

<b>Document Title</b>	Specification of Diagnostic Event Manager
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	019
<b>Document Classification</b>	Standard

<b>Document Status</b>	Final
<b>Part of AUTOSAR Release</b>	4.2.2

Document Change History		
Release	Changed by	Description
4.2.2	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>New APIs <a href="#">Dem_GetEventFreezeFrameDataEx</a> and <a href="#">Dem_GetEventExtendedDataRecordEx</a> with buffersize as parameter and corrected return value definitions.</li> <li>Providing OBD FreezFrame for UDS service 0x19 0x05</li> <li>ISO 14229-1:2013[1] NRC handling for service 0x14</li> <li>Refined service interfaces for DataElements</li> <li>minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation</li> </ul>
4.2.1	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>Support of ISO 27145 (WWH-OBD / Euro VI)[2]</li> <li>Update to support ISO 14229-1:2013[3]</li> <li>Introduction of event dependencies</li> <li>Refined DTC/Event suppression</li> </ul>
4.1.3	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>Further clarification of event combination</li> <li>Clarification of DTC groups</li> <li>Editorial changes</li> </ul>
4.1.2	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>Added API table for service interfaces</li> <li>Clarification of event combination</li> <li>Editorial changes</li> <li>Removed chapter(s) on change documentation</li> </ul>

4.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Incorporated new and reporting of several Dem internal data elements</li> <li>• Supported J1939</li> <li>• Supported OBD</li> <li>• Reworked Dem/Dcm interface</li> <li>• Introduced new debouncing behaviour and debounce counter storage</li> <li>• Extended clear DTC functionality</li> <li>• Supported event suppression</li> <li>• Extended DTC storage behavior</li> <li>• Incorporation of user-controlled warning indicator requestet bit</li> <li>• Incorporation of autostart behaviour for operation cycles</li> </ul>
4.0.3	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Introduced multiple formats per DTC</li> <li>• Reworked Dem_ResetEventStatus behavior</li> <li>• Reworked Dlt interaction</li> <li>• Reworked Dem/Dcm interface</li> <li>• Corrected include-structure and RTE interfaces</li> <li>• Refined several aspects on features</li> </ul>
4.0.1	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Reworked Dem/Dcm interface</li> <li>• Extended definition of "Diagnostic Monitor"</li> <li>• Introduced "Event significance" and "DTC suppression"</li> <li>• Reworked OBD (esp. interface for service \$02, readiness, and permanent memory)</li> <li>• Reworked file-structure</li> <li>• Finalization of issues on Revision 1</li> </ul>
3.1.4	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Improved and extended of Dem functional description (especially: status bit handling, event displacement, debouncing, fault confirmation and indicator handling, event combination, enable- &amp; storage conditions, event related data, operation cycle management)</li> <li>• Introduced the approach for functional diagnostics of SW-Cs</li> <li>• Added Dlt interaction and debugging interface</li> <li>• Document structure reworked and extended</li> <li>• Legal disclaimer revised</li> </ul>

3.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Document structure reworked and extended</li> <li>• Add APIs and configuration parameters for OBD support</li> <li>• Improve interaction between DCM and software components</li> <li>• Legal disclaimer revised</li> </ul>
3.0.1	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Improvement of RTE compliance</li> <li>• Improvement of configuration part</li> <li>• Improvement of document structure</li> <li>• Rework and tightening of data type usage</li> <li>• New API added</li> <li>• Document meta information extended</li> <li>• Small layout adaptations made</li> </ul>
2.1.115	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• "Advice for users" revised</li> <li>• "Revision Information" added</li> </ul>
2.1	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Complete rework and extension of debouncing part</li> <li>• Dem_ClearGroupOfDTC and Dem_ClearSingleDTC replaced by Dem_SingleDTC</li> <li>• Dem_GetNextFilteredDTCAndFDC, Dem_SetDTCFilterForRecords, Dem_GetSizeOfFreezeFrame, Dem_SetValueByOemId, Dem_SetEnableCondition, Xxx_DemGetFaultDetectionCounter added</li> <li>• DTCTranslationType replaced by DTCKind in several APIs</li> <li>• Chapter "Service DEM" added</li> <li>• Function IDs reworked</li> <li>• File Structure reworked and extended</li> <li>• Configuration chapter reworked and extended</li> <li>• Dem_GetNextFilteredDTC reworked</li> <li>• Legal disclaimer revised</li> </ul>
2.0	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Layout Adaptations</li> </ul>

## Disclaimer

This specification 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 specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification 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 specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

## Advice for users

AUTOSAR specifications may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the specifications for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such specifications, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

## Table of Contents

1	Introduction and functional overview	17
2	Acronyms and Abbreviations	18
2.1	Acronyms	18
2.2	Abbreviations	20
3	Related documentation	22
3.1	Input documents & related standards and norms	22
3.2	Related specification	23
4	Constraints and assumptions	24
4.1	Limitations	24
4.2	Applicability to car domains	25
5	Dependencies to other modules	26
5.1	File structure	27
5.1.1	Code file structure	27
5.1.2	Header file structure	27
5.2	Integration Hints	29
6	Requirements traceability	30
7	Functional specification	44
7.1	Startup behavior	44
7.2	Monitor re-initialization	45
7.3	Diagnostic event definition	47
7.3.1	Event priority	50
7.3.2	Event occurrence	50
7.3.3	Event kind	50
7.3.4	Event destination	51
7.3.5	Diagnostic monitor definition	51
7.3.6	Event dependencies	52
7.3.7	Component availability	53
7.4	Diagnostic trouble code definition	53
7.4.1	DTC kind	55
7.4.2	DTC format	56
7.4.3	DTC groups	57
7.4.4	DTC severity	58
7.4.5	Functional unit	58
7.4.6	DTC significance	59
7.4.7	Suppress DTC output	59
7.4.8	Availability of events (visibility and computation)	61
7.5	Monitored Component Definition	64
7.5.1	Definition of components and dependencies	65
7.6	Event memory description	66

7.6.1	Event status management . . . . .	67
7.6.1.1	Status bit support . . . . .	68
7.6.1.2	Status bit update . . . . .	68
7.6.1.3	Status bit transitions . . . . .	70
7.6.1.4	Active/Passive status . . . . .	73
7.6.1.5	Notification of status bit changes . . . . .	74
7.6.2	Event memory management . . . . .	75
7.6.2.1	Event retention . . . . .	77
7.6.2.2	Clearing event memory entries . . . . .	79
7.6.2.3	Event memory overflow indication . . . . .	82
7.6.2.4	Event displacement . . . . .	83
7.6.2.5	Reporting order of event memory entries . . . . .	86
7.6.3	Debouncing of diagnostic events . . . . .	87
7.6.3.1	Counter based debounce algorithm . . . . .	88
7.6.3.2	Time based debounce algorithm . . . . .	92
7.6.3.3	Further specific debounce algorithms . . . . .	97
7.6.3.4	Monitor internal debounce algorithm . . . . .	97
7.6.3.5	Debounce algorithm initialization and reset conditions . . . . .	97
7.6.3.6	Fault detection counter retrieval . . . . .	98
7.6.3.7	Fault detection counter reporting . . . . .	99
7.6.4	Fault confirmation . . . . .	100
7.6.4.1	Method for grouping of association of events for OBD purpose . . . . .	100
7.6.5	Event Combination . . . . .	101
7.6.5.1	Combination On Storage . . . . .	103
7.6.5.2	Combination On Retrieval . . . . .	103
7.6.6	Enable and storage conditions of diagnostic events . . . . .	104
7.6.7	Event related data . . . . .	108
7.6.7.1	Storage of freeze frame data . . . . .	109
7.6.7.2	Pre-storage of freeze frame data . . . . .	112
7.6.7.3	Storage of extended data . . . . .	113
7.6.7.4	Configuration of Event related data . . . . .	115
7.6.7.5	Notification of data changes . . . . .	121
7.6.8	Operation cycle management . . . . .	122
7.6.9	Operation Cycle Counters . . . . .	126
7.6.9.1	Cycles since last failed . . . . .	126
7.6.9.2	Cycles since first failed . . . . .	126
7.6.9.3	Failed cycles . . . . .	127
7.6.10	Aging of diagnostic events . . . . .	127
7.6.11	Healing of diagnostic events . . . . .	130
7.6.11.1	Warning indicator handling . . . . .	131
7.6.11.2	User controlled WarningIndicatorRequested-bit . . . . .	133
7.6.11.3	Handling of the warning indicator lamp (MIL) . . . . .	134
7.6.11.4	Notification and Set of the warning indicator status . . . . .	135
7.7	BSW Error Handling . . . . .	135
7.8	OBD-specific functionality . . . . .	137

7.8.1	General overview and restrictions . . . . .	137
7.8.2	PIDs provided by Dem . . . . .	140
7.8.2.1	Centralized PID \$21 / \$31 / \$4D / \$4E handling . . . . .	141
7.8.3	Readiness status . . . . .	143
7.8.4	In-Use-Monitor Performance Ratio (IUMPR) Support . . . . .	144
7.8.5	OBD for light duty (OBD2) . . . . .	149
7.8.5.1	Service \$01 Read Current Powertrain Diagnostic Data	149
7.8.5.2	Service \$02 Read Powertrain Freeze Frame Data . . . . .	149
7.8.5.3	Service \$03 - Read Emission-Related Powertrain Diagnostic Trouble Codes . . . . .	151
7.8.5.4	Service \$04 - Clear Reset Emission-Related Diagnostic Information . . . . .	151
7.8.5.5	Service \$06 - Support of central DTR handling . . . . .	151
7.8.5.6	Service \$07 - Read emission-related diagnostic trouble codes detected during during current or last completed driving cycle . . . . .	155
7.8.5.7	Service \$0A - Read Emission-Related Diagnostic Trouble Codes with Permanent Status . . . . .	155
7.8.5.8	MIL Handling . . . . .	156
7.8.6	WWH-OBD . . . . .	156
7.8.6.1	DTC WWHOBD class . . . . .	156
7.8.6.2	Continuous-MI counter . . . . .	157
7.8.6.3	Cumulative Continuous-MI counter (Master ECU) . . . . .	157
7.8.6.4	Class B1 counter . . . . .	158
7.8.6.5	Activation Mode . . . . .	158
7.8.6.6	Freeze Frame 0x00 . . . . .	159
7.8.6.7	Extended Data Record 0x90 . . . . .	160
7.8.6.8	Aging . . . . .	160
7.8.6.9	Service \$19 42 - Read WWH-OBD DTC By Mask Record . . . . .	160
7.8.6.10	Service \$14 FFFF33 - Clear Emission Related DTCs . . . . .	161
7.9	J1939 specific functionality . . . . .	161
7.9.1	Read DTC . . . . .	161
7.9.1.1	Composite Malfunction Indicator Lamp Status . . . . .	162
7.9.1.2	Composite Red Stop Lamp Status . . . . .	163
7.9.1.3	Composite Amber Warning Lamp Status . . . . .	163
7.9.1.4	Composite Protect Lamp Status . . . . .	163
7.9.1.5	DTC data acquisition . . . . .	164
7.9.2	Clear DTCs . . . . .	165
7.9.3	DM31 . . . . .	165
7.9.3.1	Malfunction Indicator Lamp . . . . .	166
7.9.3.2	Red Stop Lamp . . . . .	166
7.9.3.3	Amber Warning Lamp . . . . .	167
7.9.3.4	Protect Lamp . . . . .	167
7.9.4	FreezeFrame . . . . .	168
7.9.4.1	SPNs in ExpandedFreezeFrame . . . . .	169

7.9.5	Diagnostic Readiness . . . . .	170
7.9.6	Monitor Performance Ratio . . . . .	171
7.10	Interaction with other Software Modules . . . . .	172
7.10.1	Interaction with Software Components (SW-C) . . . . .	172
7.10.2	Interaction with Diagnostic Communication Manager (Dcm) . . . . .	173
7.10.2.1	Access DTCs and Status Information . . . . .	174
7.10.2.2	Access event related data . . . . .	177
7.10.2.3	Clear diagnostic information . . . . .	180
7.10.2.4	Control DTC setting . . . . .	181
7.10.2.5	Asynchronous Dcm operations . . . . .	182
7.10.3	Interaction with J1939 Diagnostic Manager (J1939) . . . . .	182
7.10.4	Interaction with Function Inhibition Manager (FIM) . . . . .	182
7.10.5	Interaction with NVRAM Manager (NvM) . . . . .	183
7.10.6	Interaction with Default Error Tracer (Det) . . . . .	184
7.10.7	Interaction with Diagnostic Log & Trace (Dlt) . . . . .	184
7.10.8	Required data by the Dem module . . . . .	185
7.11	Version check . . . . .	186
7.12	Error classification . . . . .	186
7.12.1	Development Errors . . . . .	186
7.12.2	Runtime Errors . . . . .	188
7.12.3	Transient Faults . . . . .	188
7.12.4	Production Errors . . . . .	188
7.12.5	Extended Production Errors . . . . .	188
7.13	Error detection . . . . .	188
7.14	Error notification . . . . .	188
7.15	Support for Debugging . . . . .	188
8	API specification . . . . .	189
8.1	Imported Types . . . . .	192
8.2	Type definitions . . . . .	192
8.2.1	Dem data types . . . . .	192
8.2.1.1	Dem_ComponentIdType . . . . .	192
8.2.1.2	Dem_ConfigType . . . . .	192
8.2.1.3	Dem_EventIdType . . . . .	193
8.2.1.4	Dem_EventStatusType . . . . .	193
8.2.1.5	Dem_DebouncingStateType . . . . .	194
8.2.1.6	Dem_DebounceResetStatusType . . . . .	194
8.2.1.7	Dem_UdsStatusByteType . . . . .	195
8.2.1.8	Dem_OperationCycleStateType . . . . .	196
8.2.1.9	Dem_IndicatorStatusType . . . . .	196
8.2.1.10	Dem_DTCKindType . . . . .	196
8.2.1.11	Dem_DTCFormatType . . . . .	197
8.2.1.12	Dem_DTCOriginType . . . . .	197
8.2.1.13	Dem_DTCRequestType . . . . .	198
8.2.1.14	Dem_DTCTranslationFormatType . . . . .	198
8.2.1.15	Dem_DTCSeverityType . . . . .	199

8.2.1.16	Dem_RatioldType . . . . .	200
8.2.1.17	Dem_DTRControlType . . . . .	200
8.2.1.18	Dem_InitMonitorReasonType . . . . .	201
8.2.1.19	Dem_lumprDenomCondIdType . . . . .	201
8.2.1.20	Dem_lumprDenomCondStatusType . . . . .	202
8.2.1.21	Dem_J1939DcmDTCStatusFilterType . . . . .	202
8.2.1.22	Dem_J1939DcmSetClearFilterType . . . . .	203
8.2.1.23	Dem_J1939DcmSetFreezeFrameFilterType . . . . .	203
8.2.1.24	Dem_J1939DcmLampStatusType . . . . .	203
8.2.1.25	Dem_J1939DcmDiagnosticReadiness1Type . . . . .	204
8.2.1.26	Dem_J1939DcmDiagnosticReadiness2Type . . . . .	204
8.2.1.27	Dem_J1939DcmDiagnosticReadiness3Type . . . . .	205
8.2.2	Dem return types . . . . .	206
8.2.2.1	Dem_ReturnGetStatusOfDTCType . . . . .	206
8.2.2.2	Dem_ReturnGetSeverityOfDTCType . . . . .	206
8.2.2.3	Dem_ReturnGetFunctionalUnitOfDTCType . . . . .	207
8.2.2.4	Dem_ReturnSetFilterType . . . . .	207
8.2.2.5	Dem_ReturnGetNumberOfFilteredDTCType . . . . .	207
8.2.2.6	Dem_ReturnGetNextFilteredElementType . . . . .	208
8.2.2.7	Dem_ReturnGetDTCByOccurrenceTimeType . . . . .	208
8.2.2.8	Dem_ReturnDisableDTCRecordUpdateType . . . . .	209
8.2.2.9	Dem_ReturnGetFreezeFrameDataByDTCType . . . . .	209
8.2.2.10	Dem_ReturnGetExtendedDataRecordByDTCType . . . . .	210
8.2.2.11	Dem_ReturnGetSizeOfDataByDTCType . . . . .	210
8.2.2.12	Dem_ReturnClearDTCType . . . . .	211
8.2.2.13	Dem_ReturnControlDTCSettingType . . . . .	211
8.3	Function definitions . . . . .	212
8.3.1	Dem_GetVersionInfo . . . . .	212
8.3.2	Interface ECU State Manager <=> Dem . . . . .	213
8.3.2.1	Dem_Prelit . . . . .	213
8.3.2.2	Dem_Init . . . . .	213
8.3.2.3	Dem_Shutdown . . . . .	214
8.3.3	Interface BSW modules / SW-Components <=> Dem . . . . .	214
8.3.3.1	Dem_ReportErrorStatus . . . . .	214
8.3.3.2	Dem_SetEventAvailable . . . . .	215
8.3.3.3	Dem_SetEventStatus . . . . .	215
8.3.3.4	Dem_ResetEventDebounceStatus . . . . .	216
8.3.3.5	Dem_ResetEventStatus . . . . .	216
8.3.3.6	Dem_PrestoreFreezeFrame . . . . .	217
8.3.3.7	Dem_ClearPrestoredFreezeFrame . . . . .	218
8.3.3.8	Dem_SetOperationCycleState . . . . .	218
8.3.3.9	Dem_GetOperationCycleState . . . . .	219
8.3.3.10	Dem_SetAgingCycleState . . . . .	219
8.3.3.11	Dem_SetWIRStatus . . . . .	220
8.3.3.12	Dem_GetComponentFailed . . . . .	221
8.3.3.13	Dem_GetEventStatus . . . . .	221

8.3.3.14	Dem_GetEventFailed . . . . .	222
8.3.3.15	Dem_GetEventTested . . . . .	222
8.3.3.16	Dem_GetDebouncingOfEvent . . . . .	223
8.3.3.17	Dem_GetDTCOfEvent . . . . .	223
8.3.3.18	Dem_SetEnableCondition . . . . .	224
8.3.3.19	Dem_SetStorageCondition . . . . .	225
8.3.3.20	Dem_GetFaultDetectionCounter . . . . .	225
8.3.3.21	Dem_GetIndicatorStatus . . . . .	226
8.3.3.22	Dem_GetEventFreezeFrameData . . . . .	226
8.3.3.23	Dem_GetEventFreezeFrameDataEx . . . . .	227
8.3.3.24	Dem_GetEventExtendedDataRecord . . . . .	228
8.3.3.25	Dem_GetEventExtendedDataRecordEx . . . . .	229
8.3.3.26	Dem_GetEventMemoryOverflow . . . . .	230
8.3.3.27	Dem_GetNumberOfEventMemoryEntries . . . . .	231
8.3.3.28	Dem_SetComponentAvailable . . . . .	231
8.3.3.29	Dem_SetDTCSuppression . . . . .	232
8.3.3.30	Dem_ClearDTC . . . . .	232
<b>8.3.4</b>	<b>Interface Dcm &lt;=&gt; Dem . . . . .</b>	<b>233</b>
8.3.4.1	Access DTCs and Status Information . . . . .	233
8.3.4.1.1	Dem_DcmGetTranslationType . . . . .	233
8.3.4.1.2	Dem_DcmGetDTCStatusAvailabilityMask . . . . .	234
8.3.4.1.3	Dem_DcmGetStatusOfDTC . . . . .	234
8.3.4.1.4	Dem_DcmGetSeverityOfDTC . . . . .	235
8.3.4.1.5	Dem_DcmGetFunctionalUnitOfDTC . . . . .	235
8.3.4.1.6	Dem_DcmSetDTCFilter . . . . .	236
8.3.4.1.7	Dem_DcmGetNumberOfFilteredDTC . . . . .	238
8.3.4.1.8	Dem_DcmGetNextFilteredDTC . . . . .	238
8.3.4.1.9	Dem_DcmGetNextFilteredDTCAndFDC . . . . .	239
8.3.4.1.10	Dem_DcmGetNextFilteredDTCAndSeverity . . . . .	240
8.3.4.1.11	Dem_DcmSetFreezeFrameRecordFilter . . . . .	241
8.3.4.1.12	Dem_DcmGetNextFilteredRecord . . . . .	241
8.3.4.1.13	Dem_DcmGetDTCByOccurrenceTime . . . . .	242
8.3.4.1.14	Dem_DcmControlDTCStatusChangedNotification . . . . .	243
8.3.4.2	Access extended data records and FreezeFrame data . . . . .	243
8.3.4.2.1	Dem_DcmDisableDTCRecordUpdate . . . . .	243
8.3.4.2.2	Dem_DcmEnableDTCRecordUpdate . . . . .	244
8.3.4.2.3	Dem_DcmGetFreezeFrameDataByDTC . . . . .	244
8.3.4.2.4	Dem_DcmGetSizeOfFreezeFrameByDTC . . . . .	245
8.3.4.2.5	Dem_DcmGetExtendedDataRecordByDTC . . . . .	246
8.3.4.2.6	Dem_DcmGetSizeOfExtendedDataRecordByDTC . . . . .	247
8.3.4.3	DTC storage . . . . .	248
8.3.4.3.1	Dem_DcmCheckClearParameter . . . . .	248
8.3.4.3.2	Dem_DcmClearDTC . . . . .	249
8.3.4.3.3	Dem_DcmDisableDTCSetting . . . . .	249
8.3.4.3.4	Dem_DcmEnableDTCSetting . . . . .	250
<b>8.3.5</b>	<b>OBD-specific Dcm &lt;=&gt; Dem Interfaces . . . . .</b>	<b>250</b>

