Definition of EcucIntegerParamDef DemStorageConditionId
[ECUC_Dem_00731]	Definition of EcucBooleanParamDef DemStorageConditionStatus
[ECUC_Dem_00733]	Definition of EcucReferenceDef DemPidDataElementClassRef
[ECUC_Dem_00734]	Definition of EcucParamConfContainerDef DemRatio
[ECUC_Dem_00735]	Definition of EcucReferenceDef DemDiagnosticEventRef
[ECUC_Dem_00736]	Definition of EcucReferenceDef DemFunctionIdRef
[ECUC_Dem_00737]	Definition of EcucEnumerationParamDef DemIUMPRGroup
[ECUC_Dem_00740]	Definition of EcucEnumerationParamDef DemEventCombinationSupport
[ECUC_Dem_00741]	Definition of EcucEnumerationParamDef DemRatioKind
[ECUC_Dem_00745]	Definition of EcucParamConfContainerDef DemEnableConditionGroup
[ECUC_Dem_00746]	Definition of EcucReferenceDef DemEnableConditionGroupRef
[ECUC_Dem_00748]	Definition of EcucIntegerParamDef DemIndicatorHealingCycleCounter Threshold
[ECUC_Dem_00750]	Definition of EcucIntegerParamDef DemIndicatorFailureCycleCounter Threshold
[ECUC_Dem_00755]	Definition of EcucEnumerationParamDef DemEventOBDReadinessGroup
[ECUC_Dem_00756]	Definition of EcucParamConfContainerDef DemGeneralOBD
[ECUC_Dem_00757]	Definition of EcucChoiceReferenceDef DemOBDInputEngineSpeed
[ECUC_Dem_00758]	Definition of EcucChoiceReferenceDef DemOBDInputVehicleSpeed
[ECUC_Dem_00759]	Definition of EcucChoiceReferenceDef DemOBDInputDistanceInformation
[ECUC_Dem_00760]	Definition of EcucChoiceReferenceDef DemOBDInputProgrammingEvent
[ECUC_Dem_00761]	Definition of EcucChoiceReferenceDef DemOBDInputAmbientTemperature
[ECUC_Dem_00762]	Definition of EcucChoiceReferenceDef DemOBDInputAmbientPressure

▽

△

Number	Heading
[ECUC_Dem_00763]	Definition of EcucChoiceReferenceDef DemOBDInputAcceleratorPedal Information
[ECUC_Dem_00765]	Definition of EcucBooleanParamDef DemTriggerMonitorInitBeforeClearOk
[ECUC_Dem_00766]	Definition of EcucEnumerationParamDef DemClearDTCBehavior
[ECUC_Dem_00768]	Definition of EcucReferenceDef DemStorageConditionRef
[ECUC_Dem_00769]	Definition of EcucReferenceDef DemStorageConditionGroupRef
[ECUC_Dem_00771]	Definition of EcucReferenceDef DemDataElementClassRef
[ECUC_Dem_00772]	Definition of EcucChoiceReferenceDef DemOBDInputEngineTemperature
[ECUC_Dem_00773]	Definition of EcucParamConfContainerDef DemStorageConditionGroup
[ECUC_Dem_00774]	Definition of EcucReferenceDef DemExtendedDataRecordClassRef
[ECUC_Dem_00775]	Definition of EcucParamConfContainerDef DemFreezeFrameRecNumClass
[ECUC_Dem_00776]	Definition of EcucReferenceDef DemFreezeFrameRecNumClassRef
[ECUC_Dem_00777]	Definition of EcucIntegerParamDef DemFreezeFrameRecordNumber
[ECUC_Dem_00779]	Definition of EcucEnumerationParamDef DemDTCSignificance
[ECUC_Dem_00784]	Definition of EcucEnumerationParamDef DemStatusBitHandlingTestFailed SinceLastClear
[ECUC_Dem_00786]	Definition of EcucEnumerationParamDef DemDebounceBehavior
[ECUC_Dem_00787]	Definition of EcucIntegerParamDef DemRatioId
[ECUC_Dem_00788]	Definition of EcucEnumerationParamDef DemClearEventAllowedBehavior
[ECUC_Dem_00790]	Definition of EcucEnumerationParamDef DemClearDTCLimitation
[ECUC_Dem_00791]	Definition of EcucBooleanParamDef DemDebounceCounterStorage
[ECUC_Dem_00792]	Definition of EcucBooleanParamDef DemEventAvailable
[ECUC_Dem_00793]	Definition of EcucEnumerationParamDef DemSuppressionSupport
[ECUC_Dem_00794]	Definition of EcucBooleanParamDef DemOBDCentralizedPID21Handling
[ECUC_Dem_00795]	Definition of EcucIntegerParamDef DemOBDCompliancy
[ECUC_Dem_00796]	Definition of EcucBooleanParamDef DemOBDEventDisplacement
[ECUC_Dem_00799]	Definition of EcucBooleanParamDef DemResetConfirmedBitOnOverflow
[ECUC_Dem_00800]	Definition of EcucReferenceDef DemFreezeFrameRecordClassRef
[ECUC_Dem_00801]	Definition of EcucParamConfContainerDef DemFreezeFrameRecordClass
[ECUC_Dem_00802]	Definition of EcucEnumerationParamDef DemFreezeFrameRecordUpdate
[ECUC_Dem_00803]	Definition of EcucEnumerationParamDef DemFreezeFrameRecordTrigger
[ECUC_Dem_00804]	Definition of EcucEnumerationParamDef DemExtendedDataRecordTrigger
[ECUC_Dem_00806]	Definition of EcucParamConfContainerDef DemDtr
[ECUC_Dem_00807]	Definition of EcucIntegerParamDef DemDtrId
[ECUC_Dem_00808]	Definition of EcucReferenceDef DemDtrEventRef
[ECUC_Dem_00809]	Definition of EcucIntegerParamDef DemDtrMid
[ECUC_Dem_00810]	Definition of EcucIntegerParamDef DemDtrTid
[ECUC_Dem_00811]	Definition of EcucIntegerParamDef DemDtrUasid
[ECUC_Dem_00812]	Definition of EcucEnumerationParamDef DemDtrUpdateKind

▽

△

Number	Heading
[ECUC_Dem_00813]	Definition of EcucFloatParamDef DemDtrCompuNumerator0
[ECUC_Dem_00814]	Definition of EcucFloatParamDef DemDtrCompuNumerator1
[ECUC_Dem_00815]	Definition of EcucFloatParamDef DemDtrCompuDenominator0
[ECUC_Dem_00820]	Definition of EcucReferenceDef DemRedStopLampIndicatorRef
[ECUC_Dem_00821]	Definition of EcucReferenceDef DemAmberWarningLampIndicatorRef
[ECUC_Dem_00822]	Definition of EcucReferenceDef DemProtectLampIndicatorRef
[ECUC_Dem_00823]	Definition of EcucParamConfContainerDef DemCallbackJ1939DTCStatus Changed
[ECUC_Dem_00824]	Definition of EcucFunctionNameDef DemCallbackDTCStatusChangedFnc
[ECUC_Dem_00825]	Definition of EcucParamConfContainerDef DemCallbackOBDDTCStatus Changed
[ECUC_Dem_00826]	Definition of EcucFunctionNameDef DemCallbackDTCStatusChangedFnc
[ECUC_Dem_00827]	Definition of EcucChoiceReferenceDef DemOBDTimeSinceEngineStart
[ECUC_Dem_00828]	Definition of EcucParamConfContainerDef DemJ1939FreezeFrameClass
[ECUC_Dem_00829]	Definition of EcucReferenceDef DemSPNClassRef
[ECUC_Dem_00830]	Definition of EcucParamConfContainerDef DemSPNClass
[ECUC_Dem_00831]	Definition of EcucIntegerParamDef DemSPNId
[ECUC_Dem_00832]	Definition of EcucReferenceDef DemSPNDataElementClassRef
[ECUC_Dem_00834]	Definition of EcucReferenceDef DemJ1939ExpandedFreezeFrameClassRef
[ECUC_Dem_00835]	Definition of EcucReferenceDef DemJ1939FreezeFrameClassRef
[ECUC_Dem_00838]	Definition of EcucEnumerationParamDef DemIUMPRDenGroup
[ECUC_Dem_00839]	Definition of EcucReferenceDef DemOBDGroupingAssociativeEventsRef
[ECUC_Dem_00841]	Definition of EcucEnumerationParamDef DemDataElementEndianness
[ECUC_Dem_00842]	Definition of EcucChoiceContainerDef DemSRDataElementClass
[ECUC_Dem_00843]	Definition of EcucChoiceContainerDef DemDiagnosisScaling
[ECUC_Dem_00844]	Definition of EcucParamConfContainerDef DemAlternativeDataInterface
[ECUC_Dem_00845]	Definition of EcucForeignReferenceDef DemDataElement
[ECUC_Dem_00846]	Definition of EcucForeignReferenceDef DemPortInterfaceMapping
[ECUC_Dem_00847]	Definition of EcucParamConfContainerDef DemAlternativeDataType
[ECUC_Dem_00848]	Definition of EcucForeignReferenceDef DemApplicationDataType
[ECUC_Dem_00849]	Definition of EcucParamConfContainerDef DemTextTableMapping
[ECUC_Dem_00850]	Definition of EcucIntegerParamDef DemInternalDataValue
[ECUC_Dem_00851]	Definition of EcucIntegerParamDef DemDiagnosisRepresentationDataValue
[ECUC_Dem_00858]	Definition of EcucEnumerationParamDef DemDataElementDefault Endianness
[ECUC_Dem_00860]	Definition of EcucInstanceReferenceDef DemSubElementInDataElement InstanceRef
[ECUC_Dem_00861]	Definition of EcucInstanceReferenceDef DemDataElementInstanceRef
[ECUC_Dem_00864]	Definition of EcucParamConfContainerDef DemGeneralJ1939