8.3.5.1	Dem_DcmGetInfoTypeValue08 . . . . .	250
8.3.5.2	Dem_DcmGetInfoTypeValue0B . . . . .	251
8.3.5.3	Dem_DcmReadDataOfPID01 . . . . .	252
8.3.5.4	Dem_DcmReadDataOfPID1C . . . . .	253
8.3.5.5	Dem_DcmReadDataOfPID21 . . . . .	253
8.3.5.6	Dem_DcmReadDataOfPID30 . . . . .	254
8.3.5.7	Dem_DcmReadDataOfPID31 . . . . .	255
8.3.5.8	Dem_DcmReadDataOfPID41 . . . . .	255
8.3.5.9	Dem_DcmReadDataOfPID4D . . . . .	256
8.3.5.10	Dem_DcmReadDataOfPID4E . . . . .	257
8.3.5.11	Dem_DcmReadDataOfPID91 . . . . .	257
8.3.5.12	Dem_DcmReadDataOfOBDFreezeFrame . . . . .	258
8.3.5.13	Dem_DcmGetDTCOfOBDFreezeFrame . . . . .	259
8.3.5.14	Dem_DcmGetAvailableOBDMIDs . . . . .	260
8.3.5.15	Dem_DcmGetNumTIDsOfOBDMID . . . . .	260
8.3.5.16	Dem_DcmGetDTRData . . . . .	261
8.3.6	Interface J1939Dcm <=> Dem . . . . .	262
8.3.6.1	Access DTCs and Status Information . . . . .	262
8.3.6.1.1	Dem_J1939DcmSetDTCFilter . . . . .	262
8.3.6.1.2	Dem_J1939DcmGetNumberOfFilteredDTC . . . . .	262
8.3.6.1.3	Dem_J1939DcmGetNextFilteredDTC . . . . .	263
8.3.6.1.4	Dem_J1939DcmFirstDTCwithLampStatus . . . . .	264
8.3.6.1.5	Dem_J1939DcmGetNextDTCwithLampStatus . . . . .	264
8.3.6.2	DTC storage . . . . .	265
8.3.6.2.1	Dem_J1939DcmClearDTC . . . . .	265
8.3.6.2.2	Dem_J1939DcmSetFreezeFrameFilter . . . . .	265
8.3.6.2.3	Dem_J1939DcmGetNextFreezeFrame . . . . .	266
8.3.6.2.4	Dem_J1939DcmGetNextSPNInFreezeFrame . . . . .	267
8.3.6.3	Reporting . . . . .	267
8.3.6.3.1	Dem_J1939DcmSetRatioFilter . . . . .	267
8.3.6.3.2	Dem_J1939DcmGetNextFilteredRatio . . . . .	268
8.3.6.3.3	Dem_J1939DcmReadDiagnosticReadiness1 . . . . .	269
8.3.6.3.4	Dem_J1939DcmReadDiagnosticReadiness2 . . . . .	269
8.3.6.3.5	Dem_J1939DcmReadDiagnosticReadiness3 . . . . .	270
8.3.7	Interface Dlt <=> Dem . . . . .	270
8.3.7.1	Dem_DltGetMostRecentFreezeFrameRecordData . . . . .	270
8.3.7.2	Dem_DltGetAllExtendedDataRecords . . . . .	271
8.3.8	OBD-specific Interfaces . . . . .	272
8.3.8.1	Dem_SetEventDisabled . . . . .	272
8.3.8.2	Dem_ReplIUMPRFaultDetect . . . . .	273
8.3.8.3	Dem_SetIUMPRDenCondition . . . . .	273
8.3.8.4	Dem_GetIUMPRDenCondition . . . . .	274
8.3.8.5	Dem_ReplIUMPRDenLock . . . . .	275
8.3.8.6	Dem_ReplIUMPRDenRelease . . . . .	275
8.3.8.7	Dem_SetPtoStatus . . . . .	276
8.3.8.8	Dem_ReadDataOfPID01 . . . . .	276

8.3.8.9	Dem_GetDataOfPID21 . . . . .	277
8.3.8.10	Dem_SetDataOfPID21 . . . . .	278
8.3.8.11	Dem_SetDataOfPID31 . . . . .	278
8.3.8.12	Dem_SetDataOfPID4D . . . . .	279
8.3.8.13	Dem_SetDataOfPID4E . . . . .	279
8.3.8.14	Dem_SetPfcCycleQualified . . . . .	280
8.3.8.15	Dem_GetPfcCycleQualified . . . . .	281
8.3.8.16	Dem_SetClearDTC . . . . .	281
8.3.8.17	Dem_DcmGetDTCSeverityAvailabilityMask . . . . .	282
8.3.8.18	Dem_GetB1Counter . . . . .	283
8.3.8.19	Dem_SetDTR . . . . .	283
<b>8.4</b>	<b>Expected Interfaces . . . . .</b>	<b>284</b>
8.4.1	Mandatory Interfaces . . . . .	284
8.4.2	Optional Interfaces . . . . .	284
8.4.3	Configurable interfaces . . . . .	285
8.4.3.1	Interface BSW modules / SW-Components <=> Dem . . . . .	286
8.4.3.1.1	InitMonitorForEvent . . . . .	286
8.4.3.1.2	DemTriggerOnComponentStatus . . . . .	287
8.4.3.2	EventStatusChanged . . . . .	287
8.4.3.3	DTCStatusChanged . . . . .	288
8.4.3.3.1	SetClearDTC . . . . .	288
8.4.3.3.2	EventDataChanged . . . . .	289
8.4.3.3.3	ClearEventAllowed . . . . .	290
8.4.3.3.4	ReadDataElement . . . . .	290
8.4.3.4	GetFaultDetectionCounter . . . . .	291
<b>8.5</b>	<b>Scheduled functions . . . . .</b>	<b>291</b>
8.5.1	Dem_MainFunction . . . . .	291
8.5.2	Runnable Entity MainFunction . . . . .	292
<b>8.6</b>	<b>Service Interfaces . . . . .</b>	<b>292</b>
8.6.1	Sender-Receiver-Interfaces . . . . .	292
8.6.1.1	DataServices_Data . . . . .	292
8.6.2	Client-Server-Interfaces . . . . .	293
8.6.2.1	AgingCycle . . . . .	293
8.6.2.2	CallbackClearEventAllowed . . . . .	294
8.6.2.3	CallbackEventStatusChange . . . . .	294
8.6.2.4	CallbackEventDataChanged . . . . .	295
8.6.2.5	CallbackGetFaultDetectCounter . . . . .	296
8.6.2.6	CallbackInitMonitorForEvent . . . . .	296
8.6.2.7	CallbackDTCStatusChange . . . . .	297
8.6.2.8	GeneralCallbackEventDataChanged . . . . .	298
8.6.2.9	GeneralCallbackEventStatusChange . . . . .	299
8.6.2.10	CallbackComponentStatusChanged . . . . .	299
8.6.2.11	Service Interface Cddlf . . . . .	300
8.6.2.12	Service Interface DTCSuppression . . . . .	301
8.6.2.13	DataServices_{Data} . . . . .	302
8.6.2.14	DcmIf . . . . .	303

8.6.2.15	DTRCentralReport . . . . .	305
8.6.2.16	EnableCondition . . . . .	306
8.6.2.17	DiagnosticMonitor . . . . .	307
8.6.2.18	EventStatus . . . . .	308
8.6.2.19	EventAvailable . . . . .	309
8.6.2.20	DiagnosticInfo . . . . .	310
8.6.2.21	GeneralDiagnosticInfo . . . . .	315
8.6.2.22	IndicatorStatus . . . . .	321
8.6.2.23	IUMPRDenominator . . . . .	321
8.6.2.24	IUMPRDenominatorCondition . . . . .	322
8.6.2.25	IUMPRNumerator . . . . .	323
8.6.2.26	OperationCycle . . . . .	324
8.6.2.27	EvMemOverflowIndication . . . . .	325
8.6.2.28	PfcCycleQualified . . . . .	326
8.6.2.29	PowerTakeOff . . . . .	327
8.6.2.30	SetClearDTC . . . . .	327
8.6.2.31	GetDataOfPID21 . . . . .	328
8.6.2.32	SetDataOfPID21 . . . . .	329
8.6.2.33	SetDataOfPID31 . . . . .	330
8.6.2.34	StorageCondition . . . . .	330
8.6.3	Implementation Data Types . . . . .	331
8.6.4	Ports . . . . .	332
8.6.4.1	Dem_AgingCycle . . . . .	332
8.6.4.2	Dem_CBDataEvt . . . . .	332
8.6.4.3	Dem_CBFaultDetectCtr . . . . .	333
8.6.4.4	Dem_CBInitEvt . . . . .	333
8.6.4.5	Dem_CBStatusDTC . . . . .	334
8.6.4.6	Dem_CBStatusEvt . . . . .	334
8.6.4.7	Dem_CBStatusComp . . . . .	334
8.6.4.8	Dem_Cdd . . . . .	335
8.6.4.9	Dem_ControlDTCSuppression . . . . .	335
8.6.4.10	Dem_ControlEventAvailable . . . . .	335
8.6.4.11	DataServices . . . . .	336
8.6.4.12	Dem_Dcm . . . . .	336
8.6.4.13	Dem_DTR . . . . .	336
8.6.4.14	Dem_EnableCond . . . . .	337
8.6.4.15	Dem_Event . . . . .	337
8.6.4.16	Dem_EventStatus . . . . .	337
8.6.4.17	Dem_EvtInfo . . . . .	337
8.6.4.18	Dem_GeneralCBDataEvt . . . . .	338
8.6.4.19	Dem_GeneralCBStatusEvt . . . . .	338
8.6.4.20	Dem_GeneralEvtInfo . . . . .	338
8.6.4.21	Dem_IndStatus . . . . .	339
8.6.4.22	Dem_IUMPRDenominator . . . . .	339
8.6.4.23	Dem_IUMPRDenominatorCondition . . . . .	339
8.6.4.24	Dem_IUMPRNumerator . . . . .	340

8.6.4.25	Dem_OpCycle . . . . .	340
8.6.4.26	Dem_OverflowIndMirrorMemory . . . . .	340
8.6.4.27	Dem_OverflowIndPermanentMemory . . . . .	341
8.6.4.28	Dem_OverflowIndPrimaryMemory . . . . .	341
8.6.4.29	Dem_OverflowIndUserDefinedMemory . . . . .	341
8.6.4.30	Dem_PfcCycleQualified . . . . .	342
8.6.4.31	Dem_PowerTakeOffStatus . . . . .	342
8.6.4.32	Dem_SetClearDTC_dependend . . . . .	342
8.6.4.33	Dem_SetClearDTC_master . . . . .	343
8.6.4.34	Dem_GetDataOfPID21 . . . . .	343
8.6.4.35	Dem_SetDataOfPID21 . . . . .	343
8.6.4.36	Dem_SetDataOfPID31 . . . . .	344
8.6.4.37	Dem_StorageCond . . . . .	344
<b>9 Sequence Diagrams</b>		<b>345</b>
9.1	ControlDTCSetting . . . . .	345
9.2	Dem_DcmClearDTC . . . . .	345
9.3	Dem_DcmGetDTCByOccurrenceTime . . . . .	346
9.4	Dem_DcmGetExtendedDataRecordByDTC . . . . .	347
9.5	Dem_DcmGetStatusOfDTC . . . . .	347
9.6	Dem_DcmGetFreezeFrameDataByDTC . . . . .	348
9.7	GetOBDFaultInformation . . . . .	349
9.8	ReportDTCByStatusMask . . . . .	350
9.9	FiM_DemTriggerOnEventStatus . . . . .	350
9.10	ProcessEvent (Example) . . . . .	351
<b>10 Configuration specification</b>		<b>352</b>
10.1	How to read this chapter . . . . .	352
10.2	Containers and configuration parameters . . . . .	352
10.2.1	Variants . . . . .	352
10.2.2	Dem . . . . .	356
10.2.3	DemGeneral . . . . .	357
10.2.4	DemGeneralOBD . . . . .	374
10.2.5	DemGeneralJ19139 . . . . .	382
10.2.6	DemOperationCycle . . . . .	387
10.2.7	DemEnableCondition . . . . .	389
10.2.8	DemEnableConditionGroup . . . . .	390
10.2.9	DemStorageCondition . . . . .	391
10.2.10	DemStorageConditionGroup . . . . .	392
10.2.11	DemIndicator . . . . .	393
10.2.12	DemNvRamBlockId . . . . .	394
10.2.13	DemGroupOfDTC . . . . .	394
10.2.14	DemRatio . . . . .	395
10.2.15	DemPrimaryMemory . . . . .	398
10.2.16	DemMirrorMemory . . . . .	399
10.2.17	DemUserDefinedMemory . . . . .	400
10.2.18	DemCallbackDTCStatusChanged . . . . .	401

10.2.19	DemCallbackOBDDTCStatusChanged . . . . .	402
10.2.20	DemCallbackJ1939DTCStatusChanged . . . . .	403
10.2.21	DemConfigSet . . . . .	404
10.2.22	DemComponent . . . . .	404
10.2.23	DemObdDTC . . . . .	406
10.2.24	DemDTC . . . . .	408
10.2.25	DemJ1939Node . . . . .	412
10.2.26	DemPidClass . . . . .	412
10.2.27	DemPidDataElement . . . . .	413
10.2.28	DemDTCAtributes . . . . .	414
10.2.29	DemEventParameter . . . . .	424
10.2.30	DemIndicatorAttribute . . . . .	430
10.2.31	DemDebounceAlgorithmClass . . . . .	432
10.2.32	DemDebounceCounterBased . . . . .	432
10.2.33	DemDebounceCounterBasedClass . . . . .	433
10.2.34	DemDebounceTimeBase . . . . .	437
10.2.35	DemDebounceTimeBaseClass . . . . .	438
10.2.36	DemDebounceMonitorInternal . . . . .	440
10.2.37	DemCallbackGetFDC . . . . .	440
10.2.38	DemCallbackClearEventAllowed . . . . .	441
10.2.39	DemCallbackEventDataChanged . . . . .	442
10.2.40	DemCallbackEventStatusChanged . . . . .	443
10.2.41	DemCallbackInitMForE . . . . .	444
10.2.42	DemDtrs . . . . .	445
10.2.43	DemDtr . . . . .	446
10.2.44	DemFreezeFrameClass . . . . .	450
10.2.45	DemDidClass . . . . .	450
10.2.46	DemJ1939FreezeFrameClass . . . . .	451
10.2.47	DemSPNClass . . . . .	452
10.2.48	DemFreezeFrameRecordClass . . . . .	453
10.2.49	DemFreezeFrameRecNumClass . . . . .	454
10.2.50	DemExtendedDataClass . . . . .	455
10.2.51	DemExtendedDataRecordClass . . . . .	456
10.2.52	DemDataElementClass . . . . .	458
10.2.53	DemDataElementInstance . . . . .	459
10.2.54	DemInternalDataElementClass . . . . .	460
10.2.55	DemExternalCSDDataElementClass . . . . .	461
10.2.56	DemExternalSRDataElementClass . . . . .	463
10.2.57	DemSRDataElementClass . . . . .	465
10.2.58	DemSubElementInDataElementInstance . . . . .	465
10.2.59	DemSubElementInImplDataElementInstance . . . . .	466
10.2.60	DemDiagnosisScaling . . . . .	467
10.2.61	DemAlternativeDataInterface . . . . .	468
10.2.62	DemAlternativeDataType . . . . .	469
10.2.63	DemAlternativeDataProps . . . . .	470
10.2.64	DemLinearScale . . . . .	471

10.2.65 DemTextTableMapping . . . . .	472
10.3 Published information . . . . .	474
11 Not Applicable Requirements	475

## 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 <a href="#">event memory</a> .
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 <a href="#">event memory</a> .
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 <a href="#">7.5</a> )
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 (Denominator <sub>m</sub> ) 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.
DtcGroupAllDtcs	Grouping of all configured DTCS (representation is 0xFFFFFFF).
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.

Acronym	Description
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 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 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 DTC 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 related data	Event related data is additional data, e.g. sensor values or time stamp/mileage, which is stored with an event in an <a href="#">event memory</a> . ISO defines two types of event related data: freeze frames (snapshot records) and extended data.
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 <a href="#">event memory</a> .
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 [1]. 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[1].
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: $IUPR_m = \text{Numerator}_m / \text{Denominator}_m$
Master ECU	As a primary ECU a Master ECU stores "it's own" and "reported errors" of related dep. / sec ECUs in its <a href="#">event memory</a> . 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 (Numerator <sub>m</sub> ) 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.

Acronym	Description
NvM is marked for NvM_WriteAll	The <a href="#">Dem</a> has called NvM_SetRamBlockStatus() to set the according NvM Block to be written by NvM_WriteAll()
Operating cycle	An 'Operating 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: <ul style="list-style-type: none"> <li>• <a href="#">Master ECU</a> (one per vehicle)</li> <li>• Primary ECU (several per vehicle)</li> <li>• <a href="#">Dependent / Secondary ECUs</a> (several per vehicle)</li> </ul>
P-Code	Power train code
PFC cycle	Permanent fault code - driving cycle (OBD Term)
Positive Callback from NvM	The <a href="#">Dem</a> module shall use the <a href="#">APIs</a> NvM_WriteBlock and NvM_GetErrorStatus of the NVRAMManager [5], if there is the necessity to store data between <a href="#">Dem_Init</a> and <a href="#">Dem_Shutdown</a> . Furthermore the <a href="#">API</a> 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 its <a href="#">event memory</a>
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 DTC Status Byte.
Triggered to NvM	The Dem module shall use the <a href="#">API</a> NvM_WriteBlock of the NVRAMManager [5], if there is the necessity to trigger the storage of data between <a href="#">Dem_Init</a> and <a href="#">Dem_Shutdown</a> . Furthermore the <a href="#">Dem</a> module shall wait for positive response of NvM_WriteBlock if request has been accepted.

## 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
Dlt	Diagnostic Log and Trace

<b>Abbreviation</b>	<b>Description</b>
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)
MIL	Malfunction Indicator Light (SAE J1979) or Lamp (SAE J1939)
NVRAM	Non volatile RAM
OBD	On-Board-Diagnostics
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

#### Bibliography

- [1] Unified diagnostic services (UDS) - Part 1: Specification and requirements (Release 2013-03)  
<http://www.iso.org>
- [2] Road vehicles - Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements - Part 1: General information and use case definition  
<http://www.iso.org>
- [3] Unified diagnostic services (UDS) - Part 1: Specification and requirements (Release 2006-12)  
<http://www.iso.org>
- [4] Glossary  
AUTOSAR\_TR\_Glossary
- [5] Specification of NVRAM Manager  
AUTOSAR\_SWS\_NVRAMManager
- [6] General Specification of Basic Software Modules  
AUTOSAR\_SWS\_BSWGeneral
- [7] Specification of a Diagnostic Communication Manager for SAE J1939  
AUTOSAR\_SWS\_SAEJ1939DiagnosticCommunicationManager
- [8] Requirements on Function Inhibition Manager  
AUTOSAR\_SRS\_FunctionInhibitionManager
- [9] Specification of Diagnostic Log and Trace  
AUTOSAR\_SWS\_DiagnosticLogAndTrace
- [10] Specification of Diagnostic Communication Manager  
AUTOSAR\_SWS\_DiagnosticCommunicationManager
- [11] Requirements on Diagnostic  
AUTOSAR\_SRS\_Diagnostic
- [12] General Requirements on Basic Software Modules  
AUTOSAR\_SRS\_BSWGeneral
- [13] Road vehicles - Communication between vehicle and external equipment for emission-related diagnostic - Part 5: Emission-related diagnostic services.  
<http://www.iso.org>
- [14] SAE J2012-DA Digital

- [15] SAE J1939-73 Application Layer - Diagnostics
- [16] Road vehicles - Interchange of digital information on electrical connections between towing and towed vehicles - Part 4: Diagnostic communication  
<http://www.iso.org>
- [17] Communication  
<http://portal.osek-vdx.org/files/pdf/specs/osekcom303.pdf>
- [18] SAE J1979
- [19] SAE J1979-DA Digital Annex of E/E Diagnostic Test Modes
- [20] Title 13, California Code Regulations, Section 1971.1, On-Board Diagnostic System Requirements for 2013 and Subsequent Model-Year Heavy-Duty Engines (HD OBD)
- [21] 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>
- [22] Road vehicles - Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements - Part 3: Common message dictionary  
<http://www.iso.org>
- [23] Requirements on Diagnostic Log and Trace  
AUTOSAR\_SRS\_DiagnosticLogAndTrace

### 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]** [There shall only be one Dem module available per ECU.]  
*(SRS\_Diag\_04002)*

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 constraints 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.

Callbacks for Event and DTCStatusChange are not deterministic in rare cases, wherefore it shall not be used for safety-relevant use-cases. Consider the general statement in ISO-14229-1[1] that it is not recommended to link the [DTC](#) status with failsafe strategies.

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, mirror 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 [subsubsection 7.10.2.2](#)).

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).

Post build time loadable configuration is not supported by the Dem configuration model and respective parameter set.