▽

△

Number	Heading
[ECUC_Dem_00865]	Definition of EcucBooleanParamDef DemJ1939ReadingDtcSupport
[ECUC_Dem_00866]	Definition of EcucBooleanParamDef DemJ1939FreezeFrameSupport
[ECUC_Dem_00867]	Definition of EcucBooleanParamDef DemJ1939RatioSupport
[ECUC_Dem_00868]	Definition of EcucBooleanParamDef DemJ1939Dm31Support
[ECUC_Dem_00869]	Definition of EcucBooleanParamDef DemJ1939Readiness1Support
[ECUC_Dem_00870]	Definition of EcucBooleanParamDef DemJ1939Readiness2Support
[ECUC_Dem_00871]	Definition of EcucBooleanParamDef DemJ1939Readiness3Support
[ECUC_Dem_00872]	Definition of EcucBooleanParamDef DemJ1939ClearDtcSupport
[ECUC_Dem_00873]	Definition of EcucBooleanParamDef DemJ1939ExpandedFreezeFrame Support
[ECUC_Dem_00874]	Definition of EcucParamConfContainerDef DemSubElementInDataElement Instance
[ECUC_Dem_00875]	Definition of EcucParamConfContainerDef DemDataElementInstance
[ECUC_Dem_00877]	Definition of EcucBooleanParamDef DemAgingRequiresTestedCycle
[ECUC_Dem_00878]	Definition of EcucEnumerationParamDef DemAvailabilitySupport
[ECUC_Dem_00879]	Definition of EcucBooleanParamDef DemOBDCentralizedPID31Handling
[ECUC_Dem_00880]	Definition of EcucBooleanParamDef DemGeneralInterfaceSupport
[ECUC_Dem_00881]	Definition of EcucParamConfContainerDef DemDebounceCounterBased Class
[ECUC_Dem_00882]	Definition of EcucParamConfContainerDef DemDebounceTimeBasedClass
[ECUC_Dem_00883]	Definition of EcucReferenceDef DemDebounceCounterBasedClassRef
[ECUC_Dem_00884]	Definition of EcucParamConfContainerDef DemObdDTC
[ECUC_Dem_00885]	Definition of EcucIntegerParamDef DemDtcValue
[ECUC_Dem_00886]	Definition of EcucParamConfContainerDef DemDTC
[ECUC_Dem_00887]	Definition of EcucIntegerParamDef DemDtcValue
[ECUC_Dem_00888]	Definition of EcucReferenceDef DemDTCRef
[ECUC_Dem_00889]	Definition of EcucReferenceDef DemObdDTCRef
[ECUC_Dem_00890]	Definition of EcucChoiceReferenceDef DemMemoryDestinationRef
[ECUC_Dem_00891]	Definition of EcucReferenceDef DemDebounceTimeBasedClassRef
[ECUC_Dem_00892]	Definition of EcucIntegerParamDef DemJ1939DTCValue
[ECUC_Dem_00893]	Definition of EcucReferenceDef DemStorageConditionReplacementEvent Ref
[ECUC_Dem_00894]	Definition of EcucEnumerationParamDef DemReportBehavior
[ECUC_Dem_00896]	Definition of EcucParamConfContainerDef DemPidDataElement
[ECUC_Dem_00898]	Definition of EcucIntegerParamDef DemOperationCycleId
[ECUC_Dem_00900]	Definition of EcucEnumerationParamDef DemOBDEngineType
[ECUC_Dem_00901]	Definition of EcucParamConfContainerDef DemPrimaryMemory
[ECUC_Dem_00903]	Definition of EcucIntegerParamDef DemUserDefinedMemoryIdentifier
[ECUC_Dem_00904]	Definition of EcucParamConfContainerDef DemComponent