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

## 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:

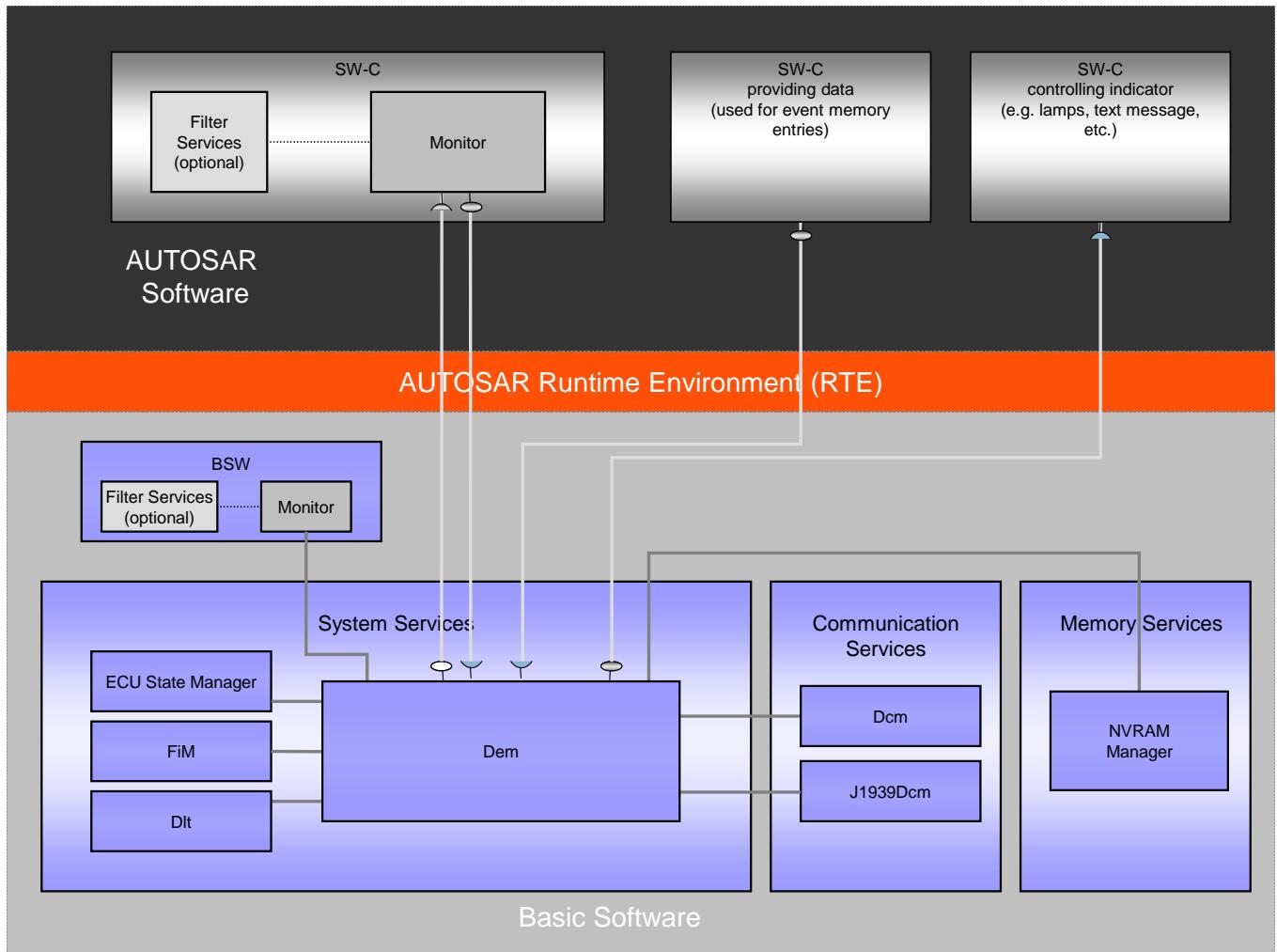


Figure 5.1: Dependencies of the **Dem** to other software modules

- 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 event status in order to stop or release function entities according to assigned dependencies.
- The **Diagnostic Log and Trace (Dlt)** (refer to [9, SWS Dlt]) provides a generic Logging and Tracing functionality for the Dem. The Dem informs and updates the Diagnostic Log and Trace (Dlt) upon changes of the event status and provides access on the current event related data in order to log and trace this information.
- The **Diagnostic Communication Manager (Dcm)** (refer to [10, SWS Dcm]) is in charge of the UDS and SAE J1979 communication path and execution of diag-

nostic 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 event 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 event 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 the chapter 5.1.6 “Code File Structure” in SWS\_BSWGeneral[6].  
(5.1.6.3 Link time configuration source & 5.1.6.4 Post-build time configuration source)

### 5.1.2 Header file structure

[SWS\_Dem\_00151] [ The header-file structure shall contain the following files named (in addition to SWS\_BSWGeneral):

- Dem\_Dcm.h - for Dem APIs used by the Dcm exclusively
- Dem\_Types.h - for Dem data types (not defined in Rte\_Dem\_Type.h)
- Dcm\_Types.h - for all imported Dcm types (refer to chapter 8.1)

- J1939Dcm\_Types.h - for all imported J1939Dcm types (optional)
- FiM.h - for Function Inhibition Manager symbols (optional)
- Dlt.h - for Diagnostic Log & Trace symbols (optional)
- NvM.h - for NVRAM Manager symbols
- <...>.h - contains all C-callback declarations (refer to DemHeaderFileInclusion in DemGeneral) configured for the Dem

」(SRS\_BSW\_00447)

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.

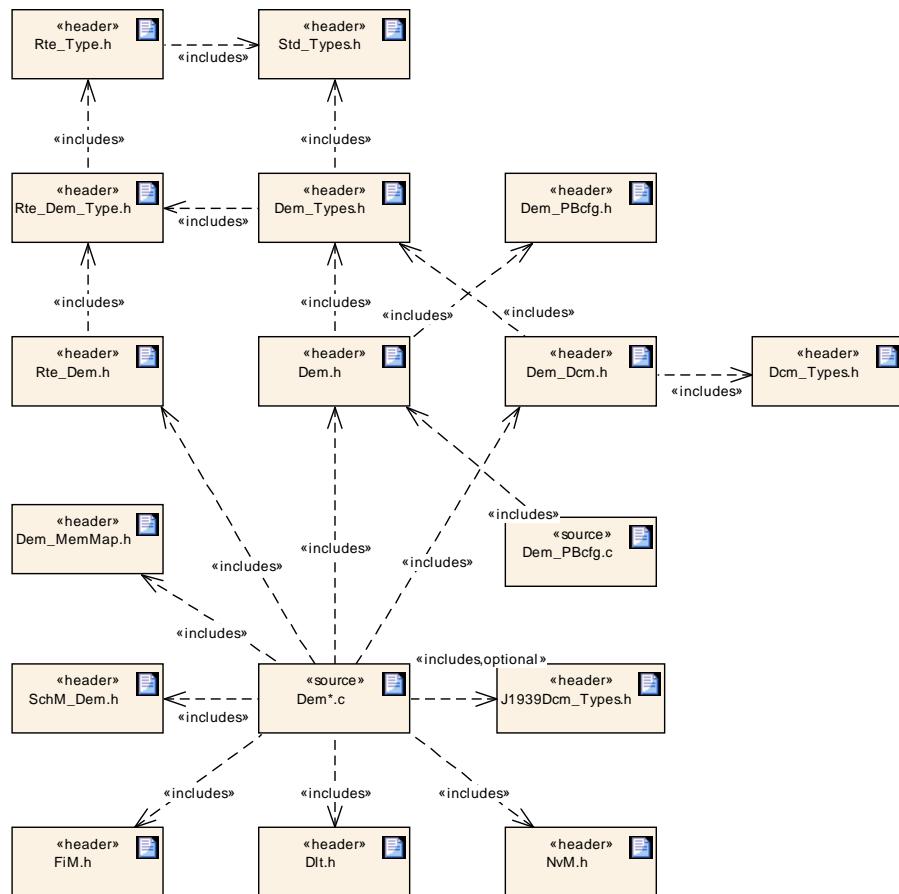


Figure 5.2: Header file structure

## 5.2 Integration Hints

Due to cross-module communication it is recommended 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 [11] as well as [12] and links to the fulfillment of these.

Requirement	Description	Satisfied by
[SRS_BSW_00005]	Modules of the $\mu$ C Abstraction Layer (MCAL) may not have hard coded horizontal interfaces	[SWS_Dem_00999]
[SRS_BSW_00101]	The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function	[SWS_Dem_00181] [SWS_Dem_00340] [SWS_Dem_00550]
[SRS_BSW_00161]	The AUTOSAR Basic Software shall provide a microcontroller abstraction layer which provides a standardized interface to higher software layers	[SWS_Dem_00999]
[SRS_BSW_00162]	The AUTOSAR Basic Software shall provide a hardware abstraction layer	[SWS_Dem_00999]
[SRS_BSW_00164]	The Implementation of interrupt service routines shall be done by the Operating System, complex drivers or modules	[SWS_Dem_00999]
[SRS_BSW_00168]	SW components shall be tested by a function defined in a common API in the Basis-SW	[SWS_Dem_00999]
[SRS_BSW_00170]	The AUTOSAR SW Components shall provide information about their dependency from faults, signal qualities, driver demands	[SWS_Dem_00999]
[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] [SWS_Dem_00999]
[SRS_BSW_00301]	All AUTOSAR Basic Software Modules shall only import the necessary information	[SWS_Dem_00176]
[SRS_BSW_00324]	No description	[SWS_Dem_00999]
[SRS_BSW_00326]	No description	[SWS_Dem_00999]
[SRS_BSW_00327]	Error values naming convention	[SWS_Dem_00999]

Requirement	Description	Satisfied by
[SRS_BSW_00331]	All Basic Software Modules shall strictly separate error and status information	[SWS_Dem_00999]
[SRS_BSW_00334]	All Basic Software Modules shall provide an XML file that contains the meta data	[SWS_Dem_00267] [SWS_Dem_00268]
[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_00999]
[SRS_BSW_00338]	No description	[SWS_Dem_00999]
[SRS_BSW_00339]	Reporting of production relevant error status	[SWS_Dem_00107] [SWS_Dem_00167] [SWS_Dem_00207] [SWS_Dem_00851] [SWS_Dem_00852] [SWS_Dem_00999] [SWS_Dem_01079]
[SRS_BSW_00341]	Module documentation shall contain all needed informations	[SWS_Dem_00999]
[SRS_BSW_00345]	BSW Modules shall support pre-compile configuration	[SWS_Dem_00267] [SWS_Dem_00268]
[SRS_BSW_00347]	A Naming separation of different instances of BSW drivers shall be in place	[SWS_Dem_00999]
[SRS_BSW_00348]	All AUTOSAR standard types and constants shall be placed and organized in a standard type header file	[SWS_Dem_00999]
[SRS_BSW_00350]	All AUTOSAR Basic Software Modules shall apply a specific naming rule for enabling/disabling the detection and reporting of development errors	[SWS_Dem_00999]
[SRS_BSW_00353]	All integer type definitions of target and compiler specific scope shall be placed and organized in a single type header	[SWS_Dem_00999]
[SRS_BSW_00357]	For success/failure of an API call a standard return type shall be defined	[SWS_Dem_00999]
[SRS_BSW_00359]	All AUTOSAR Basic Software Modules callback functions shall avoid return types other than void if possible	[SWS_Dem_00999]
[SRS_BSW_00360]	AUTOSAR Basic Software Modules callback functions are allowed to have parameters	[SWS_Dem_00999]

Requirement	Description	Satisfied by
[SRS_BSW_00361]	All mappings of not standardized keywords of compiler specific scope shall be placed and organized in a compiler specific type and keyword header	[SWS_Dem_00999]
[SRS_BSW_00369]	All AUTOSAR Basic Software Modules shall not return specific development error codes via the API	[SWS_Dem_00999]
[SRS_BSW_00374]	All Basic Software Modules shall provide a readable module vendor identification	[SWS_Dem_00999]
[SRS_BSW_00375]	Basic Software Modules shall report wake-up reasons	[SWS_Dem_00999]
[SRS_BSW_00379]	All software modules shall provide a module identifier in the header file and in the module XML description file.	[SWS_Dem_00999]
[SRS_BSW_00382]	No description	[SWS_Dem_00999]
[SRS_BSW_00387]	No description	[SWS_Dem_00999]
[SRS_BSW_00396]	The Basic Software Module specifications shall specify the supported configuration classes for changing values and multiplicities for each parameter/container	[SWS_Dem_00267] [SWS_Dem_00268]
[SRS_BSW_00397]	The configuration parameters in pre-compile time are fixed before compilation starts	[SWS_Dem_00267] [SWS_Dem_00268]
[SRS_BSW_00398]	The link-time configuration is achieved on object code basis in the stage after compiling and before linking	[SWS_Dem_00267] [SWS_Dem_00268]
[SRS_BSW_00399]	Parameter-sets shall be located in a separate segment and shall be loaded after the code	[SWS_Dem_00267] [SWS_Dem_00268]
[SRS_BSW_00400]	Parameter shall be selected from multiple sets of parameters after code has been loaded and started	[SWS_Dem_00267] [SWS_Dem_00268]
[SRS_BSW_00401]	Documentation of multiple instances of configuration parameters shall be available	[SWS_Dem_00267] [SWS_Dem_00268]

Requirement	Description	Satisfied by
[SRS_BSW_00402]	Each module shall provide version information	[SWS_Dem_00177]
[SRS_BSW_00404]	BSW Modules shall support post-build configuration	[SWS_Dem_00267] [SWS_Dem_00268]
[SRS_BSW_00405]	BSW Modules shall support multiple configuration sets	[SWS_Dem_00267] [SWS_Dem_00268]
[SRS_BSW_00406]	A static status variable denoting if a BSW module is initialized shall be initialized with value 0 before any APIs of the BSW module is called	[SWS_Dem_00124] [SWS_Dem_00169] [SWS_Dem_00364] [SWS_Dem_01112]
[SRS_BSW_00407]	Each BSW module shall provide a function to read out the version information of a dedicated module implementation	[SWS_Dem_00177]
[SRS_BSW_00417]	Software which is not part of the SW-C shall report error events only after the DEM is fully operational.	[SWS_Dem_00107]
[SRS_BSW_00433]	Main processing functions are only allowed to be called from task bodies provided by the BSW Scheduler	[SWS_Dem_00999]
[SRS_BSW_00434]	No description	[SWS_Dem_00999]
[SRS_BSW_00447]	Standardizing Include file structure of BSW Modules Implementing Autosar Service	[SWS_Dem_00151]
[SRS_Diag_04000]	The DEM and DCM shall support the Diagnostic Standard UDS (ISO14229-1)	[SWS_Dem_00013] [SWS_Dem_00061] [SWS_Dem_00080] [SWS_Dem_00645]
[SRS_Diag_04001]	The DEM and DCM shall support the Diagnostic Standard OBD (ISO15031-5)	[SWS_Dem_00299] [SWS_Dem_00359] [SWS_Dem_00645] [SWS_Dem_00697] [SWS_Dem_00745] [SWS_Dem_00753] [SWS_Dem_00763] [SWS_Dem_01076] [SWS_Dem_01077] [SWS_Dem_01092]
[SRS_Diag_04002]	The Diagnostic event (fault) management shall be established as Basic SW Module	[SWS_Dem_00126]
[SRS_Diag_04005]	The DCM shall manage Security Access level handling	[SWS_Dem_00999]
[SRS_Diag_04006]	Session handling is managed by DCM	[SWS_Dem_00999]

Requirement	Description	Satisfied by
[SRS_Diag_04007]	The DCM shall provide a diagnostic service handling for the SW-Components which are using the DCM	[SWS_Dem_00999]
[SRS_Diag_04010]	The DEM module and DCM module shall ensure interaction in order to fulfill ISO 14229-1 and ISO 15031-5	[SWS_Dem_00060] [SWS_Dem_00079] [SWS_Dem_00172] [SWS_Dem_00204] [SWS_Dem_00210] [SWS_Dem_00216] [SWS_Dem_00228] [SWS_Dem_00264] [SWS_Dem_00277] [SWS_Dem_00415] [SWS_Dem_00417] [SWS_Dem_00439] [SWS_Dem_00584] [SWS_Dem_00645] [SWS_Dem_00657] [SWS_Dem_00671] [SWS_Dem_00827]
[SRS_Diag_04015]	The DCM shall support timing handling according to ISO15765-3	[SWS_Dem_00999]
[SRS_Diag_04016]	The DCM shall support "Busy handling" by sending a negative response 0x78	[SWS_Dem_00999]
[SRS_Diag_04019]	The DCM shall confirm transmitting if complete to continue processing	[SWS_Dem_00999]
[SRS_Diag_04020]	DCM shall suppress responses to diagnostic tool requests	[SWS_Dem_00999]
[SRS_Diag_04021]	The DCM shall support the handling of different diagnostic sessions in parallel	[SWS_Dem_00999]
[SRS_Diag_04024]	The DCM module shall be able to access and handle specific data elements and data element groups if requested by an external scan tool	[SWS_Dem_00479] [SWS_Dem_01194]
[SRS_Diag_04030]	The DEM shall provide an interface via the RTE to monitoring SW components for reporting and processing diagnostic test results	[SWS_Dem_00331] [SWS_Dem_00608] [SWS_Dem_00744] [SWS_Dem_00756] [SWS_Dem_00759] [SWS_Dem_00850]
[SRS_Diag_04031]	The DEM shall 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]
[SRS_Diag_04032]	Different diagnostic addresses shall be supported by multiple (physical) channels	[SWS_Dem_00999]

Requirement	Description	Satisfied by
[SRS_Diag_04033]	The DCM shall support the upload/download services for reading/writing data in an ECU in an extended and manufacturer specific diagnostic session	[SWS_Dem_00999]
[SRS_Diag_04036]	The DCM shall check the format of diagnostic service	[SWS_Dem_00999]
[SRS_Diag_04057]	The DEM shall support a classification of events for series production, OBD and expert usage	[SWS_Dem_00057] [SWS_Dem_00518] [SWS_Dem_00593] [SWS_Dem_00649] [SWS_Dem_00906]
[SRS_Diag_04058]	The DCM module shall be able to access different event memories provided by the DEM module	[SWS_Dem_00171]
[SRS_Diag_04059]	The DCM and DEM shall support the configuration of timing parameters	[SWS_Dem_00999]
[SRS_Diag_04061]	The DEM shall provide mechanisms to distinguish between the reported fault of different applications and basic software modules	[SWS_Dem_00153] [SWS_Dem_00602] [SWS_Dem_00616] [SWS_Dem_00621] [SWS_Dem_00745] [SWS_Dem_01092]
[SRS_Diag_04063]	The DEM module shall process a dedicated event identifier (EventId) for each monitoring path to support an autonomous handling of different events/faults	[SWS_Dem_00153] [SWS_Dem_00154]
[SRS_Diag_04064]	The DEM shall support buffers of scalable sizes for the storage of the events, status information and environmental data	[SWS_Dem_00999]
[SRS_Diag_04065]	The DEM and DCM shall be able to remove a specific event or event groups from the configured event memory	[SWS_Dem_00009] [SWS_Dem_00077] [SWS_Dem_00569] [SWS_Dem_00570] [SWS_Dem_00571] [SWS_Dem_00572] [SWS_Dem_00573] [SWS_Dem_00661] [SWS_Dem_00662] [SWS_Dem_00663] [SWS_Dem_00664] [SWS_Dem_01057] [SWS_Dem_01059] [SWS_Dem_01061] [SWS_Dem_01075] [SWS_Dem_01185]
[SRS_Diag_04066]	The DEM module shall provide different event memories which have to be configurable per event	[SWS_Dem_00010] [SWS_Dem_00162] [SWS_Dem_00212] [SWS_Dem_00236] [SWS_Dem_00238] [SWS_Dem_00239] [SWS_Dem_00240] [SWS_Dem_00241] [SWS_Dem_00548]

Requirement	Description	Satisfied by
[SRS_Diag_04067]	The DCM and DEM shall provide the diagnostic status information according to ISO 14229-1	[SWS_Dem_00011] [SWS_Dem_00016] [SWS_Dem_00051] [SWS_Dem_00052] [SWS_Dem_00053] [SWS_Dem_00059] [SWS_Dem_00060] [SWS_Dem_00187] [SWS_Dem_00330] [SWS_Dem_00333] [SWS_Dem_00409] [SWS_Dem_00523] [SWS_Dem_00524] [SWS_Dem_00529] [SWS_Dem_00566] [SWS_Dem_00638] [SWS_Dem_00700] [SWS_Dem_01186]
[SRS_Diag_04068]	The DEM module shall provide event specific debounce algorithms	[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_00430] [SWS_Dem_00431] [SWS_Dem_00432] [SWS_Dem_00433] [SWS_Dem_00434] [SWS_Dem_00435] [SWS_Dem_00436] [SWS_Dem_00437] [SWS_Dem_00438] [SWS_Dem_00439] [SWS_Dem_00526] [SWS_Dem_00527] [SWS_Dem_00643] [SWS_Dem_00684] [SWS_Dem_00684] [SWS_Dem_00772] [SWS_Dem_00774] [SWS_Dem_00778] [SWS_Dem_00779] [SWS_Dem_00818] [SWS_Dem_00985]
[SRS_Diag_04069]	The DEM module shall provide event specific information of warning indicators requested by SW-Cs or other BSW modules	[SWS_Dem_00046] [SWS_Dem_00500] [SWS_Dem_00506] [SWS_Dem_00510] [SWS_Dem_00511] [SWS_Dem_00533] [SWS_Dem_00535] [SWS_Dem_00701] [SWS_Dem_00865] [SWS_Dem_00886] [SWS_Dem_00896] [SWS_Dem_00967]
[SRS_Diag_04070]	The DEM module shall process the order of the event occurrences in an appropriate and obvious manner	[SWS_Dem_00161] [SWS_Dem_00219] [SWS_Dem_00221] [SWS_Dem_00410] [SWS_Dem_00412] [SWS_Dem_00787] [SWS_Dem_00852]
[SRS_Diag_04071]	The DEM module shall process events according to their defined importance like priority and/or severity	[SWS_Dem_00033] [SWS_Dem_00382] [SWS_Dem_00383] [SWS_Dem_00692]
[SRS_Diag_04072]	The DEM shall provide additional event information to report the occurrence of an event by km-stamp, driving cycles or time	[SWS_Dem_00999]
[SRS_Diag_04073]	DEM shall process combined events which shall consist of several different events	[SWS_Dem_00024] [SWS_Dem_00163] [SWS_Dem_00442] [SWS_Dem_00539] [SWS_Dem_00541] [SWS_Dem_01051] [SWS_Dem_01052] [SWS_Dem_01053]

Requirement	Description	Satisfied by
[SRS_Diag_04074]	The DEM module shall process event related data (e.g. freeze frames and extended data records)	<a href="#">[SWS_Dem_00039]</a> <a href="#">[SWS_Dem_00040]</a> <a href="#">[SWS_Dem_00071]</a> <a href="#">[SWS_Dem_00074]</a> <a href="#">[SWS_Dem_00075]</a> <a href="#">[SWS_Dem_00076]</a> <a href="#">[SWS_Dem_00188]</a> <a href="#">[SWS_Dem_00189]</a> <a href="#">[SWS_Dem_00191]</a> <a href="#">[SWS_Dem_00193]</a> <a href="#">[SWS_Dem_00225]</a> <a href="#">[SWS_Dem_00271]</a> <a href="#">[SWS_Dem_00464]</a> <a href="#">[SWS_Dem_00465]</a> <a href="#">[SWS_Dem_00576]</a> <a href="#">[SWS_Dem_00582]</a> <a href="#">[SWS_Dem_00585]</a> <a href="#">[SWS_Dem_00630]</a> <a href="#">[SWS_Dem_00631]</a> <a href="#">[SWS_Dem_00775]</a> <a href="#">[SWS_Dem_00796]</a> <a href="#">[SWS_Dem_00807]</a> <a href="#">[SWS_Dem_00969]</a> <a href="#">[SWS_Dem_00995]</a> <a href="#">[SWS_Dem_00996]</a> <a href="#">[SWS_Dem_00997]</a>
[SRS_Diag_04075]	The DEM module shall support a configuration to assign specific event to a customer specific DTC	<a href="#">[SWS_Dem_00999]</a>
[SRS_Diag_04076]	The DEM module shall provide a set of system cycles that may qualify the event in an additional manner	<a href="#">[SWS_Dem_00019]</a> <a href="#">[SWS_Dem_00194]</a> <a href="#">[SWS_Dem_00338]</a> <a href="#">[SWS_Dem_00481]</a> <a href="#">[SWS_Dem_00482]</a> <a href="#">[SWS_Dem_00483]</a> <a href="#">[SWS_Dem_00484]</a> <a href="#">[SWS_Dem_00577]</a> <a href="#">[SWS_Dem_00601]</a> <a href="#">[SWS_Dem_00602]</a> <a href="#">[SWS_Dem_00673]</a> <a href="#">[SWS_Dem_00693]</a> <a href="#">[SWS_Dem_00698]</a> <a href="#">[SWS_Dem_00773]</a> <a href="#">[SWS_Dem_00826]</a> <a href="#">[SWS_Dem_00968]</a> <a href="#">[SWS_Dem_00985]</a> <a href="#">[SWS_Dem_01078]</a>
[SRS_Diag_04077]	The DEM uses standard mechanisms provided by NVRAM-Manager	<a href="#">[SWS_Dem_00164]</a> <a href="#">[SWS_Dem_00329]</a>
[SRS_Diag_04079]	The size of a FreezeFrame shall be reported to the DCM by the DEM	<a href="#">[SWS_Dem_00074]</a> <a href="#">[SWS_Dem_00650]</a> <a href="#">[SWS_Dem_00822]</a>
[SRS_Diag_04082]	The diagnostic modules DCM and DEM shall provide standardized interfaces to support OBD services as defined in ISO15031-5 and SAE J1979	<a href="#">[SWS_Dem_00291]</a> <a href="#">[SWS_Dem_00293]</a> <a href="#">[SWS_Dem_00294]</a> <a href="#">[SWS_Dem_00296]</a> <a href="#">[SWS_Dem_00297]</a> <a href="#">[SWS_Dem_00298]</a> <a href="#">[SWS_Dem_00300]</a> <a href="#">[SWS_Dem_00301]</a> <a href="#">[SWS_Dem_00304]</a> <a href="#">[SWS_Dem_00308]</a> <a href="#">[SWS_Dem_00346]</a> <a href="#">[SWS_Dem_00347]</a> <a href="#">[SWS_Dem_00348]</a> <a href="#">[SWS_Dem_00349]</a> <a href="#">[SWS_Dem_00351]</a> <a href="#">[SWS_Dem_00354]</a> <a href="#">[SWS_Dem_00355]</a> <a href="#">[SWS_Dem_00356]</a> <a href="#">[SWS_Dem_00357]</a> <a href="#">[SWS_Dem_00358]</a> <a href="#">[SWS_Dem_00360]</a> <a href="#">[SWS_Dem_00361]</a> <a href="#">[SWS_Dem_00362]</a> <a href="#">[SWS_Dem_00377]</a>

Requirement	Description	Satisfied by
		<a href="#">[SWS_Dem_00378]</a> <a href="#">[SWS_Dem_00528]</a> <a href="#">[SWS_Dem_00590]</a> <a href="#">[SWS_Dem_00596]</a> <a href="#">[SWS_Dem_00597]</a> <a href="#">[SWS_Dem_00623]</a> <a href="#">[SWS_Dem_00703]</a> <a href="#">[SWS_Dem_00704]</a> <a href="#">[SWS_Dem_00709]</a> <a href="#">[SWS_Dem_00710]</a> <a href="#">[SWS_Dem_00711]</a> <a href="#">[SWS_Dem_00712]</a> <a href="#">[SWS_Dem_00715]</a> <a href="#">[SWS_Dem_00717]</a> <a href="#">[SWS_Dem_00718]</a> <a href="#">[SWS_Dem_00719]</a> <a href="#">[SWS_Dem_00721]</a> <a href="#">[SWS_Dem_00722]</a> <a href="#">[SWS_Dem_00723]</a> <a href="#">[SWS_Dem_00725]</a> <a href="#">[SWS_Dem_00728]</a> <a href="#">[SWS_Dem_00746]</a> <a href="#">[SWS_Dem_00748]</a> <a href="#">[SWS_Dem_00754]</a> <a href="#">[SWS_Dem_00755]</a> <a href="#">[SWS_Dem_00757]</a> <a href="#">[SWS_Dem_00762]</a> <a href="#">[SWS_Dem_00764]</a> <a href="#">[SWS_Dem_00966]</a> <a href="#">[SWS_Dem_01095]</a> <a href="#">[SWS_Dem_01096]</a> <a href="#">[SWS_Dem_01097]</a> <a href="#">[SWS_Dem_01098]</a> <a href="#">[SWS_Dem_01099]</a> <a href="#">[SWS_Dem_01137]</a> <a href="#">[SWS_Dem_01138]</a> <a href="#">[SWS_Dem_01139]</a> <a href="#">[SWS_Dem_01188]</a>
[SRS_Diag_04085]	The Default Error Tracer shall provide an interface to receive error reports	<a href="#">[SWS_Dem_00463]</a>
[SRS_Diag_04086]	Report errors shall contain a dedicated set of information	<a href="#">[SWS_Dem_00999]</a>
[SRS_Diag_04087]	The Default Error Tracer shall provide a development error report reception service	<a href="#">[SWS_Dem_00999]</a>
[SRS_Diag_04089]	Fan-out of received error reports	<a href="#">[SWS_Dem_00999]</a>
[SRS_Diag_04090]	A configurable list of error report receivers shall be	<a href="#">[SWS_Dem_00999]</a>
[SRS_Diag_04091]	Notifications about valid freeze frame data	<a href="#">[SWS_Dem_00999]</a>
[SRS_Diag_04092]	Control of event handling	<a href="#">[SWS_Dem_00514]</a> <a href="#">[SWS_Dem_00515]</a> <a href="#">[SWS_Dem_00516]</a> <a href="#">[SWS_Dem_00667]</a> <a href="#">[SWS_Dem_00668]</a> <a href="#">[SWS_Dem_00669]</a>
[SRS_Diag_04093]	Memory Overflow indication	<a href="#">[SWS_Dem_00397]</a> <a href="#">[SWS_Dem_00398]</a> <a href="#">[SWS_Dem_00399]</a> <a href="#">[SWS_Dem_00559]</a>
[SRS_Diag_04095]	The DEM module shall provide the ability to handle event specific enable and storage conditions	<a href="#">[SWS_Dem_00202]</a> <a href="#">[SWS_Dem_00446]</a> <a href="#">[SWS_Dem_00447]</a> <a href="#">[SWS_Dem_00449]</a> <a href="#">[SWS_Dem_00450]</a> <a href="#">[SWS_Dem_00453]</a> <a href="#">[SWS_Dem_00455]</a> <a href="#">[SWS_Dem_00458]</a> <a href="#">[SWS_Dem_00459]</a> <a href="#">[SWS_Dem_00543]</a> <a href="#">[SWS_Dem_00591]</a> <a href="#">[SWS_Dem_00605]</a> <a href="#">[SWS_Dem_01113]</a>

Requirement	Description	Satisfied by
[SRS_Diag_04096]	The DEM module shall support the UDS DTC status bit support & handling according to ISO14229-1	[SWS_Dem_00006] [SWS_Dem_00036] [SWS_Dem_00053] [SWS_Dem_00195] [SWS_Dem_00333] [SWS_Dem_00379] [SWS_Dem_00385] [SWS_Dem_00386] [SWS_Dem_00387] [SWS_Dem_00388] [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_00525] [SWS_Dem_00539] [SWS_Dem_01186]
[SRS_Diag_04097]	Decentralized modular diagnostic configuration of SW-Cs	[SWS_Dem_00999]
[SRS_Diag_04098]	Standard bootloader interaction	[SWS_Dem_00999]
[SRS_Diag_04099]	The DEM should forward incoming events to the DLT interface	[SWS_Dem_00517] [SWS_Dem_00633] [SWS_Dem_00634] [SWS_Dem_00635] [SWS_Dem_00636] [SWS_Dem_00637] [SWS_Dem_00992] [SWS_Dem_00993]
[SRS_Diag_04100]	The DCM shall provide an interface for DLT to transport log and trace data over a diagnostic service	[SWS_Dem_00999]
[SRS_Diag_04101]	The DET module shall forward its trace events to the DLT	[SWS_Dem_00999]
[SRS_Diag_04102]	The DEM module shall provide a chronological reporting order of the events located in the configured event memory	[SWS_Dem_00411] [SWS_Dem_00412] [SWS_Dem_00477] [SWS_Dem_00695] [SWS_Dem_00696]
[SRS_Diag_04104]	The DEM module shall support a signal based configuration of event related data (define freeze frames and extended data records)	[SWS_Dem_00469] [SWS_Dem_00779] [SWS_Dem_00821] [SWS_Dem_00995]
[SRS_Diag_04105]	Event memory management	[SWS_Dem_00580] [SWS_Dem_00607] [SWS_Dem_00783] [SWS_Dem_00784] [SWS_Dem_00785] [SWS_Dem_00786] [SWS_Dem_00829] [SWS_Dem_00922] [SWS_Dem_00923]
[SRS_Diag_04107]	Defensive behavior of the DEM module	[SWS_Dem_00339]
[SRS_Diag_04109]	The DEM module shall provide an interface to retrieve the number of event memory entries	[SWS_Dem_00651] [SWS_Dem_00652]

Requirement	Description	Satisfied by
[SRS_Diag_04110]	The DEM module shall support SAE J1939 lamp status	[SWS_Dem_00858] [SWS_Dem_00859] [SWS_Dem_00860] [SWS_Dem_00861] [SWS_Dem_00862] [SWS_Dem_00863] [SWS_Dem_00866] [SWS_Dem_00868] [SWS_Dem_00872] [SWS_Dem_00873] [SWS_Dem_00883] [SWS_Dem_00884] [SWS_Dem_00885] [SWS_Dem_00887] [SWS_Dem_00889] [SWS_Dem_00893] [SWS_Dem_00894] [SWS_Dem_00895] [SWS_Dem_00897]
[SRS_Diag_04111]	The DEM module shall support SAE J1939 Expanded- / FreezeFrame	[SWS_Dem_00877] [SWS_Dem_00899] [SWS_Dem_00900] [SWS_Dem_00901] [SWS_Dem_00902] [SWS_Dem_00903] [SWS_Dem_00904] [SWS_Dem_00905] [SWS_Dem_00906] [SWS_Dem_00907]
[SRS_Diag_04112]	The DEM module shall support DTCs according to SAE J1939	[SWS_Dem_00645] [SWS_Dem_00845] [SWS_Dem_00856] [SWS_Dem_00864] [SWS_Dem_00874]
[SRS_Diag_04113]	The DEM module shall support a set of SAE J1939 DM-messages	[SWS_Dem_00880] [SWS_Dem_00881] [SWS_Dem_00882] [SWS_Dem_00913] [SWS_Dem_00921] [SWS_Dem_01180]
[SRS_Diag_04115]	The optional parameter DTCSettingControlOption Record as part of UDS service ControlDTCSetting shall be limited to GroupOf DTC	[SWS_Dem_00080] [SWS_Dem_00626]
[SRS_Diag_04117]	The DEM shall provide a configurable behavior for the deletion of DTC	[SWS_Dem_00343] [SWS_Dem_00620] [SWS_Dem_00670] [SWS_Dem_00679] [SWS_Dem_00680] [SWS_Dem_00879]
[SRS_Diag_04118]	The DEM shall 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]
[SRS_Diag_04119]	The DCM shall handle the execution of diagnostic services according to the assigned diagnostic session	[SWS_Dem_00999]
[SRS_Diag_04120]	The DCM shall support a predefined AddressAnd LengthFormatIdentifier	[SWS_Dem_00999]
[SRS_Diag_04121]	The DCM shall provide the handling of service DynamicallyDefineData Identifier according to ISO 14229-1	[SWS_Dem_00999]

Requirement	Description	Satisfied by
[SRS_Diag_04122]	The ClearDTC command in DEM shall be usable for a Complex Device Driver	<a href="#">[SWS_Dem_00514]</a> <a href="#">[SWS_Dem_00515]</a> <a href="#">[SWS_Dem_00516]</a> <a href="#">[SWS_Dem_00569]</a> <a href="#">[SWS_Dem_00570]</a> <a href="#">[SWS_Dem_00571]</a> <a href="#">[SWS_Dem_00572]</a> <a href="#">[SWS_Dem_00573]</a> <a href="#">[SWS_Dem_00659]</a> <a href="#">[SWS_Dem_00660]</a> <a href="#">[SWS_Dem_00661]</a> <a href="#">[SWS_Dem_00662]</a> <a href="#">[SWS_Dem_00663]</a> <a href="#">[SWS_Dem_00664]</a> <a href="#">[SWS_Dem_00665]</a> <a href="#">[SWS_Dem_00667]</a> <a href="#">[SWS_Dem_00668]</a> <a href="#">[SWS_Dem_00669]</a>
[SRS_Diag_04123]	The DEM module shall support harmonized Driving-/WarmUp cycles	<a href="#">[SWS_Dem_00699]</a>
[SRS_Diag_04124]	The DEM shall be able to store the current debounce counter value non-volatile to over a power-down cycle	<a href="#">[SWS_Dem_00674]</a> <a href="#">[SWS_Dem_00675]</a> <a href="#">[SWS_Dem_00676]</a> <a href="#">[SWS_Dem_00782]</a>
[SRS_Diag_04125]	The behaviour of the event debounce counter shall be configurable	<a href="#">[SWS_Dem_00625]</a> <a href="#">[SWS_Dem_00654]</a> <a href="#">[SWS_Dem_00655]</a> <a href="#">[SWS_Dem_00656]</a> <a href="#">[SWS_Dem_00677]</a> <a href="#">[SWS_Dem_00678]</a> <a href="#">[SWS_Dem_00681]</a> <a href="#">[SWS_Dem_00682]</a> <a href="#">[SWS_Dem_00776]</a> <a href="#">[SWS_Dem_00788]</a> <a href="#">[SWS_Dem_00789]</a> <a href="#">[SWS_Dem_00790]</a> <a href="#">[SWS_Dem_00791]</a> <a href="#">[SWS_Dem_00792]</a> <a href="#">[SWS_Dem_00793]</a> <a href="#">[SWS_Dem_00794]</a> <a href="#">[SWS_Dem_00795]</a>
[SRS_Diag_04126]	Configurable suppression of events	<a href="#">[SWS_Dem_00586]</a> <a href="#">[SWS_Dem_00687]</a> <a href="#">[SWS_Dem_01080]</a> <a href="#">[SWS_Dem_01103]</a> <a href="#">[SWS_Dem_01104]</a> <a href="#">[SWS_Dem_01105]</a> <a href="#">[SWS_Dem_01106]</a> <a href="#">[SWS_Dem_01107]</a> <a href="#">[SWS_Dem_01108]</a> <a href="#">[SWS_Dem_01109]</a> <a href="#">[SWS_Dem_01110]</a> <a href="#">[SWS_Dem_01111]</a>
[SRS_Diag_04127]	Configurable record numbers and trigger options for DTCSnapshot Records and DTCExtended DataRecords	<a href="#">[SWS_Dem_00002]</a> <a href="#">[SWS_Dem_00334]</a> <a href="#">[SWS_Dem_00337]</a> <a href="#">[SWS_Dem_00461]</a> <a href="#">[SWS_Dem_00471]</a> <a href="#">[SWS_Dem_00581]</a> <a href="#">[SWS_Dem_00797]</a> <a href="#">[SWS_Dem_00798]</a> <a href="#">[SWS_Dem_00799]</a> <a href="#">[SWS_Dem_00800]</a> <a href="#">[SWS_Dem_00801]</a> <a href="#">[SWS_Dem_00802]</a> <a href="#">[SWS_Dem_00803]</a> <a href="#">[SWS_Dem_00804]</a> <a href="#">[SWS_Dem_00805]</a> <a href="#">[SWS_Dem_00806]</a> <a href="#">[SWS_Dem_00808]</a> <a href="#">[SWS_Dem_00809]</a> <a href="#">[SWS_Dem_00810]</a> <a href="#">[SWS_Dem_00811]</a> <a href="#">[SWS_Dem_00812]</a> <a href="#">[SWS_Dem_00813]</a> <a href="#">[SWS_Dem_00814]</a> <a href="#">[SWS_Dem_00815]</a> <a href="#">[SWS_Dem_00816]</a> <a href="#">[SWS_Dem_00817]</a> <a href="#">[SWS_Dem_00819]</a> <a href="#">[SWS_Dem_00820]</a> <a href="#">[SWS_Dem_00991]</a> <a href="#">[SWS_Dem_01068]</a> <a href="#">[SWS_Dem_01069]</a> <a href="#">[SWS_Dem_01070]</a> <a href="#">[SWS_Dem_01071]</a> <a href="#">[SWS_Dem_01081]</a> <a href="#">[SWS_Dem_01082]</a> <a href="#">[SWS_Dem_01083]</a>
[SRS_Diag_04128]	DEM interface to set/reset the WarningIndicator Requested bit	<a href="#">[SWS_Dem_00606]</a> <a href="#">[SWS_Dem_00831]</a> <a href="#">[SWS_Dem_00832]</a> <a href="#">[SWS_Dem_00833]</a> <a href="#">[SWS_Dem_00834]</a> <a href="#">[SWS_Dem_00835]</a> <a href="#">[SWS_Dem_00836]</a>

Requirement	Description	Satisfied by
[SRS_Diag_04129]	The DCM and DEM shall provide OBD-specific configuration capabilities	[SWS_Dem_00700] [SWS_Dem_00752]
[SRS_Diag_04130]	The DEM shall provide the capability to process a new request	[SWS_Dem_01042]
[SRS_Diag_04131]	Event management mechanisms in DEM	[SWS_Dem_01063] [SWS_Dem_01064]
[SRS_Diag_04133]	The DEM module shall support Aging	[SWS_Dem_00019] [SWS_Dem_00442] [SWS_Dem_00489] [SWS_Dem_00492] [SWS_Dem_00493] [SWS_Dem_00498] [SWS_Dem_00698] [SWS_Dem_00824] [SWS_Dem_00985] [SWS_Dem_01054] [SWS_Dem_01185]
[SRS_Diag_04134]	The DEM shall be able to store internal data to Extended Data Records and Snapshot 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]
[SRS_Diag_04135]	The DCM shall support UDS service \$38 (Request FileTransfer)	[SWS_Dem_00999]
[SRS_Diag_04136]	Support of configurable "confirmed" threshold	[SWS_Dem_00999]
[SRS_Diag_04137]	Definition of replacement failure in DEM	[SWS_Dem_01085] [SWS_Dem_01086] [SWS_Dem_01087] [SWS_Dem_01088] [SWS_Dem_01089] [SWS_Dem_01090] [SWS_Dem_01091]
[SRS_Diag_04138]	Limitatation of execution of diagnostic requests	[SWS_Dem_00999]
[SRS_Diag_04139]	The DCM and DEM shall support subfunction 0x42 of UDS service 0x19	[SWS_Dem_00999]
[SRS_Diag_04141]	The DCM and DEM shall support ISO27145 (WWH-OBD)	[SWS_Dem_00351] [SWS_Dem_00748] [SWS_Dem_01137] [SWS_Dem_01138] [SWS_Dem_01139] [SWS_Dem_01140] [SWS_Dem_01141] [SWS_Dem_01142] [SWS_Dem_01143] [SWS_Dem_01144] [SWS_Dem_01145] [SWS_Dem_01146] [SWS_Dem_01147] [SWS_Dem_01148] [SWS_Dem_01149] [SWS_Dem_01150] [SWS_Dem_01151] [SWS_Dem_01152] [SWS_Dem_01153] [SWS_Dem_01154] [SWS_Dem_01155] [SWS_Dem_01156] [SWS_Dem_01157] [SWS_Dem_01158]

Requirement	Description	Satisfied by
		[SWS_Dem_01159] [SWS_Dem_01160] [SWS_Dem_01161] [SWS_Dem_01162] [SWS_Dem_01163] [SWS_Dem_01164] [SWS_Dem_01165] [SWS_Dem_01166] [SWS_Dem_01170] [SWS_Dem_01171] [SWS_Dem_01172] [SWS_Dem_01173] [SWS_Dem_01174] [SWS_Dem_01175] [SWS_Dem_01176] [SWS_Dem_01177] [SWS_Dem_01178] [SWS_Dem_01179] [SWS_Dem_01181] [SWS_Dem_01182] [SWS_Dem_01183]
[SRS_Diag_04142]	The DEM module shall support component dependencies	[SWS_Dem_00999] [SWS_Dem_01114] [SWS_Dem_01115] [SWS_Dem_01116] [SWS_Dem_01117] [SWS_Dem_01118] [SWS_Dem_01119] [SWS_Dem_01120] [SWS_Dem_01121] [SWS_Dem_01122] [SWS_Dem_01123] [SWS_Dem_01124] [SWS_Dem_01125] [SWS_Dem_01126] [SWS_Dem_01127] [SWS_Dem_01128] [SWS_Dem_01129] [SWS_Dem_01130] [SWS_Dem_01131] [SWS_Dem_01132] [SWS_Dem_01133] [SWS_Dem_01134] [SWS_Dem_01135] [SWS_Dem_01136] [SWS_Dem_01195] [SWS_Dem_01197]
[SRS_Diag_04143]	The Default Error Tracer shall provide an interface to receive runtime error reports	[SWS_Dem_00999]
[SRS_Diag_04144]	The Default Error Tracer shall provide an interface to receive transient fault reports	[SWS_Dem_00999]
[SRS_Diag_04145]	The Default Error Tracer shall forward received runtime error reports to configured integrator code	[SWS_Dem_00999]
[SRS_Diag_04146]	The Default Error Tracer shall forward received transient fault reports to configured integrator code	[SWS_Dem_00999]
[SRS_Rte_00182]	Self Scaling Signals at Port Interfaces	[SWS_Dem_00850]
[SWS_BSW_00050]	Check parameters passed to Initialization functions	[SWS_Dem_00173]
[SWS_BSW_00212]	NULL pointer checking	[SWS_Dem_00173]

## 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.

Figure 7.1 shows the Dem configuration. `DemGeneral` contains the global part of the configuration and `DemConfigSet` contains the multiple configuration part.