▽

△

Number	Heading
[ECUC_Dem_00905]	Definition of EcucBooleanParamDef DemComponentIgnoresPriority
[ECUC_Dem_00906]	Definition of EcucFunctionNameDef DemComponentFailedCallbackFnc
[ECUC_Dem_00907]	Definition of EcucReferenceDef DemImmediateChildComponentRef
[ECUC_Dem_00908]	Definition of EcucReferenceDef DemComponentClassRef
[ECUC_Dem_00909]	Definition of EcucIntegerParamDef DemComponentPriority
[ECUC_Dem_00910]	Definition of EcucParamConfContainerDef DemUserDefinedMemory
[ECUC_Dem_00911]	Definition of EcucReferenceDef DemUDSOBDFreezeFrameClassRef
[ECUC_Dem_00912]	Definition of EcucEnumerationParamDef DemWWHOBDDTCClass
[ECUC_Dem_00913]	Definition of EcucParamConfContainerDef DemDtrs
[ECUC_Dem_00914]	Definition of EcucIntegerParamDef DemCounterBasedFdcThresholdStorage Value
[ECUC_Dem_00915]	Definition of EcucFloatParamDef DemTimeBasedFdcThresholdStorage Value
[ECUC_Dem_00916]	Definition of EcucBooleanParamDef DemEventRecoverableInSame OperationCycle
[ECUC_Dem_00917]	Definition of EcucBooleanParamDef DemOBDDelayedDCYConfirmedAnd MIL
[ECUC_Dem_00918]	Definition of EcucBooleanParamDef DemAgingRequiresNotFailedCycle
[ECUC_Dem_00919]	Definition of EcucReferenceDef DemLeadingCycleRef
[ECUC_Dem_00920]	Definition of EcucReferenceDef DemOBDPFCCycleRef
[ECUC_Dem_00921]	Definition of EcucFloatParamDef DemCausalityDelayTime
[ECUC_Dem_00922]	Definition of EcucReferenceDef DemOBDWarmUpCycleRef
[ECUC_Dem_00923]	Definition of EcucReferenceDef DemOBDDrivingCycleRef
[ECUC_Dem_00924]	Definition of EcucIntegerParamDef DemEventConfirmationThreshold
[ECUC_Dem_00925]	Definition of EcucParamConfContainerDef DemClearDTCNotification
[ECUC_Dem_00926]	Definition of EcucFunctionNameDef DemClearDtcNotificationFnc
[ECUC_Dem_00927]	Definition of EcucEnumerationParamDef DemClearDtcNotificationTime
[ECUC_Dem_00928]	Definition of EcucModuleDef Dem
[ECUC_Dem_00929]	Definition of EcucBooleanParamDef DemEventConfirmationThreshold CounterAdaptable
[ECUC_Dem_00930]	Definition of EcucIntegerParamDef DemComponentId
[ECUC_Dem_00931]	Definition of EcucParamConfContainerDef DemClient
[ECUC_Dem_00932]	Definition of EcucIntegerParamDef DemClientId
[ECUC_Dem_00933]	Definition of EcucBooleanParamDef DemClientUsesRte
[ECUC_Dem_00934]	Definition of EcucParamConfContainerDef DemAlternativeDiagnosticData Element
[ECUC_Dem_00935]	Definition of EcucForeignReferenceDef DemDiagnosticDataElementRef
[ECUC_Dem_00936]	Definition of EcucParamConfContainerDef DemCallbackMonitorStatus Changed
[ECUC_Dem_00937]	Definition of EcucFunctionNameDef DemCallbackMonitorStatusChanged Fnc

▽

△

Number	Heading
[ECUC_Dem_00938]	Definition of EcucFunctionNameDef DemGeneralCallbackMonitorStatus ChangedFnc
[ECUC_Dem_00939]	Definition of EcucParamConfContainerDef DemEventMemorySet
[ECUC_Dem_00940]	Definition of EcucReferenceDef DemOBDEventMemorySetRef
[ECUC_Dem_00941]	Definition of EcucReferenceDef DemClearEventsWithoutDTCEventMemory Ref
[ECUC_Dem_00942]	Definition of EcucReferenceDef DemEventMemorySetRef
[ECUC_Dem_00943]	Definition of EcucEnumerationParamDef DemClientFunctionality
[ECUC_Dem_00944]	Definition of EcucParamConfContainerDef DemMultiEventTriggering
[ECUC_Dem_00945]	Definition of EcucReferenceDef DemMultiEventTriggeringMasterEventRef
[ECUC_Dem_00946]	Definition of EcucReferenceDef DemMultiEventTriggeringSlaveEventRef
[ECUC_Dem_00947]	Definition of EcucFunctionNameDef DemComponentFailedCallbackUsePort
[ECUC_Dem_00948]	Definition of EcucBooleanParamDef DemFFPrestorageInNvm
[ECUC_Dem_00949]	Definition of EcucIntegerParamDef DemDataElementArraySize
[ECUC_Dem_00950]	Definition of EcucEnumerationParamDef DemDataElementDataType
[ECUC_Dem_00951]	Definition of EcucBooleanParamDef DemDataElementProvideMonitorData
[ECUC_Dem_00967]	Definition of EcucIntegerParamDef DemDataElementArraySize
[ECUC_Dem_00968]	Definition of EcucBooleanParamDef DemEventAvailableInVariant
[ECUC_Dem_00969]	Definition of EcucIntegerParamDef DemEventAssociatedIdentification
[ECUC_Dem_00970]	Definition of EcucEnumerationParamDef DemEventReportingType
[ECUC_Dem_00978]	Definition of EcucIntegerParamDef DemDtcStatusAvailabilityMask
[ECUC_Dem_00980]	Definition of EcucEnumerationParamDef DemEventDisplacementStrategy
[ECUC_Dem_00982]	Definition of EcucEnumerationParamDef DemEventMemoryEntryStorage Trigger
[ECUC_Dem_00984]	Definition of EcucEnumerationParamDef DemOccurrenceCounter Processing
[ECUC_Dem_00986]	Definition of EcucEnumerationParamDef DemTypeOfFreezeFrameRecord Numeration
[ECUC_Dem_00988]	Definition of EcucIntegerParamDef DemDtcValue3Byte
[ECUC_Dem_00989]	Definition of EcucBooleanParamDef DemSupportedObdUdsDtcSeparation
[ECUC_Dem_00990]	Definition of EcucEnumerationParamDef DemDataElementFncSignature
[ECUC_Dem_00991]	Definition of EcucBooleanParamDef DemResetPendingBitOnOverflow
[ECUC_Dem_00993]	Definition of EcucBooleanParamDef DemAffectedByControlDTCSetting
[ECUC_Dem_00994]	Definition of EcucBooleanParamDef DemDtcOBDRelevance
[ECUC_Dem_00995]	Definition of EcucBooleanParamDef DemFreezeMANWhenGeneral DenominatorInhibited
[SWS_Dem_00024]	
[SWS_Dem_00029]	
[SWS_Dem_00124]	
[SWS_Dem_00176]	Definition of imported datatypes of module Dem