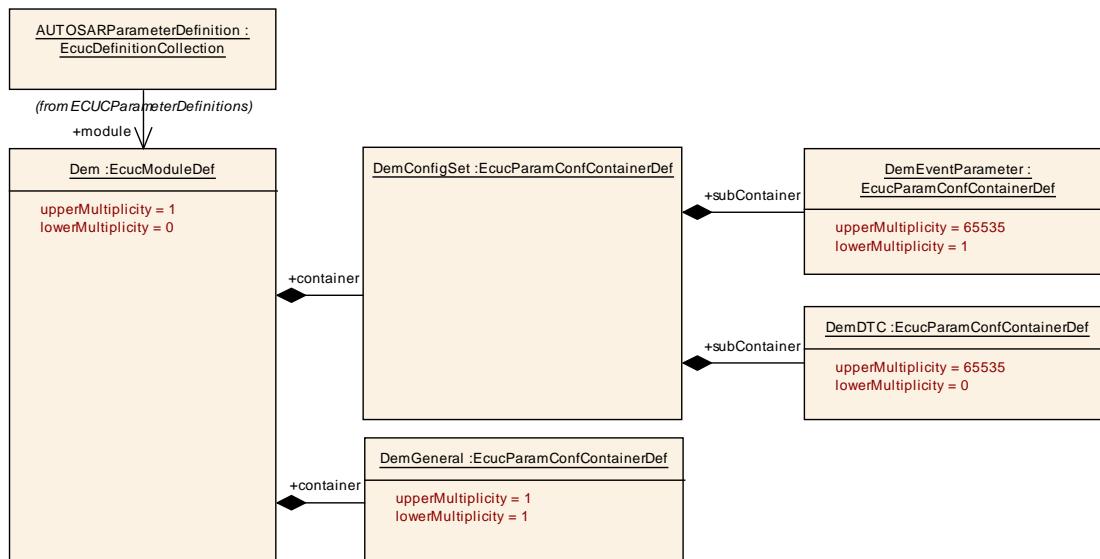


Figure 7.1: Top-level view of Dem configuration

### 7.1 Startup behavior

**[SWS\_Dem\_00169]** [ The Dem module shall distinguish between a pre-initialization mode and a full-initialized mode (operation mode). ] ([SRS\\_BSW\\_00406](#))

**[SWS\_Dem\_00180]** [ The function `Dem_Prelit` (refer to chapter 8.3.2.1) shall initialize the internal states of the Dem module necessary to process events and reset `debounce counters` reported by BSW modules by using `Dem_ReportErrorStatus` (refer to chapter 7.7) and `Dem_ResetEventDebounceStatus` (refer to chapter 7.6.3). ]()

The function `Dem_Prelit` is called by the ECU State Manager during the startup phase of the ECU before the NVRAM Manager is initialized. For all operation cycles having a autostart functionality by configuration parameter `DemOperationCycleAutostart`, the BSW modules can start reporting of related events via `Dem_ReportErrorStatus` (refer to [SWS\_Dem\_00854], [SWS\_Dem\_00851], [SWS\_Dem\_00167]).

The function `Dem_Init` (refer to [SWS\_Dem\_00340]and chapter 8.3.2.2) 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]** [ The Dem module shall provide the interface [InitMonitorForEvent](#) to trigger the initialization of a diagnostic monitor (refer also to chapter [7.6.3](#) and [\[SWS\\_Dem\\_00573\]](#)). ]()

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](#) (in [DemEventParameter](#)) is used to specify the related port or c-callback per event.

**[SWS\_Dem\_00679]** [ The API [Dem\\_SetOperationCycleState](#) (refer to chapter [7.6.8](#)) shall trigger the callback function [InitMonitorForEvent](#) of the related event(s) in case of starting or restarting the operation cycle of the event(s). The [InitMonitorReason](#) parameter shall be set to [DEM\\_INIT\\_MONITOR\\_RESTART](#). ]([SRS\\_Diag\\_04117](#))

**[SWS\_Dem\_00680]** [ Any [Dem\\_<...>ClearDTC](#) API (refer to chapter [7.6.2.2](#)) 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](#). ]([SRS\\_Diag\\_04117](#))

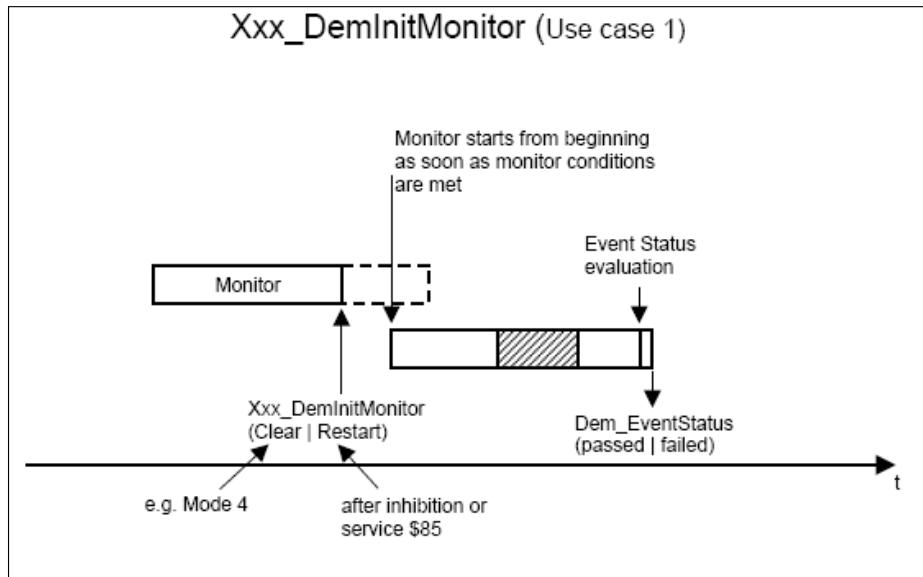
**[SWS\_Dem\_00681]** [ The API [Dem\\_SetEnableCondition](#) (refer to chapter [7.6.6](#)) shall trigger the callback function [InitMonitorForEvent](#) of the related event(s) in case an enable condition of the event(s) is changed to fulfilled and thus all enable conditions of the event(s) are fulfilled. The [InitMonitorReason](#) parameter shall be set to [DEM\\_INIT\\_MONITOR\\_REENABLED](#). ]([SRS\\_Diag\\_04125](#))

**[SWS\_Dem\_00682]** [ The API [Dem\\_DcmEnableDTCSetting](#) (refer to chapter [7.10.2.4](#)) 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](#). ]([SRS\\_Diag\\_04125](#))

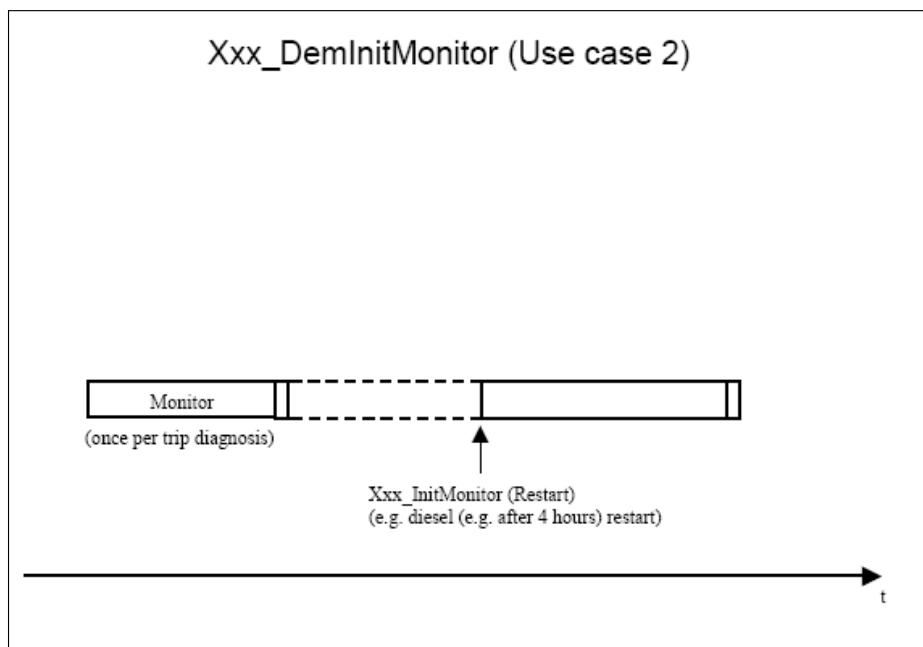
**[SWS\_Dem\_01113]** [ The API [Dem\\_SetStorageCondition](#) (refer to chapter [7.6.6](#)) shall trigger the callback function [InitMonitorForEvent](#) of the related event(s) in case a storage condition of the event(s) is changed to fulfilled and thus all storage conditions of the event(s) are fulfilled. 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`.  
`](SRS_Diag_04095)`

The following figures show two examples:



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



**Figure 7.3: 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[13] request).

**[SWS\_Dem\_01046]** [ In case multiple triggers for `DemInitMonitorForEvent` 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` (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]** [ The Dem module shall represent each Diagnostic Event by an EventId and the related EventName. ]([SRS\\_Diag\\_04061](#), [SRS\\_Diag\\_04063](#))

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]** [ The EventId and the related EventName shall be unique per Dem module represented by the ECU configuration (refer to [\[SWS\\_Dem\\_00126\]](#)). ]([SRS\\_Diag\\_04063](#))

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

The Dem module may use an internal event status. However, when requested by the Dcm, the extended event status will be 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.

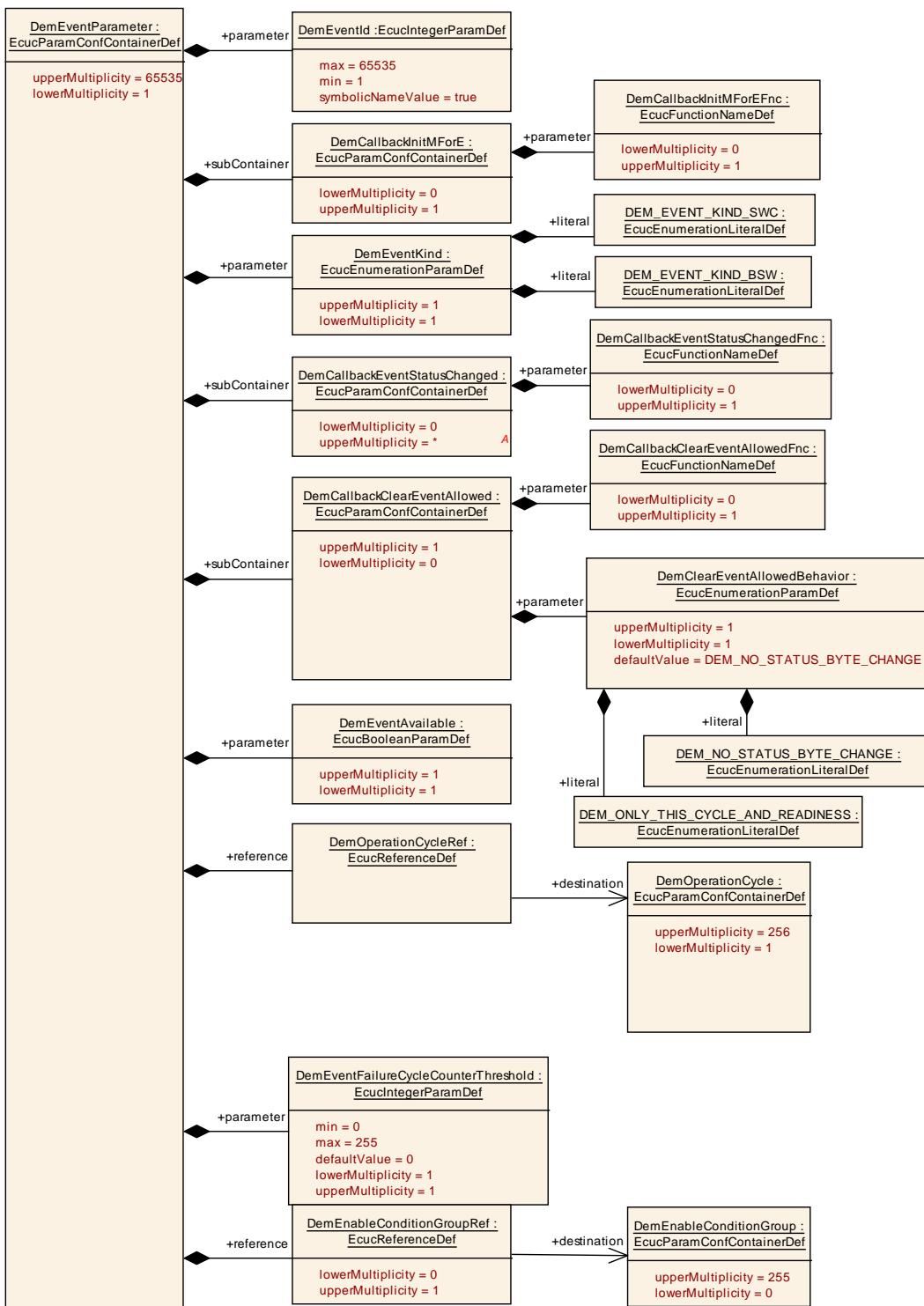
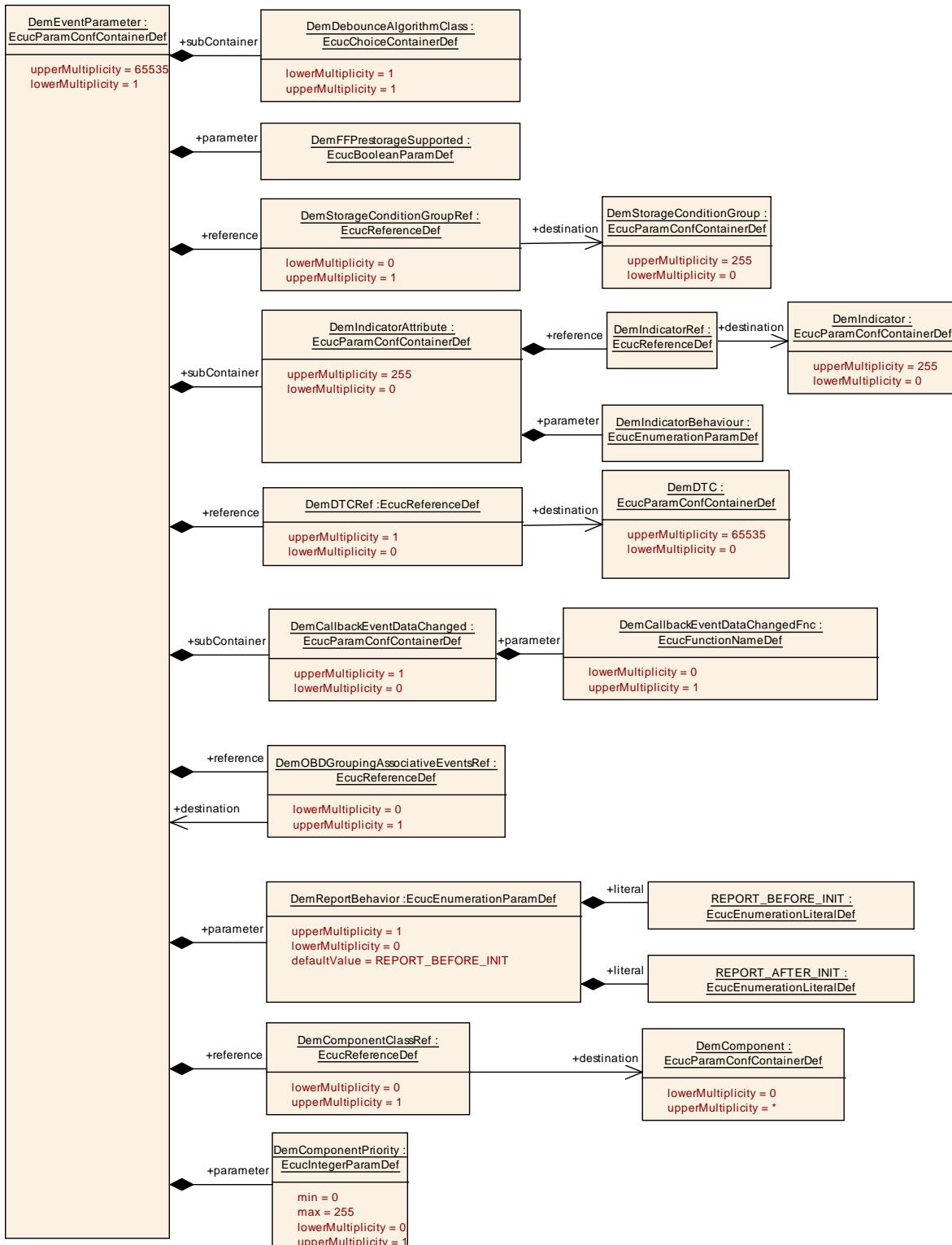


Figure 7.4: Event parameter configuration (part 1)


**Figure 7.5: 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]** [ Each supported event shall have a priority assigned to it (refer to parameter [DemDTCPriority](#) in [DemDTCAttributes](#)). ]([SRS\\_Diag\\_04118](#), [SRS\\_Diag\\_04071](#))

**[SWS\_Dem\_00383]** [ A priority value of 1 shall be the highest priority. Larger priority value shall define lower importance. ]([SRS\\_Diag\\_04118](#), [SRS\\_Diag\\_04071](#))

### 7.3.2 Event occurrence

**[SWS\_Dem\_00011]** [ The Dem module shall provide an occurrence counter per [event memory entry](#). ]([SRS\\_Diag\\_04067](#))

**[SWS\_Dem\_00523]** [ The Dem module shall initialize the occurrence counter with the value one if the related event is entered in the respective [event memory](#). ]([SRS\\_Diag\\_04067](#))

**[SWS\_Dem\_00524]** [ If the configuration parameter [DemOccurrenceCounterProcessing](#) (refer to [DemGeneral](#)) is [DEM\\_PROCESS\\_OCCCTR\\_TF](#) , the Dem module shall increment the occurrence counter by one, triggered by each UDS DTC status bit 0 (TestFailed) transition from 0 to 1, if the related event is already stored in the [event memory](#). ]([SRS\\_Diag\\_04067](#))

**[SWS\_Dem\_00580]** [ If the configuration parameter [DemOccurrenceCounterProcessing](#) (refer to [DemGeneral](#)) is [DEM\\_PROCESS\\_OCCCTR\\_CDTC](#) , the Dem module shall increment the occurrence counter by one, triggered by each UDS DTC status bit 0 (TestFailed) transition from 0 to 1, if the related event is already stored in the [event memory](#) and the UDS DTC Status bit 3 (ConfirmedDTC) is equal to 1 (refer to chapter 0). ]([SRS\\_Diag\\_04105](#))

**[SWS\_Dem\_00625]** [ The Dem module shall not increment the event-specific occurrence counter anymore, if it has reached its maximum value (255, refer to [\[SWS\\_Dem\\_00471\]](#)). ]([SRS\\_Diag\\_04125](#))

### 7.3.3 Event kind

There are two different types of events:

- BSW-related events (reported via C-API - [Dem\\_ReportErrorStatus](#))
- 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 [7.7](#)).

### 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 [7.6.7](#)).

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 [7.8.5.8](#)). 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.

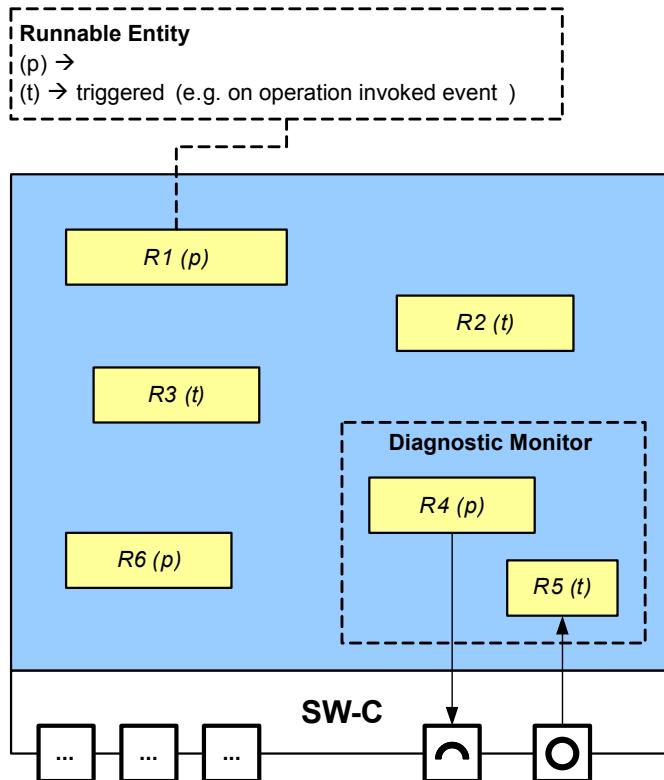
The definition and use of the different memory types is OEM specific.

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, permanent and mirror memory).