▽

△

Number	Heading
[SWS_Dem_00259]	Definition of configurable interface <DemGeneralTriggerOnEventUdsStatus>
[SWS_Dem_00293]	
[SWS_Dem_00347]	PID $1E support based on PTO support
[SWS_Dem_00368]	
[SWS_Dem_00377]	PID $1E value according to reported PTO state
[SWS_Dem_00378]	DTC processing for PTO-affected DTCs
[SWS_Dem_00386]	TestFailed Bit behavior
[SWS_Dem_00536]	Support event combination
[SWS_Dem_00608]	Definition of ClientServerInterface DTCSuppression
[SWS_Dem_00672]	
[SWS_Dem_00788]	
[SWS_Dem_00792]	
[SWS_Dem_00855]	
[SWS_Dem_00882]	
[SWS_Dem_00883]	
[SWS_Dem_00884]	
[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_00976]	Definition of API function Dem_J1939DcmClearDTC
[SWS_Dem_00981]	Definition of API function Dem_J1939DcmGetNextFilteredRatio
[SWS_Dem_01007]	Definition of Port CBStatusDTC_{Name} required by module Dem
[SWS_Dem_01010]	Definition of Port ControlDTCSuppression provided by module Dem
[SWS_Dem_01047]	Definition of API function Dem_SetDTCSuppression
[SWS_Dem_01101]	
[SWS_Dem_01281]	
[SWS_Dem_01412]	Setting a filter for J1939 DTCs
[SWS_Dem_91025]	Definition of API function Dem_GetDTCSuppression
[SWS_Dem_91060]	Definition of configurable interface <Module>_DemRead<DataElement>
[SWS_Dem_91201]	Definition of Std_ReturnType-extension for module Dem

Table B.2: Changed Specification Items in R26-11

B.1.3 Deleted Specification Items in R26-11

Number	Heading
[ECUC_Dem_00714]	Definition of EcucBooleanParamDef DemStatusBitStorageTestFailed
[ECUC_Dem_00721]	Definition of EcucBooleanParamDef DemVersionInfoApi
[ECUC_Dem_00789]	Definition of EcucEnumerationParamDef DemDebounceBehavior
[ECUC_Dem_00840]	Definition of EcucEnumerationParamDef DemDataElementDataType
[ECUC_Dem_00862]	Definition of EcucInstanceReferenceDef DemSubElementInImplData ElementInstanceRef
[ECUC_Dem_00876]	Definition of EcucParamConfContainerDef DemSubElementInImplData ElementInstance
[SWS_Dem_00177]	Definition of API function Dem_GetVersionInfo
[SWS_Dem_00387]	
[SWS_Dem_00388]	
[SWS_Dem_00525]	
[SWS_Dem_00539]	
[SWS_Dem_00540]	
[SWS_Dem_00541]	
[SWS_Dem_00542]	
[SWS_Dem_00789]	
[SWS_Dem_00793]	
[SWS_Dem_00828]	

Table B.3: Deleted Specification Items in R26-11

B.1.4 Added Constraints in R26-11

Number	Heading
[SWS_Dem_CONSTR_06182]	Restriction on 3-byte OBD DTC internal data element
[SWS_Dem_CONSTR_06183]	Restriction of `DemObdDTCRef` for non-OBD-relevant DTCs
[SWS_Dem_CONSTR_06184]	Restriction of `DemDtrEventRef` for non-OBD-relevant DTCs
[SWS_Dem_CONSTR_06185]	Restriction of `DemOBDGroupingAssociativeEventsRef` for non-OBD-relevant DTCs

▽

△

Number	Heading
[SWS_Dem_CONSTR_06186]	Matching event availability parameters in post-build configuration

Table B.4: Added Constraints in R26-11

B.1.5 Changed Constraints in R26-11

Number	Heading
[SWS_Dem_CONSTR_06116]	Limited use of monitor status change callbacks to events reported from SW-Cs only

Table B.5: Changed Constraints in R26-11

B.1.6 Deleted Constraints in R26-11

Number	Heading
[SWS_Dem_CONSTR_06113]	Configuration of the test failed status bit storage

Table B.6: Deleted Constraints in R26-11

B.2 Traceable item history of this document according to AUTOSAR Release R25-11

B.2.1 Added Specification Items in R25-11

Number	Heading
[ECUC_Dem_00995]	Definition of EcucBooleanParamDef DemFreezeMANWhenGeneralDenominatorInhibited
[SWS_Dem_01297]	Freeze monitor activity numerator if the general denominator is inhibited
[SWS_Dem_01412]	Setting a filter for J1939 DTCs
[SWS_Dem_01413]	Selecting all J1939 DTCs
[SWS_Dem_01414]	Selecting all J1939 emission related DTCs
[SWS_Dem_01415]	Selecting all J1939 emission related Class_A DTCs

▽

△

Number	Heading
[SWS_Dem_01416]	Selecting all J1939 emission related Class_B1 DTCs
[SWS_Dem_01417]	Selecting all J1939 emission related Class_B2 DTCs
[SWS_Dem_01418]	Selecting all J1939 emission related Class_C DTCs
[SWS_Dem_01419]	Monitor status availability
[SWS_Dem_01420]	Acceptance of Requests During the Valid Initialization Sequence
[SWS_Dem_01421]	Handling of Requests Outside the Valid Initialization Sequence
[SWS_Dem_01422]	Initialization and Storage Actions for Accepted Non-Volatile Memory Requests
[SWS_Dem_01423]	Successful Completion of Non-Volatile Memory Initialization and Storage
[SWS_Dem_01424]	Indication of Ongoing Non-Volatile Memory Storage
[SWS_Dem_01425]	Error Indication for Unsuccessful or Invalid Non-Volatile Memory Operations
[SWS_Dem_91199]	Definition of API function Dem_InitAllNvmBlocks
[SWS_Dem_91200]	Definition of API function Dem_InitAllNvmBlocksGetResult
[SWS_Dem_91201]	Definition of Std_ReturnType-extension for module Dem

Table B.7: Added Specification Items in R25-11

B.2.2 Changed Specification Items in R25-11

Number	Heading
[ECUC_Dem_00602]	Definition of EcucBooleanParamDef DemConsiderPtoStatus
[ECUC_Dem_00623]	Definition of EcucIntegerParamDef DemAgingCycleCounterThreshold
[ECUC_Dem_00624]	Definition of EcucReferenceDef DemAgingCycleRef
[ECUC_Dem_00626]	Definition of EcucParamConfContainerDef DemCallbackDTCStatusChanged
[ECUC_Dem_00634]	Definition of EcucParamConfContainerDef DemConfigSet
[ECUC_Dem_00641]	Definition of EcucParamConfContainerDef DemDTCAttributes
[ECUC_Dem_00643]	Definition of EcucIntegerParamDef DemDTCFunctionalUnit
[ECUC_Dem_00662]	Definition of EcucIntegerParamDef DemDTCPriority
[ECUC_Dem_00667]	Definition of EcucReferenceDef DemExtendedDataClassRef
[ECUC_Dem_00673]	Definition of EcucParamConfContainerDef DemFreezeFrameClass
[ECUC_Dem_00677]	Definition of EcucParamConfContainerDef DemGeneral
[ECUC_Dem_00702]	Definition of EcucReferenceDef DemOperationCycleRef
[ECUC_Dem_00707]	Definition of EcucReferenceDef DemDidClassRef
[ECUC_Dem_00746]	Definition of EcucReferenceDef DemEnableConditionGroupRef
[ECUC_Dem_00748]	Definition of EcucIntegerParamDef DemIndicatorHealingCycleCounter Threshold
[ECUC_Dem_00755]	Definition of EcucEnumerationParamDef DemEventOBDReadinessGroup
[ECUC_Dem_00756]	Definition of EcucParamConfContainerDef DemGeneralOBD