**[constr\_6104]** [ Limitations on [DemMemoryDestinationRef](#) - If [DemMirrorMemory](#) is configured as [DemMemoryDestinationRef](#), another [DemMemoryDestinationRef](#) on the same event of either [DemPrimaryMemory](#) or [DemUserDefinedMemory](#) shall be configured as a prerequisite. The same event shall not be configured two destinations if one is not [DemMirrorMemory](#). ]()

### 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 7.6: 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 7.6.3), 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 [DemComponents](#) 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. ] ([SRS\\_Diag\\_04142](#))

**[SWS\_Dem\_01127]** [ Whenever an event reports FAILED, it shall be considered as CONSECUTIVE FAULT, if any parent [DemComponent](#) is FAILED. ] ([SRS\\_Diag\\_04142](#))

**[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).] [\(SRS\\_Diag\\_04142\)](#)

**[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. ] [\(SRS\\_Diag\\_04142\)](#)

**[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). ] [\(SRS\\_Diag\\_04142\)](#)

**[SWS\_Dem\_01131]** [ The [Dem](#) shall provide the interface [Dem\\_GetComponentFailed](#), which allows querying a [DemComponents](#) FAILED status. ] [\(SRS\\_Diag\\_04142\)](#)

**[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 [FiM\\_DemTriggerOnComponentStatus](#). ] [\(SRS\\_Diag\\_04142\)](#)

**[SWS\_Dem\_01133]** [ The [Dem](#) shall allow defining a callback per [DemComponent](#) ([DemComponent FailedCallbackFnc](#); [<Module>\\_DemTriggerOnComponentStatus](#)), which shall be called on a change of the components FAILED status. ] [\(SRS\\_Diag\\_04142\)](#)

### 7.3.7 Component availability

**[SWS\_Dem\_01134]** [ The [Dem](#) shall support the availability of [DemComponents](#). 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). ] [\(SRS\\_Diag\\_04142\)](#)

**[SWS\_Dem\_01135]** [ The interface [Dem\\_SetComponentAvailable](#) shall be available to set the availability state of a component. ] [\(SRS\\_Diag\\_04142\)](#)

**[SWS\_Dem\_01136]** [ With setting a [DemComponent](#) to not available, all assigned events shall also be set as not available. ] [\(SRS\\_Diag\\_04142\)](#)

## 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 [7.10.2](#)).

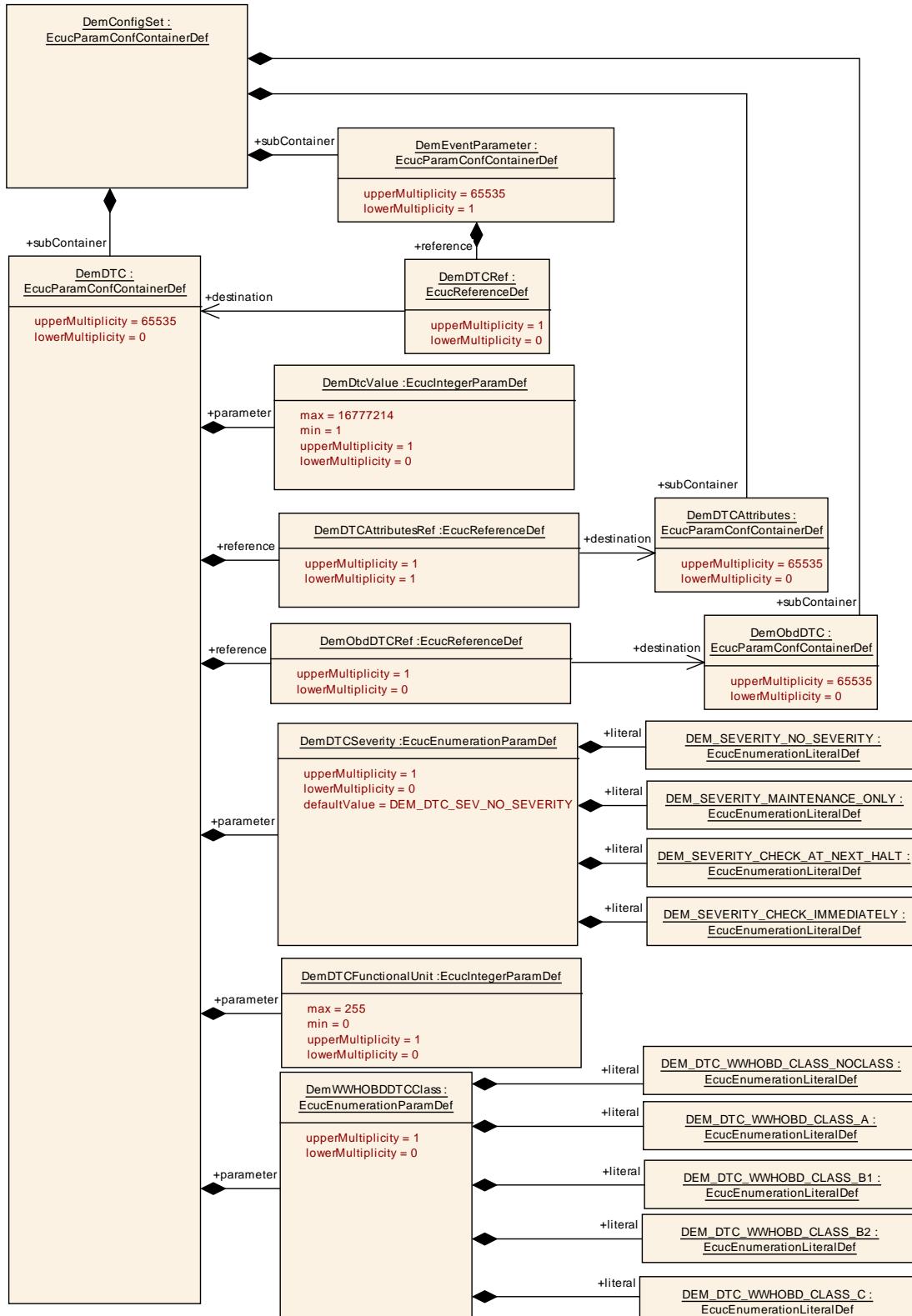
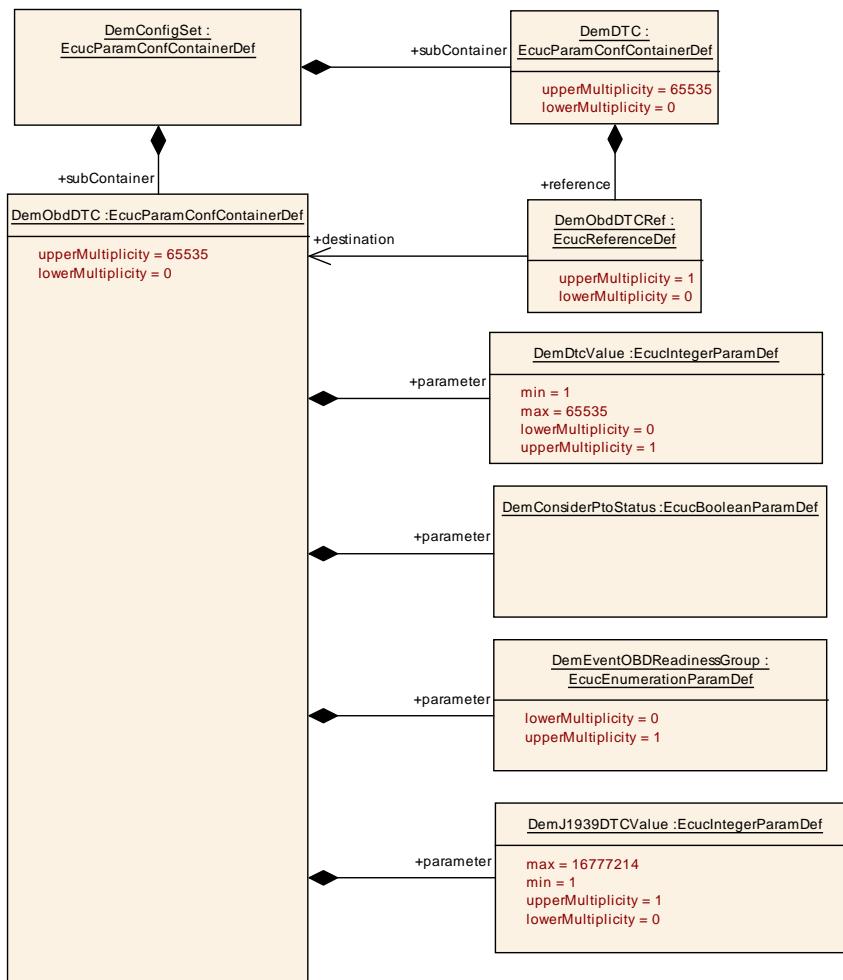


Figure 7.7: DTC configuration



**Figure 7.8: OBD-DTC configuration**

#### 7.4.1 DTC kind

There are two different kinds of DTCs:

- non OBD-relevant DTCs (UDS DTCs)
- OBD-relevant DTCs

This kind is derived implicitly from the [DemDTCAtributes](#) configuration. If the parameter [DemObdDTC](#) exists, the DTC and all related events are OBD-relevant.

Note: The DTC kind (refer to chapter [8.2.1.10](#)) is relevant for the selection of several DTCs used by the diagnostic service [ReadDTCInformation](#) and some OBD services (0x19/\$03/\$07/\$0A, refer to [Dem\\_DcmSetDTCFilter](#)), as well as for the diagnostic service [ControlDTCSetting](#) (0x85, refer to [Dem\\_DcmDisableDTCSetting](#) and [Dem\\_DcmEnableDTCSetting](#)).

### 7.4.2 DTC format

**[SWS\_Dem\_00013]** [ The Dem module shall support DTC formats for [DemDTC](#) according to:

- ISO-14229-1[1]
- SAE J2012 OBD DTC (aka 2-byte DTC) [14]
- SAE J1939-73[15]
- ISO 11992-4[16]
- SAE J2012 WWH-OBD DTC (aka 3-byte DTC) [14]

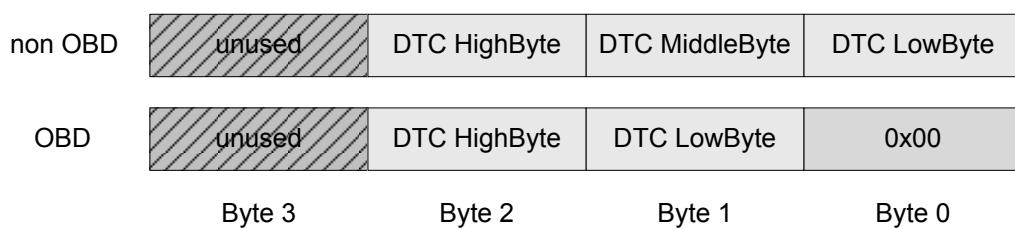
] ([SRS\\_Diag\\_04000](#))

The configuration parameter [DemTypeOfDTCSupported](#) (refer to [DemGeneral](#)) is used to select one of the supported DTC formats defined in [\[SWS\\_Dem\\_00013\]](#) of the ECU (refer to [Dem\\_DcmGetTranslationType](#)). This is required to determine the DTC format value to be reported for ISO-14229-1[1] service Read DTC Information (0x19).

**[SWS\_Dem\_00645]** [ 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](#)). ] ([SRS\\_Diag\\_04010](#), [SRS\\_Diag\\_04000](#), [SRS\\_Diag\\_04001](#), [SRS\\_Diag\\_04112](#))

A DTC can have any combination of the three formats (UDS, OBD, and J1939), i.e. one, two, or three formats at the same time. The Dem will therefore handle three DTC value lists internally. The reported format depends on the [Dem\\_DTCFormatType](#) (refer to chapter [8.2.1.11](#)) or is defined by the context of the related API.

**[SWS\_Dem\_00277]** [ 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 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. ] ([SRS\\_Diag\\_04010](#))



**Figure 7.9: DTC Byte Order**

**[SWS\_Dem\_00921]** [ The [SPN](#) of the J1939DTC shall be represented as Intel for all 19 bits in conjunction with the SPN Conversion Method Version 4 (CM = 0). ] ([SRS\\_Diag\\_04113](#)) Refer chapter [5.7.1.11](#) SPN Conversion Method in [15] for details.

**[SWS\_Dem\_01180]** [ 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) [14]. ] ([SRS\\_Diag\\_04113](#))

**[SWS\_Dem\_00269]** [ The function [Dem\\_GetDTCOfEvent](#) shall get the [DTC](#) which is mapped to EventId by the [Dem](#) configuration. ] ()

#### 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 [1] - 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. ] ([SRS\\_Diag\\_04065](#)) Note: A group with the global boundaries 0x000100 and 0xffffe00 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. ] ([SRS\\_Diag\\_04065](#))

Note: DTC groups are relevant for the diagnostic service [ClearDiagnosticInformation](#) (0x14, refer to [Dem\\_DcmClearDTC](#)), as well as for the diagnostic service [ControlDTCSetting](#) (0x85, refer to [Dem\\_DcmDisableDTCSetting](#) and [Dem\\_DcmEnableDTCSetting](#)).

The following DTC groups are provided:

- ‘all DTCs’ DTC group (mandatory, fixed value = 0xFFFFFFF)
- Emission related DTCs for WWH-OBD (0xFFFF33)

Note, that the DTC group ‘all DTCs’ will not be configured in [DemGroupOfDTC](#), because it has always to be provided by the Dem module.

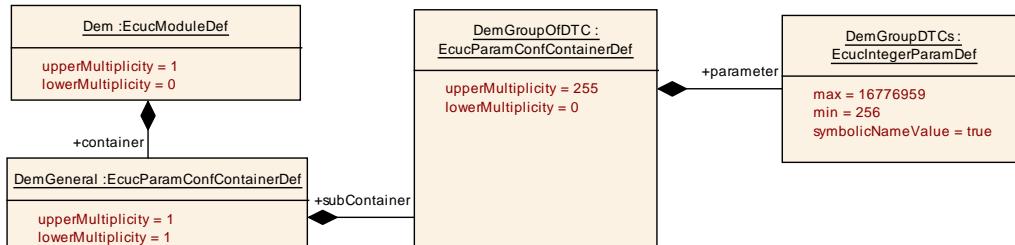


Figure 7.10: DTC group configuration

Taking into consideration that 0xFFFFFFF 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

**[SWS\_Dem\_00033]** [ The Dem module shall provide a severity value per [DTC](#) (regarding the importance of the specific events) according to ISO 14229-1, Annex D.3 “DTCSeverityMask and DTCSeverity bit definitions” [1] (refer to [Dem\\_DcmGetSeverityOfDTC](#) and [Dem\\_DcmGetNextFilteredDTCAndSeverity](#)), only if configured for at least one [DTC](#). ] ([SRS\\_Diag\\_04071](#))

The severity is only available for ISO-14229-1[1] DTCs. It is configurable per DTC optionally (refer to [DemDTCSeverity](#) in [DemDTCAttributes](#)).

Note: ISO-14229-1[1], Annex D defines the following severity levels: no severity available, Maintenance, Check at next Halt, Check immediately.

The function [Dem\\_DcmSetDTCFilter](#) (refer to [\[SWS\\_Dem\\_00057\]](#)) allows filtering for DTCs with severity information.

#### 7.4.5 Functional unit

**[SWS\_Dem\_00593]** [ The Dem module shall provide a functional unit value per DTC (refer to [Dem\\_DcmGetFunctionalUnitOfDTC](#)), only if configured for at least one DTC. ] ([SRS\\_Diag\\_04057](#))

The functional unit value is configurable per DTC (refer to [DemDTCFunctionalUnit](#) in [DemDTCAttributes](#)).

Note: The functional unit values are required by the Dcm.

#### 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 [DemDTCAtributes](#)) 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\_00586]** [ The function [Dem\\_SetDTCSuppression](#) is available if the configuration parameter [DemSuppressionSupport](#) is set to [DEM\\_DTC\\_SUPPRESSION](#). ] ([SRS\\_Diag\\_04126](#))

**[SWS\_Dem\_00915]** [ The [Dem](#) shall suppress a [DTC](#) if all related events of this [DTC](#) are not available (refer to 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]** [ A suppressed DTC shall not be visible for the following DCM query-functions, in that case the following APIs shall return the given return code:

- [Dem\\_DcmGetStatusOfDTC](#)  
return [DEM\\_STATUS\\_WRONG\\_DTC](#)
- [Dem\\_DcmGetSeverityOfDTC](#)  
return [DEM\\_GET\\_SEVERITYOFDTC\\_WRONG\\_DTC](#)

- `Dem_DcmGetFunctionalUnitOfDTC`  
return `DEM_GET_FUNCTIONALUNITOFTC_WRONG_DTC`
- `Dem_DcmGetFreezeFrameDataByDTC`  
return `DEM_GET_FFDATABYDTC_WRONG_DTC`
- `Dem_DcmGetSizeOfFreezeFrameByDTC`  
return `DEM_GETSIZEBYDTC_WRONG_DTC`
- `Dem_DcmGetExtendedDataRecordByDTC`  
return `DEM_RECORD_WRONG_DTC`
- `Dem_DcmGetSizeOfExtendedDataRecordByDTC`  
return `DEM_GETSIZEBYDTC_WRONG_DTC`
- `Dem_DcmClearDTC`  
return `DEM_CLEAR_WRONG_DTC`

]()

**[SWS\_Dem\_01101]** ┌ 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_DcmSetDTCFilter`
- `Dem_DcmGetNextFilteredDTC`
- `Dem_DcmGetNextFilteredDTCAndFDC`
- `Dem_DcmGetNextFilteredDTCAndSeverity`
- `Dem_DcmGetNextFilteredRecord`
- `Dem_DcmGetDTCByOccurrenceTime`
- `Dem_DcmGetDTCOfOBDFreezeFrame`
- `Dem_DcmReadDataOfOBDFreezeFrame`
- `Dem_DcmGetInfoTypeValue08`
- `Dem_DcmGetInfoTypeValue0B`
- `Dem_DcmReadDataOfPID01`
- `Dem_DcmGetFreezeFrameDataByDTC`
- `Dem_DcmGetSizeOfFreezeFrameByDTC`
- `Dem_DcmGetExtendedDataRecordByDTC`
- `Dem_DcmGetSizeOfExtendedDataRecordByDTC`
- `Dem_J1939DcmClearDTC`
- `Dem_J1939DcmFirstDTCwithLampStatus`

- [Dem\\_J1939DcmGetNextDTCwithLampStatus](#)
- [Dem\\_J1939DcmGetNextFreezeFrame](#)
- [Dem\\_J1939DcmSetDTCFilter](#)
- [Dem\\_J1939DcmGetNumberOfFilteredDTC](#)
- [Dem\\_J1939DcmGetNextFilteredDTC](#)
- [Dem\\_J1939DcmSetRatioFilter](#)
- [Dem\\_J1939DcmGetNextFilteredRatio](#)
- [Dem\\_J1939DcmReadDiagnosticReadiness1](#)
- [Dem\\_J1939DcmReadDiagnosticReadiness2](#)
- [Dem\\_J1939DcmReadDiagnosticReadiness3](#)
- [Dem\\_DcmGetDTRData](#)

]()

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

**[SWS\_Dem\_01102]** [ [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.

#### 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 [DemEventAvailable](#)

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

=> Interface [Dem\\_SetEventAvailable](#)

**[SWS\_Dem\_00687]** [ The Dem shall provide the functionality to treat events as if they were not configured as [DemEventParameter](#). The configuration parameter [DemAvailabilitySupport](#) is used to enable the functionality (including API [Dem\\_SetEventAvailable](#)). ]([SRS\\_Diag\\_04126](#))

**[SWS\_Dem\_01103]** [ Unavailable events shall not be considered for computation of service \$01 [PID \\$41](#). ]([SRS\\_Diag\\_04126](#))

**[SWS\_Dem\_01104]** [ [IUMPR](#) ratios referring to an unavailable event shall neither be computed nor reported. ]([SRS\\_Diag\\_04126](#))

**[SWS\_Dem\_01105]** [ [DTRs](#) referring to the event shall neither be computed nor reported ([Dem\\_SetDTR](#)). ]([SRS\\_Diag\\_04126](#))

**[SWS\_Dem\_01106]** [ 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). ]([SRS\\_Diag\\_04126](#))

**[SWS\_Dem\_01107]** [ The value of the configuration parameter [DemEventAvailable](#) shall be used as initial value for availability of an event and may be changed dynamically via API [Dem\\_SetEventAvailable](#). ]([SRS\\_Diag\\_04126](#))

Note: [Dem\\_SetEventAvailable](#) is available before [Dem\\_Init](#) to “pre-configure” the Dem in advance of the application start.

Note: If an events availability is changed by [Dem\\_SetEventAvailable](#) and the ECU is powered down and up again. The desired status has to be set again by using [Dem\\_SetEventAvailable](#).

**[SWS\_Dem\_01108]** [ 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\\_GetEventStatus](#) shall return E\_NOT\_OK). The following APIs are affected:

- [Dem\\_ReportErrorStatus](#)
- [Dem\\_SetEventStatus](#)
- [Dem\\_GetEventStatus](#)
- [Dem\\_ResetEventDebounceStatus](#)
- [Dem\\_ResetEventStatus](#)
- [Dem\\_PrestoreFreezeFrame](#)

- [Dem\\_ClearPrestoredFreezeFrame](#)
- [Dem\\_GetEventFailed](#)
- [Dem\\_GetEventTested](#)
- [Dem\\_GetDebouncingOfEvent](#)
- [Dem\\_GetDTCOfEvent](#)
- [Dem\\_GetFaultDetectionCounter](#)
- [Dem\\_GetEventFreezeFrameDataEx](#)
- [Dem\\_GetEventExtendedDataRecordEx](#)
- [Dem\\_ClearDTC](#)
- [Dem\\_DcmGetDTRData](#)
- [Dem\\_ReplUMPRFaultDetect](#)
- [Dem\\_ReplUMPRDenLock](#)
- [Dem\\_ReplUMPRDenRelease](#)
- [Dem\\_SetWIRStatus](#)
- [Dem\\_DltGetMostRecentFreezeFrameRecordData](#)
- [Dem\\_DltGetAllExtendedDataRecords](#)

]([SRS\\_Diag\\_04126](#))

Note: The availability of an event may also impact the suppression of a [DTC](#) (see [[SWS\\_Dem\\_00915](#)]).

**[SWS\_Dem\_01109]** [ The API [Dem\\_SetEventAvailable](#) shall reject the request and return E\_NOT\_OK, if an event memory entry exists already or testfailed is set to 1.  
]([SRS\\_Diag\\_04126](#))

**[SWS\_Dem\_01110]** [ In case an event changes its state to unavailable (availability==false) (via call of [Dem\\_SetEventAvailable](#)) the event status shall be set to 0x00.  
]([SRS\\_Diag\\_04126](#))

Note: Due to event status change the EventStatusChanged callbacks are called to properly provide the status to the users (e.g. Fim\_DemTriggerOnEventStatus to trigger recalculation of inhibitions).

**[SWS\_Dem\_01111]** [ In case an event changes its state to available (availability==true) (via call of [Dem\\_SetEventAvailable](#)) the event status shall be set to 0x50.  
]([SRS\\_Diag\\_04126](#))

Note: Due to event status change the EventStatusChanged callbacks are called to properly provide the status to the users (e.g. Fim\_DemTriggerOnEventStatus to trigger recalculation of inhibitions).

**[SWS\_Dem\_01112]** [ If development error detection is enabled and any instance calls any `Dem` API, excluding `Dem_ReportErrorStatus`, `Dem_ResetEventStatus`, `Dem_SetEventAvailable` and `Dem_GetVersionInfo`, before the `Dem` was fully initialized, the Dem module shall set the error code `DEM_E_UNINIT`. ] ([SRS\\_BSW\\_00406](#))

## 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 `DemEvents` and Components and finally allows to define the relation in-between the components.

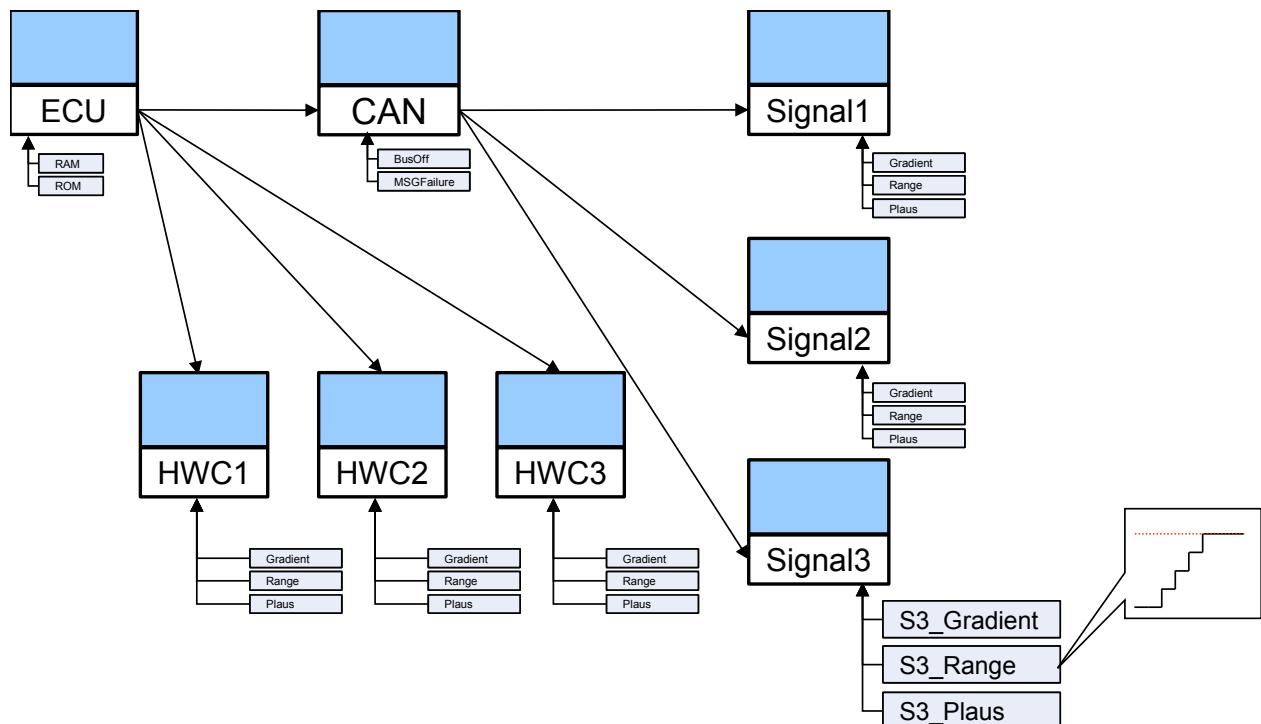


Figure 7.11: Event dependencies

### 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. ]([SRS\\_Diag\\_04142](#))

**[SWS\_Dem\_01119]** [ Each [DemComponent](#) shall have a boolean status FAILED, which represents the information whether the component has errors or not. ]([SRS\\_Diag\\_04142](#))

**[SWS\_Dem\_01120]** [ The [Dem](#) shall allow defining relations between [DemComponents](#). 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. ]([SRS\\_Diag\\_04142](#))

Example: a -> b -> c -> d

a and b are parents of c

c and d are depending on b

**[constr\_6106]** [ Only [directed acyclic graph](#) structures are supported for the dependencies of [DemComponents](#). ]()

**[SWS\_Dem\_01121]** [ The Dem shall allow to assign events to DemComponents. ]([SRS\\_Diag\\_04142](#))

**[constr\_6107]** [ 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](#). ]([SRS\\_Diag\\_04142](#))

**[SWS\_Dem\_01123]** [ On reporting an error (eventstatus FAILED) for an event, the assigned [DemComponent](#) shall be set as FAILED. ]([SRS\\_Diag\\_04142](#))

**[SWS\_Dem\_01124]** [ On setting a [DemComponent](#) to FAILED, all dependent components shall be set as FAILED. ]([SRS\\_Diag\\_04142](#))

**[SWS\_Dem\_01125]** [ A [DemComponent](#) shall be set to NOT FAILED, when all assigned events are NOT FAILED and all parent components are NOT FAILED. ]([SRS\\_Diag\\_04142](#))

## 7.6 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 extended event status and the event related data.

**[SWS\_Dem\_00010]** [ The Dem module shall support the primary [event memory](#). ]([SRS\\_Diag\\_04066](#))

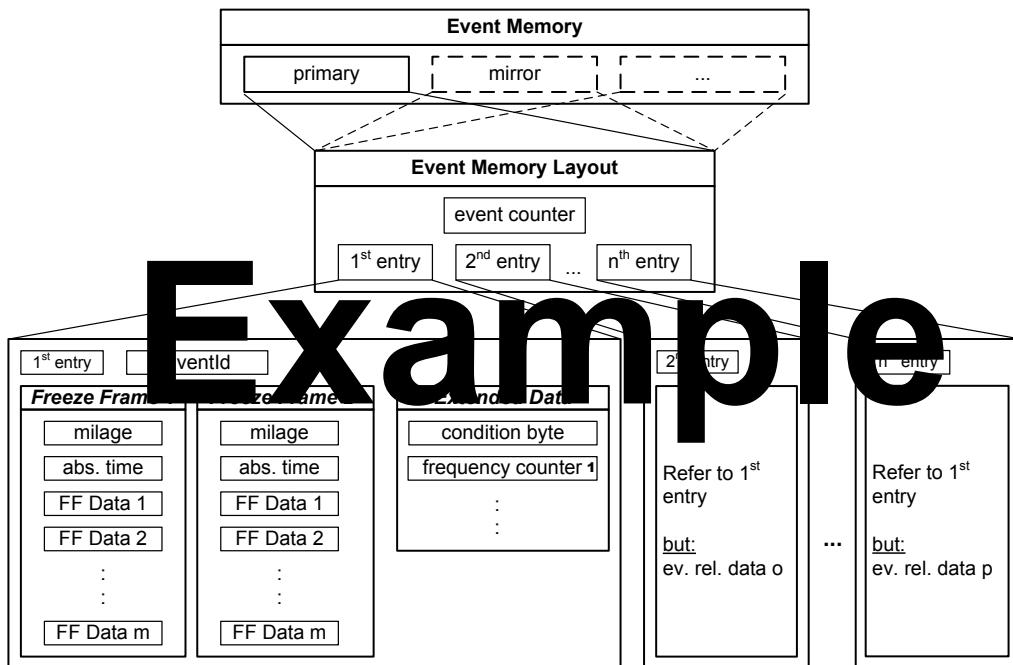
**[SWS\_Dem\_00548]** [ If configured (refer to [\[SWS\\_Dem\\_00162\]](#)) the Dem module shall support the additional [event memories](#) (user defined, mirror, permanent). ]([SRS\\_Diag\\_04066](#))

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 DemMaxNumberEventEntry<...> in [DemGeneral](#)). ]([SRS\\_Diag\\_04066](#))

**[SWS\_Dem\_00329]** [ For storing to non-volatile memory the Dem shall use the NVRAM Manager (refer to chapter [7.10.5](#)). ]([SRS\\_Diag\\_04077](#))

The following figure shows an example of a logical Dem [event memory](#) layout.



\* only at the second appearance, up to the maximal number of freeze frames  
1 frequency (occurrence) counter only increments

**Figure 7.12: 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 [7.6.2](#)).

### 7.6.1 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]** [ 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 chapter 8.3.3.3 and [SWS\_Dem\_00107]). ](SRS\_Diag\_04067)

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 11).

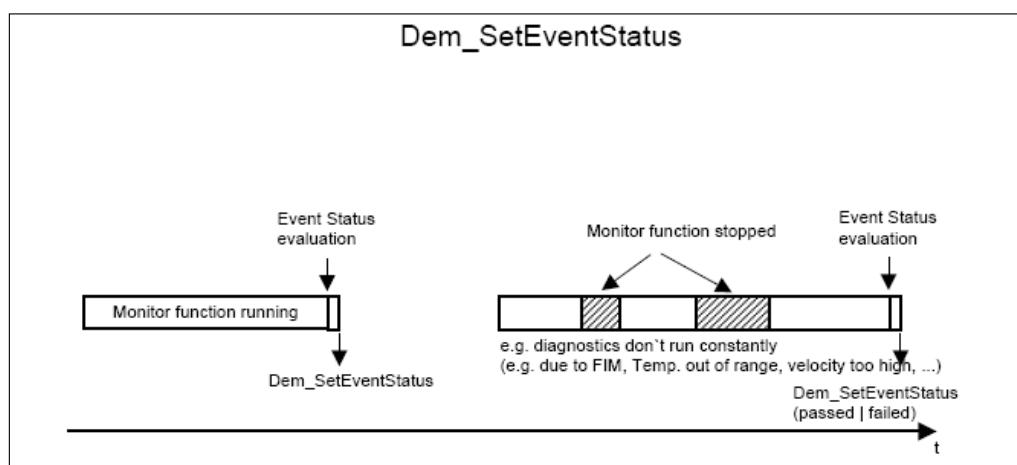


Figure 7.13: Example for using `Dem_SetEventStatus`

**[SWS\_Dem\_00331]** [ The Dem module shall provide the capability to reset the failed status of an event without reporting a passed result (refer to chapter 8.3.3.5). ](SRS\_Diag\_04030)

Note: `Dem_ResetEventStatus` does not mean an event status report (like done by `Dem_SetEventStatus`). After the call, the event 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]** [ The function `Dem_ResetEventStatus` shall set the UDS DTC status bit 0 (TestFailed) to 0 and reset the Dem-internal debounce algorithm to initial values if configured. ](SRS\_Diag\_04067)

Note: The function `Dem_ResetEventStatus` does not change the UDS DTC status bit 6 (TestNotCompletedThisOperationCycle) and does not clear the pre-stored freeze frame.

**[SWS\_Dem\_00638]** [ The function [Dem\\_ResetEventStatus](#) shall return E\_NOT\_OK, if the event was already tested this operation cycle (UDS DTC status bit 6 - TestNotCompletedThisOperationCycle is set to 0). ]([SRS\\_Diag\\_04067](#))

**[SWS\_Dem\_00051]** [ The Dem module shall provide the capability to retrieve the current UDS DTC status byte of a specific event (refer to chapter [8.3.3.13](#)). ]([SRS\\_Diag\\_04067](#))

Note: The function [Dem\\_GetEventStatus](#) is provided to be used by SW-Cs or other BSW modules (e.g. FiM) on event-level. The Dcm uses the function [Dem\\_DcmGetStatusOfDTC](#) on DTC-level instead.

**[SWS\_Dem\_00333]** [ The Dem module shall be able to provide the current event failed status (refer to chapter [8.3.3.14](#)) and the current event tested status (refer to chapter [8.3.3.15](#)). ]([SRS\\_Diag\\_04067](#), [SRS\\_Diag\\_04096](#))

**[SWS\_Dem\_00052]** [ The function [Dem\\_GetEventFailed](#) shall report the UDS DTC status bit 0 (TestFailed) of the requested diagnostic event. ]([SRS\\_Diag\\_04067](#))

**[SWS\_Dem\_00053]** [ The function [Dem\\_GetEventTested](#) shall read the negated UDS DTC status bit 6 (TestNotCompletedThisOperationCycle) of the requested diagnostic event. ]([SRS\\_Diag\\_04067](#), [SRS\\_Diag\\_04096](#))

**[SWS\_Dem\_00844]** [ 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. ]()

### 7.6.1.1 Status bit support

**[SWS\_Dem\_00006]** [ The Dem module shall by default implement the current set of all UDS DTC status bits according to the definition in ISO-14229-1[1] for each diagnostic event. ]([SRS\\_Diag\\_04096](#))

Note: For this specification, the UDS DTC status byte of ISO-14229-1[1] for events is represented by the Dem data type [Dem\\_UdsStatusByteType](#) (defined in chapter [8.2.1.7](#)). If particular DTC status bits do not need to be supported (refer also to [\[SWS\\_Dem\\_00060\]](#)), such optimizations or limitations may be implemented vendor-specific.

### 7.6.1.2 Status bit update

It is a system design decision if synchronous or asynchronous event processing is used

**[SWS\_Dem\_00036]** [ In case an event is qualifying, either by reporting (passed / failed) or by reaching the [debounce counter](#) thresholds, the Dem shall per-

form the event status transition synchronously in the reporting function (e.g. [Dem\\_SetEventStatus](#)) at least for the following status bits:

- Bit 0 TestFailed
- Bit 1 TestFailedThisOperationCycle
- Bit 6 TestNotCompletedThisOperationCycle

]([SRS\\_Diag\\_04096](#))

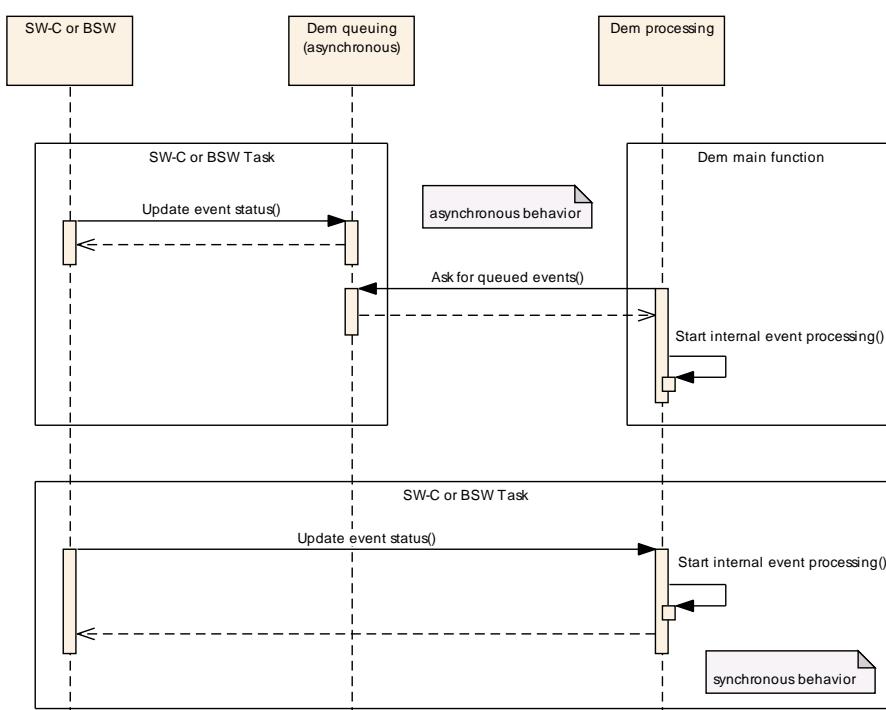
**[SWS\_Dem\_00379]** [ Depending on the design decision (synchronous or asynchronous event processing) the status update of the following bits:

- Bit 2 PendingDTC
- Bit 3 ConfirmedDTC
- Bit 7 WarningIndicatorRequested

may occur at a later point in time. ]([SRS\\_Diag\\_04096](#))

Rationale: the immediate status updates may be relevant for immediate fault reactions like update of the function inhibition (FIM).

Note: If the status update of the bits described in [\[SWS\\_Dem\\_00379\]](#) is implemented asynchronously, a queuing mechanism is needed which ensures that all the changes applied to bit 0, 1, 4, 5 and 6 will be considered when calculating bit 2, 3 and 7. Similarly, all processing and data storage associated with bit 2, 3 and 7 needs to be queued (refer to Figure 12 describing different approaches of synchronous and asynchronous event processing design).



**Figure 7.14: Synchronous and asynchronous event processing**

Note: Refer to chapter 7.6.2 for the [event memory](#) management.

**[SWS\_Dem\_01064]** [ The mirror memory is not relevant for calculation of the extended event status. ] ([SRS\\_Diag\\_04131](#))

### 7.6.1.3 Status bit transitions

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

**[SWS\_Dem\_00385]** [ After a clear command has been applied to a specific DTC (refer to chapter 7.6.2.2) 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). ] ([SRS\\_Diag\\_04096](#))

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]** [ The Dem module shall implement the status bit transitions for UDS DTC status bit 0 (TestFailed) according to figure 7.15. ] ([SRS\\_Diag\\_04096](#))

**[SWS\_Dem\_00387]** [ The Dem module shall support the configuration parameter [DemStatusBitStorageTestFailed](#) (refer to [DemGeneral](#)) to determine whether the information for UDS DTC status bit 0 (TestFailed) is stored volatile or non-volatile. ] ([SRS\\_Diag\\_04096](#))

**[SWS\_Dem\_00388]** [ If the configuration parameter [DemStatusBitStorageTestFailed](#) is set to False, the Dem module shall not retain the information for UDS DTC status bit 0 (TestFailed) over power cycles (volatile) ] ([SRS\\_Diag\\_04096](#))

**[constr\_6113] Configuration of the test failed status bit storage** [ For WWH-OBD ECU the DemStatusBitStorageTestFailed shall be set to True. ] ()

**[SWS\_Dem\_00525]** [ If the configuration parameter [DemStatusBitStorageTestFailed](#) is set to True, the Dem module shall retain the information for UDS DTC status bit 0 (TestFailed) over power cycles (non-volatile). ] ([SRS\\_Diag\\_04096](#))

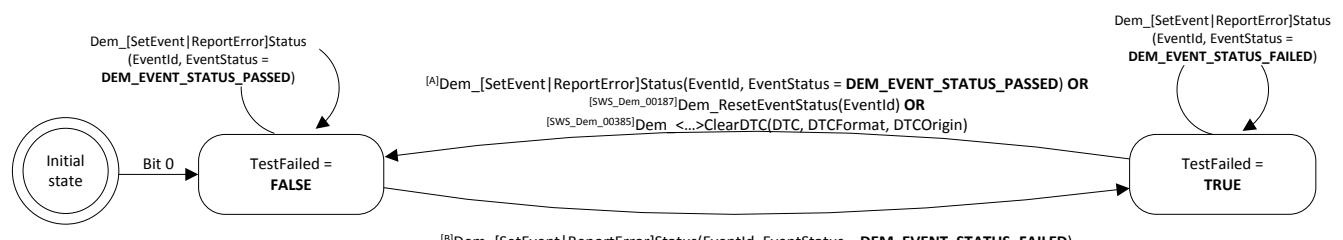
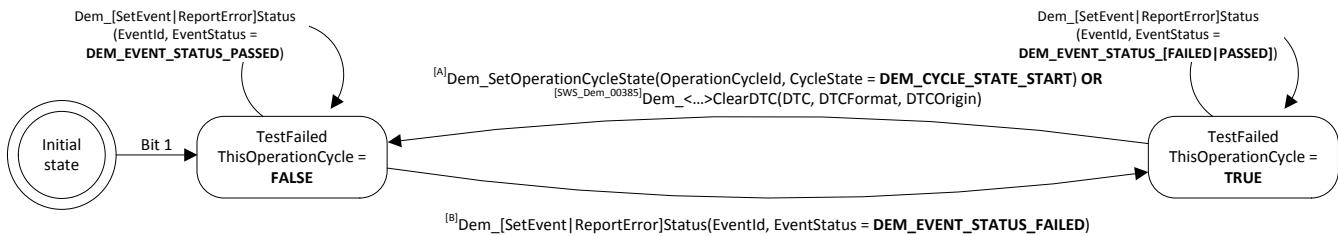


Figure 7.15: DTC status bit 0 TestFailed logic

**[SWS\_Dem\_00389]** [ The Dem module shall implement the status bit transition for UDS DTC status bit 1 (TestFailedThisOperationCycle) according to figure 7.16. ] ([SRS\\_Diag\\_04096](#))

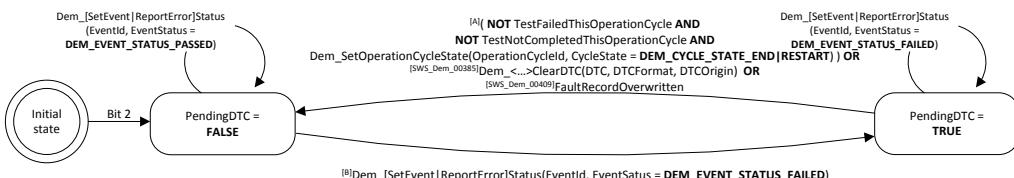
Note: The information for UDS DTC status bit 1 (TestFailedThisOperationCycle) is non-volatile, if the PendingDTC bit is used (refer to [SWS\_Dem\_00006]) or if the Dem module supports operation cycles over power cycles (refer to [DemOperationCycleStatusStorage](#)).



**Figure 7.16: DTC status bit 1 TestFailedThisOperationCycle logic**

[SWS\_Dem\_00390] The Dem module shall implement the status bit transition for UDS DTC status bit 2 (PendingDTC) according to figure 7.17. ]([SRS\\_Diag\\_04096](#))

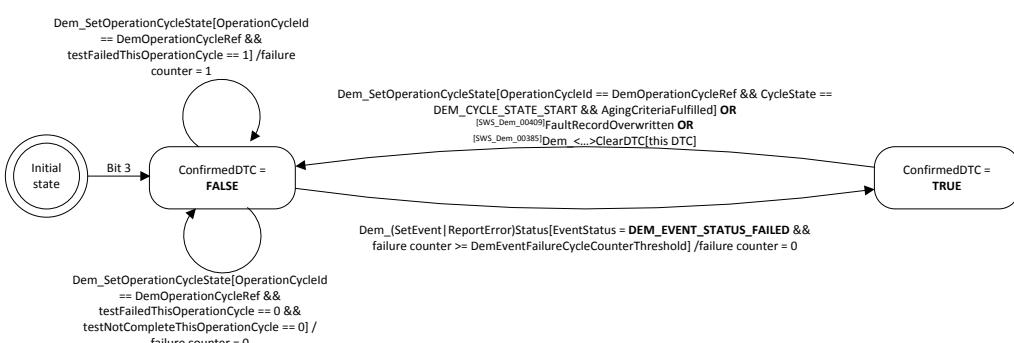
[SWS\_Dem\_01183] The information for UDS DTC status bit 2 (PendingDTC) needs to be stored non-volatile ]([SRS\\_Diag\\_04141](#))



**Figure 7.17: DTC status bit 2 PendingDTC logic**

[SWS\_Dem\_00391] The Dem module shall implement the status bit transition for UDS DTC status bit 3 (ConfirmedDTC) according to figure 7.18. ]([SRS\\_Diag\\_04096](#))

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



**Figure 7.18: DTC status bit 3 ConfirmedDTC logic**

Note: The “AgingCriteriaFulfilled” condition is specified by [SWS\_Dem\_00498]. The “FaultRecordOverwritten” condition is specified by [SWS\_Dem\_00409].