▽

△

Number	Heading
[ECUC_Dem_00823]	Definition of EcucParamConfContainerDef DemCallbackJ1939DTCStatus Changed
[ECUC_Dem_00825]	Definition of EcucParamConfContainerDef DemCallbackOBDDTCStatus Changed
[ECUC_Dem_00840]	Definition of EcucEnumerationParamDef DemDataElementDataType
[ECUC_Dem_00864]	Definition of EcucParamConfContainerDef DemGeneralJ1939
[ECUC_Dem_00912]	Definition of EcucEnumerationParamDef DemWWHOBDDTCClass
[ECUC_Dem_00924]	Definition of EcucIntegerParamDef DemEventConfirmationThreshold
[ECUC_Dem_00931]	Definition of EcucParamConfContainerDef DemClient
[ECUC_Dem_00950]	Definition of EcucEnumerationParamDef DemDataElementDataType
[SWS_Dem_00124]	
[SWS_Dem_00171]	
[SWS_Dem_00172]	
[SWS_Dem_00284]	
[SWS_Dem_00493]	
[SWS_Dem_00503]	
[SWS_Dem_00907]	
[SWS_Dem_00927]	Definition of ImplementationDataType Dem_DebounceResetStatusType
[SWS_Dem_00932]	Definition of datatype Dem_DTCKindType
[SWS_Dem_00979]	Definition of API function Dem_J1939DcmGetNextSPNInFreezeFrame
[SWS_Dem_00987]	
[SWS_Dem_01007]	Definition of Port CBStatusDTC_{Name} required by module Dem
[SWS_Dem_01046]	
[SWS_Dem_01054]	
[SWS_Dem_01113]	
[SWS_Dem_01326]	Monitor status initialization
[SWS_Dem_91026]	Definition of ImplementationDataType Dem_DataType_{Data}
[SWS_Dem_91033]	Definition of ImplementationDataType Dem_DataPrimitiveType_{Data}
[SWS_Dem_91034]	Definition of ImplementationDataType Dem_DataArrayType_{Data}

Table B.8: Changed Specification Items in R25-11

B.2.3 Deleted Specification Items in R25-11

Number	Heading
[ECUC_Dem_00897]	Definition of EcucIntegerParamDef DemAgingCycleCounterThresholdFor TFSLC

Table B.9: Deleted Specification Items in R25-11

B.2.4 Added Constraints in R25-11

Number	Heading
[SWS_Dem_CONSTR_06180]	Restriction on 64 bit array parameter size
[SWS_Dem_CONSTR_06181]	Restriction on 64 bit array parameter size

Table B.10: Added Constraints in R25-11

B.2.5 Changed Constraints in R25-11

Number	Heading
[SWS_Dem_CONSTR_06147]	Limit readiness to OBD ECUs only
[SWS_Dem_CONSTR_06158]	
[SWS_Dem_CONSTR_06161]	

Table B.11: Changed Constraints in R25-11

B.2.6 Deleted Constraints in R25-11

Number	Heading
[SWS_Dem_CONSTR_06126]	Dependency for DemAgingCycleCounterThresholdForTFSLC

Table B.12: Deleted Constraints in R25-11

B.3 Traceable item history of this document according to AUTOSAR Release R24-11

B.3.1 Added Specification Items in R24-11

Number	Heading
[ECUC_Dem_00993]	Definition of EcucBooleanParamDef DemAffectedByControlDTCSetting
[ECUC_Dem_00994]	Definition of EcucBooleanParamDef DemDtcOBDRelevance
[SWS_Dem_01402]	
[SWS_Dem_01403]	
[SWS_Dem_01404]	Latching of Internal Data Element DEM_CYCLES_TESTED_SINCE_LAST_FAILED
[SWS_Dem_01405]	Latching of Internal Data Element DEM_CYCLES_TESTED_SINCE_FIRST_FAILED
[SWS_Dem_01406]	Availability of internal data element Internal Data Element DEM_CYCLES_TESTED_SINCE_LAST_FAILED
[SWS_Dem_01407]	Availability of internal data element Internal Data Element DEM_CYCLES_TESTED_SINCE_FIRST_FAILED
[SWS_Dem_01408]	Behavior on invalid use of pre qualified results
[SWS_Dem_01409]	UDS relevant OBD DTCs
[SWS_Dem_01410]	
[SWS_Dem_01411]	

Table B.13: Added Specification Items in R24-11

B.3.2 Changed Specification Items in R24-11

Number	Heading
[ECUC_Dem_00616]	Definition of EcucEnumerationParamDef DemInternalDataElement
[ECUC_Dem_00618]	Definition of EcucIntegerParamDef DemDebounceCounterFailedThreshold
[ECUC_Dem_00635]	Definition of EcucIntegerParamDef DemDebounceCounterDecrementStepSize
[ECUC_Dem_00636]	Definition of EcucIntegerParamDef DemDebounceCounterPassedThreshold
[ECUC_Dem_00637]	Definition of EcucIntegerParamDef DemDebounceCounterIncrementStepSize
[ECUC_Dem_00638]	Definition of EcucIntegerParamDef DemDebounceCounterJumpDownValue
[ECUC_Dem_00639]	Definition of EcucIntegerParamDef DemDebounceCounterJumpUpValue
[ECUC_Dem_00645]	Definition of EcucEnumerationParamDef DemDTCSeverity
[ECUC_Dem_00685]	Definition of EcucBooleanParamDef DemDebounceCounterJumpDown
[ECUC_Dem_00686]	Definition of EcucBooleanParamDef DemDebounceCounterJumpUp
[ECUC_Dem_00716]	Definition of EcucFloatParamDef DemDebounceTimeFailedThreshold

▽

△

Number	Heading
[ECUC_Dem_00717]	Definition of EcucFloatParamDef DemDebounceTimePassedThreshold
[ECUC_Dem_00883]	Definition of EcucReferenceDef DemDebounceCounterBasedClassRef
[ECUC_Dem_00886]	Definition of EcucParamConfContainerDef DemDTC
[ECUC_Dem_00891]	Definition of EcucReferenceDef DemDebounceTimeBasedClassRef
[ECUC_Dem_00910]	Definition of EcucParamConfContainerDef DemUserDefinedMemory
[ECUC_Dem_00914]	Definition of EcucIntegerParamDef DemCounterBasedFdcThresholdStorage Value
[ECUC_Dem_00915]	Definition of EcucFloatParamDef DemTimeBasedFdcThresholdStorage Value
[SWS_Dem_00079]	
[SWS_Dem_00212]	Definition of API function Dem_GetStatusOfDTC
[SWS_Dem_00215]	Definition of API function Dem_GetNextFilteredDTC
[SWS_Dem_00227]	Definition of API function Dem_GetNextFilteredDTCAndFDC
[SWS_Dem_00281]	Definition of API function Dem_GetNextFilteredDTCAndSeverity
[SWS_Dem_00666]	Definition of ClientServerInterface ClearDTC
[SWS_Dem_00973]	Definition of API function Dem_J1939DcmGetNextFilteredDTC
[SWS_Dem_00975]	Definition of API function Dem_J1939DcmGetNextDTCwithLampStatus
[SWS_Dem_00978]	Definition of API function Dem_J1939DcmGetNextFreezeFrame
[SWS_Dem_00981]	Definition of API function Dem_J1939DcmGetNextFilteredRatio
[SWS_Dem_01364]	Functionality of Dem_SetDTCFilterByReadinessGroup

Table B.14: Changed Specification Items in R24-11

B.3.3 Deleted Specification Items in R24-11

Number	Heading
[SWS_Dem_00437]	

Table B.15: Deleted Specification Items in R24-11

B.3.4 Added Constraints in R24-11

none

B.3.5 Changed Constraints in R24-11

none

B.3.6 Deleted Constraints in R24-11

none

B.4 Traceable item history of this document according to AUTOSAR Release R23-11

B.4.1 Added Specification Items in R23-11

Number	Heading
[SWS_Dem_01394]	
[SWS_Dem_01395]	Configurable reset of the pendingDTC bit in case of displacement
[SWS_Dem_01396]	
[SWS_Dem_01397]	
[SWS_Dem_01398]	Support of DTCExtendedDataRecordNumber 0xFE
[SWS_Dem_01399]	Support of DTCExtendedDataRecordNumber 0xFF in primary fault memory
[SWS_Dem_01400]	Support of DTCExtendedDataRecordNumber 0xFF in user defined fault memory
[SWS_Dem_01401]	

Table B.16: Added Specification Items in R23-11

B.4.2 Changed Specification Items in R23-11

Number	Heading
[SWS_Dem_00124]	
[SWS_Dem_00362]	Release of IUMPR Denominator
[SWS_Dem_00409]	Configurable reset of the confirmedDTC bit in case of displacement
[SWS_Dem_00626]	
[SWS_Dem_00913]	Reading IUMPR data for J1939
[SWS_Dem_01250]	
[SWS_Dem_01309]	
[SWS_Dem_01339]	
[SWS_Dem_01391]	Configuration of DTC priority as extended data record
[SWS_Dem_01392]	
[SWS_Dem_91017]	Definition of API function Dem_SelectExtendedDataRecord
[SWS_Dem_91037]	Definition of API function Dem_SetEventStatusWithMonitorData
[SWS_Dem_91192]	Definition of API function Dem_SetDTCFilterByExtendedDataRecord Number

▽

△

Number	Heading
[SWS_Dem_91193]	Definition of API function Dem_SetDTCFilterByReadinessGroup
[SWS_Dem_91194]	Definition of datatype Dem_EventOBDReadinessGroupType

Table B.17: Changed Specification Items in R23-11

B.4.3 Deleted Specification Items in R23-11

Number	Heading
[SWS_Dem_01352]	Setting event failed with synchronous freeze frame processing
[SWS_Dem_01353]	Synchronous snapshot record processing
[SWS_Dem_01354]	Only C-Callout data capturing for synchronous snapshot record processing
[SWS_Dem_01355]	Optional callback after synchronous snapshot record processing
[SWS_Dem_01356]	CallbackEventSyncStorageProcessed only for synchronous reporting event
[SWS_Dem_91041]	
[SWS_Dem_91042]	

Table B.18: Deleted Specification Items in R23-11

B.4.4 Added Constraints in R23-11

none

B.4.5 Changed Constraints in R23-11

none

B.4.6 Deleted Constraints in R23-11

none