**[SWS\_Dem\_00392]** [ The Dem module shall implement the status bit transition for UDS DTC status bit 4 (TestNotCompletedSinceLastClear) according to figure 7.19 . ](SRS\_Diag\_04096)

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

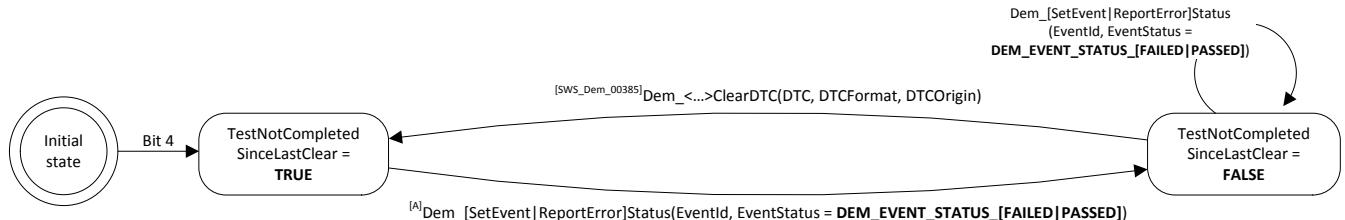


Figure 7.19: DTC status bit 4 TestNotCompletedSinceLastClear logic

**[SWS\_Dem\_00393]** [ The Dem module shall implement the status bit transition for UDS DTC status bit 5 (TestFailedSinceLastClear) according to figure 7.20 . ](SRS\_Diag\_04096)

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

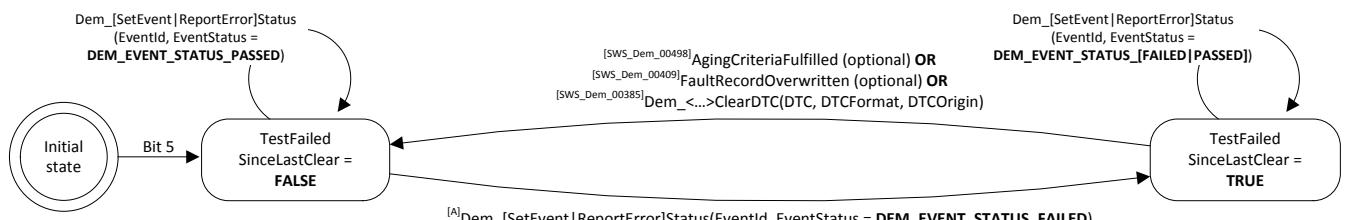
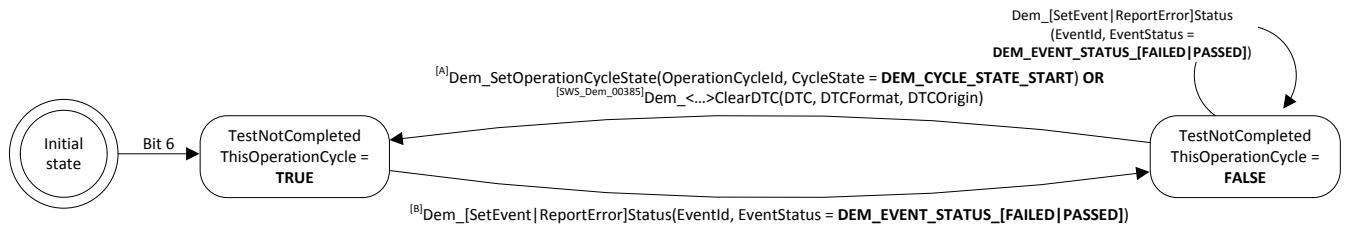


Figure 7.20: DTC status bit 5 TestFailedSinceLastClear logic

Note: The relevance of “AgingCriteriaFulfilled” condition (specified by [SWS\_Dem\_00498]) and “FaultRecordOverwritten” condition (specified by [SWS\_Dem\_00409]) depends on the configuration parameter **DemStatus-BitHandlingTestFailedSinceLastClear** (refer to [DemGeneral](#)).

**[SWS\_Dem\_00394]** [ The Dem module shall implement the status bit transition for UDS DTC status bit 6 (TestNotCompleteThisOperationCycle) according to figure 7.21 . ](SRS\_Diag\_04096)

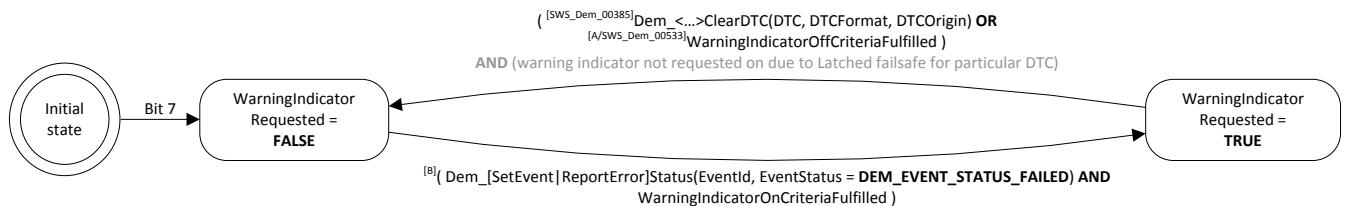
Note: The information for UDS DTC status bit 6 (TestNotCompleteThisOperationCycle) needs to be stored non-volatile, if the PendingDTC bit is used (refer to [SWS\_Dem\_00006]), or if the Dem module supports operation cycles over power cycles (refer to [DemOperationCycleStatusStorage](#)).



**Figure 7.21: DTC status bit 6 TestNotCompleteThisOperationCycle logic**

**[SWS\_Dem\_00395]** The Dem module shall implement the status bit transition for UDS DTC status bit 7 (WarningIndicatorRequested) according to figure 7.22. [\(SRS\\_Diag\\_04096\)](#)

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



**Figure 7.22: DTC status bit 7 WarningIndicatorRequested logic**

Note: The “WarningIndicatorOffCriteriaFulfilled” and the “WarningIndicatorOnCriteriaFulfilled” condition are specified in chapter 7.6.11.

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

#### 7.6.1.4 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 UDS DTC status byte.

As the TestFailed bit (UDS DTC status bit 0) is configurable in persistent storage ability (refer to configuration parameter [DemStatusBitStorageTestFailed](#)), also the meaning of active/passive is influenced:

- If the TestFailed bit is stored non-volatile, “event active” equals to TestFailed = 1 and “event passive” equals to TestFailed = 0.
- If the TestFailed bit is only stored volatile, additionally the information, if the event was already tested/reported this power cycle, is required. As long, as this information is not present, the active/passive status is undefined.

Note: For events, which are assigned to the operation cycle type DEM\_OPCYC\_POWER, the inverted TestNotCompletedThisOperationCycle bit represents the information about “already tested/reported this power cycle”.

Note: There are also ECUs, where all monitors run during startup phase. In this case, it is also sufficient, to map the active/passive status directly to the TestFailed bit.

### 7.6.1.5 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 (refer to [SWS\_Dem\_00379]), the Dem module will trigger the respective notification functions twice per event/DTC qualification.

**[SWS\_Dem\_00016]** [ The Dem module shall trigger the event-specific callback-function [EventStatusChanged](#) (refer to chapter 8.4.3.2) on each event status change. ]([SRS\\_Diag\\_04067](#))

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 [DemCallbackEventStatusChanged](#) (in [DemEvent-Parameter](#)) is used to specify one or more ports/c-callbacks per event.

Note: The respective callback-functions for FIM and Dlt are specified in chapter 7.9.

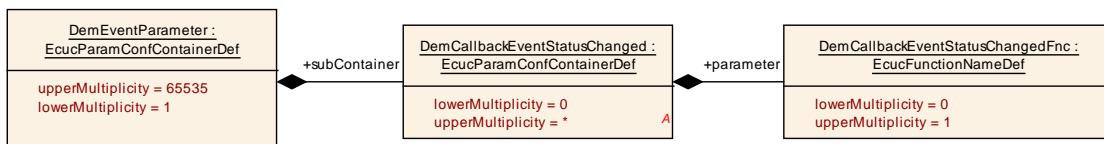


Figure 7.23: EventStatusChanged callback configuration

**[SWS\_Dem\_00284]** [ The Dem module shall trigger the callback function configured in [DemCallbackDTCStatusChanged](#) on every [DemDTC](#) status change. ]()

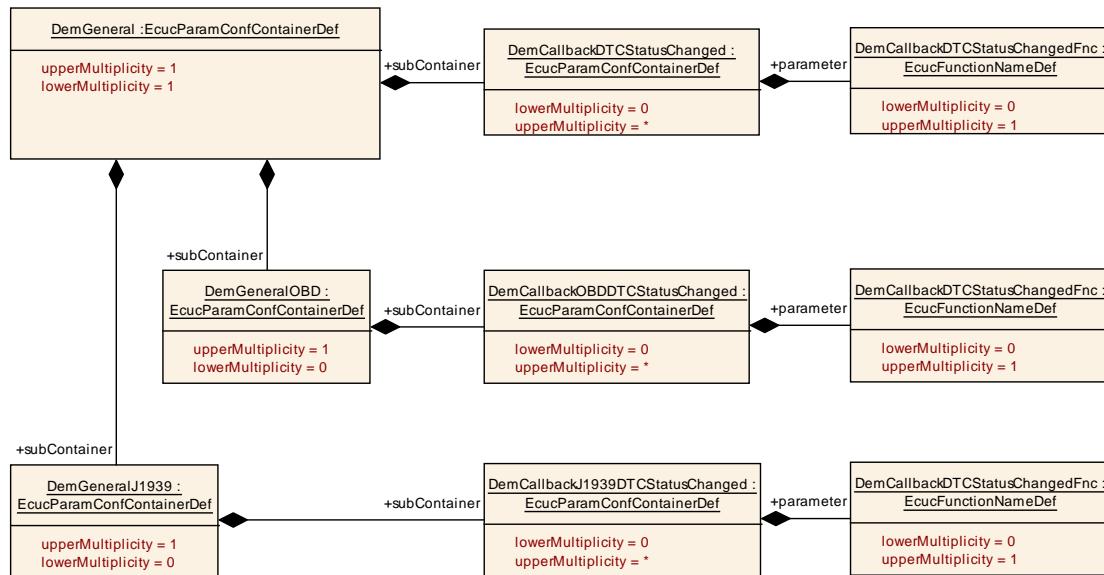
**[SWS\_Dem\_00986]** [ The Dem module shall trigger the callback function configured in [DemCallbackOBDDTCStatusChanged](#) on every [DemObdDTC](#) status change. ]()

**[SWS\_Dem\_00987]** [ The Dem module shall trigger the callback function configured in [DemCallbackJ1939DTCStatusChanged](#) on every DemJ1939DTC 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 result of EventStatusChanged may only be different to the result of DTCStatusChanged in case of [event combination](#) (refer to chapter 7.6.5).

Note: The configuration containers `DemCallbackDTCStatusChanged` (in `DemGeneral`), `DemCallbackOBDDTCStatusChanged` (in `DemGeneralOBD`), and `DemCallbackJ1939DTCStatusChanged` (in `DemGeneralJ1939`) are used to specify one or more ports/c-callbacks for the Dem module globally.



**Figure 7.24: DTCStatusChanged callback configuration**

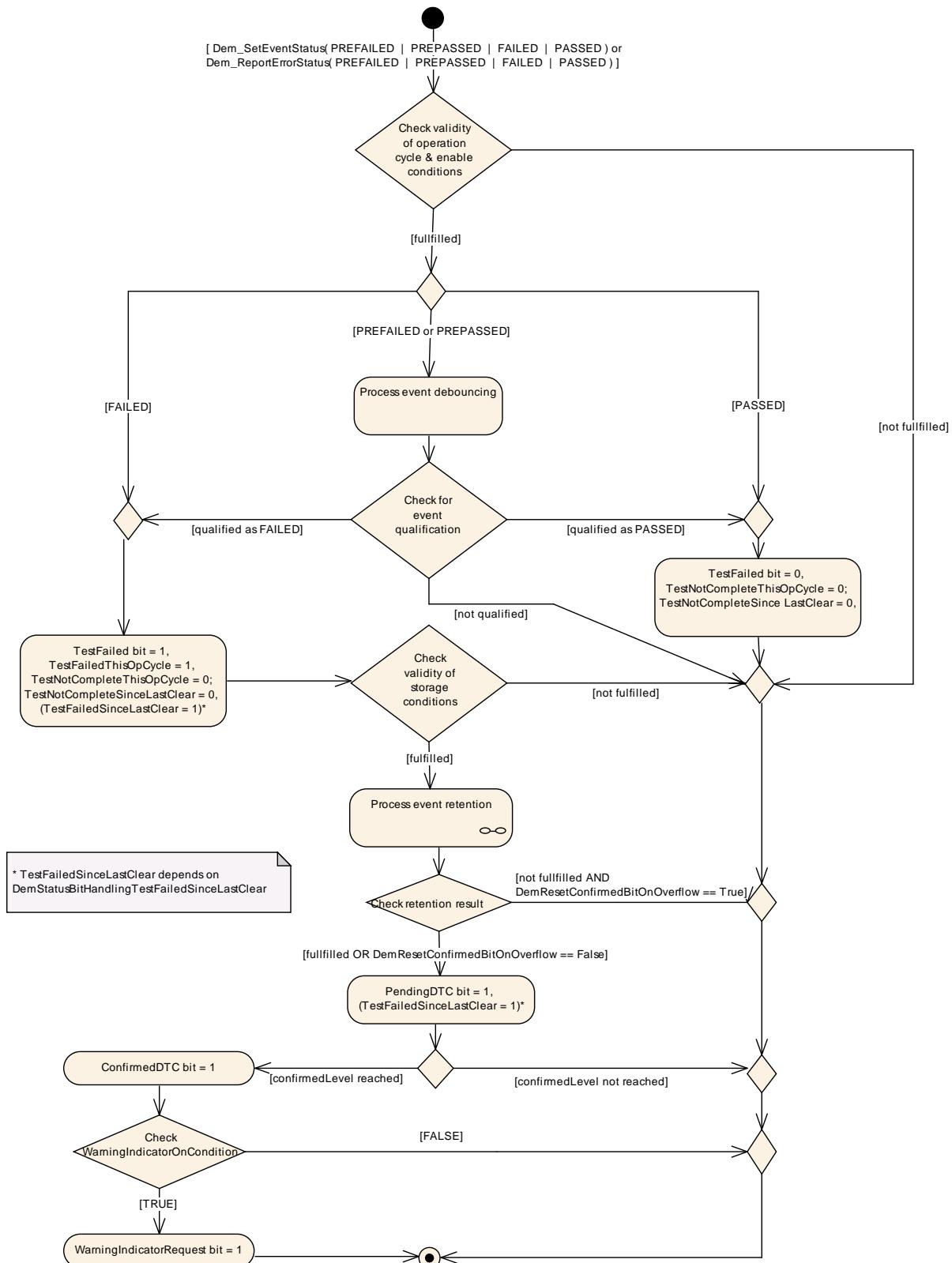
## 7.6.2 Event memory management

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\_01063]** [ The user defined memory shall have the same behavior as the primary memory(event retention, event prioritization, `aging`, displacement). ]  
*(SRS\_Diag\_04131)*

Note: The mirror memory may have a different behaviour which is project specific and is not described in this document.



**Figure 7.25: General diagnostic event storage processing**

### 7.6.2.1 Event retention

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

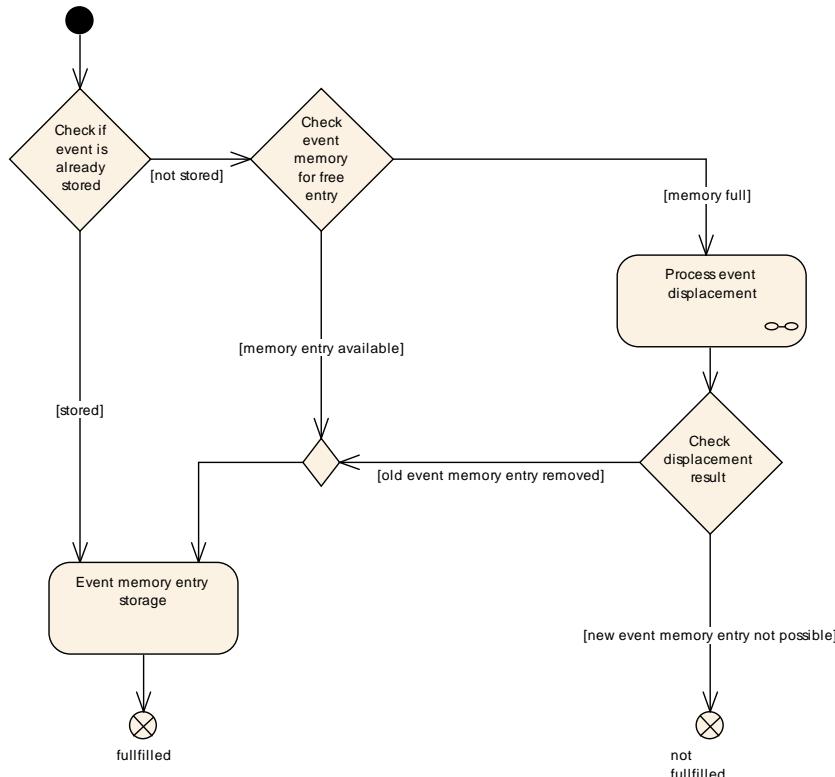


Figure 7.26: General diagnostic event retention processing

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 (refer `DemEventMemoryEntryStorageTrigger`) 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] ┌ If an event

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

the Dem module shall try to allocate according figure 7.26 an `event memory entry` in its configured `event memory` (refer to `DemMemoryDestinationRef`).  
](SRS\_Diag\_04105)

[SWS\_Dem\_00784] ┌ If an event

1. gets pending (UDS DTC status bit 2 changes from 0 to 1) and
2. the configuration parameter [DemEventMemoryEntryStorageTrigger](#) is set to [DEM\\_TRIGGER\\_ON\\_PENDING](#) and
3. no event [memory](#) is existing,

the Dem module shall try to allocate according figure [7.26](#) an [event memory entry](#) in its configured [event memory](#) (refer to [DemMemoryDestinationRef](#)).  
]([SRS\\_Diag\\_04105](#))

**[SWS\_Dem\_00922]** ┌ If an event

1. is pending (UDS DTC status bit 2 is set to 1) and gets re-qualified as failed (UDS DTC status bit 0 changes from 0 to 1) and
2. the configuration parameter [DemEventMemoryEntryStorageTrigger](#) is set to [DEM\\_TRIGGER\\_ON\\_PENDING](#) and
3. no [event memory entry](#) is existing,

the Dem module shall try to allocate according figure [7.26](#) an [event memory entry](#) in its configured [event memory](#) (refer to [DemMemoryDestinationRef](#)).  
]([SRS\\_Diag\\_04105](#))

**[SWS\_Dem\_00785]** ┌ If an event

1. gets confirmed (UDS DTC status bit 3 changes from 0 to 1) and
2. the configuration parameter [DemEventMemoryEntryStorageTrigger](#) is set to [DEM\\_TRIGGER\\_ON\\_CONFIRMED](#) and
3. no [event memory entry](#) is existing,

the Dem module shall try to allocate according figure [7.26](#) an [event memory entry](#) in its configured [event memory](#) (refer to [DemMemoryDestinationRef](#)).  
]([SRS\\_Diag\\_04105](#))

**[SWS\_Dem\_00923]** ┌ If an event

1. is confirmed (UDS DTC status bit 3 is set to 1) and gets re-qualified as failed (UDS DTC status bit 0 changes from 0 to 1) and
2. the configuration parameter [DemEventMemoryEntryStorageTrigger](#) is set to [DEM\\_TRIGGER\\_ON\\_CONFIRMED](#) and
3. no event [memory](#) exists and
4. if [DemResetConfirmedBitOnOverflow](#) is set to false,

the Dem module shall try to allocate according [Figure 7.26](#) an [event memory entry](#) in its configured [event memory](#) (refer to [DemMemoryDestinationRef](#)).  
]([SRS\\_Diag\\_04105](#))

**[SWS\_Dem\_00786]** ┌ If an event

1. has `DemDebounceAlgorithmClass` set to `DemDebounceCounterBased` and
2. the internal `debounce counter` reaches the `DemEventMemoryEntryFdcThresholdStorageValue` by a positive increment via `EventStatus` set to `preFailed` and
3. the configuration parameter `DemEventMemoryEntryStorageTrigger` is set to `DEM_TRIGGER_ON_FDC_THRESHOLD` and
4. no event `event memory` exists,

the Dem module shall try to allocate according [Figure 7.26](#) an `event memory entry` in its configured `event memory` (refer to `DemMemoryDestinationRef`). [\]\(SRS\\_Diag\\_04105\)](#)

Note: In case the `event memory entry` already exists, there might be other triggers to update content of `event memory entry`(refer chapter [7.6.7](#) “Event related data”).

### 7.6.2.2 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` (to CDDs, refer below)
- `Dem_DcmClearDTC` (for UDS to the Dcm, refer to chapter [7.10.2.3](#))
- `Dem_SetClearDTC` (for OBD to SW-Cs, refer to chapter [7.8.5.4](#))
- `Dem_J1939DcmClearDTC` (for J1939 to the J1939Dcm [7], refer to chapter [7.9.2](#)).

**[SWS\_Dem\_00659]** [\[](#) The Dem module shall provide the API `Dem_ClearDTC` to complex device drivers for deleting single DTCs, DTC groups, and all DTCs from the `event memory`. This shall trigger also initializing of related SW-Cs / BSW modules according to [\[SWS\\_Dem\\_00003\]](#). [\]](#) [\]\(SRS\\_Diag\\_04122\)](#)

Note: `Dem_ClearDTC` typically triggers further callbacks through the RTE. To indicate the respective call-tree for these runnables, a work-around is used: The respective CDD must trigger the DTC deletion using the Dem interface `CddIf` (operation `ClearDTC`, refer to chapter [8.6.2.11](#)) instead of a direct C call.

**[SWS\_Dem\_00660]** [\[](#) The API `Dem_ClearDTC` shall clear the status of all event(s) related to the specified DTC(s), as well as all associated `event memory entries` for these event(s). [\]](#) [\]\(SRS\\_Diag\\_04122\)](#)

Note: Exceptions may apply due to [\[SWS\\_Dem\\_00514\]](#).

The clearing process can be started by the different users using their related Dem\_<...>ClearDTC interface. 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\_00661]** [ The first call of any Dem\_<...>ClearDTC API (e.g. Dem\_ClearDTC) while the clearing process is not locked shall lock the clearing process for all other callers. ](*SRS\_Diag\_04065, SRS\_Diag\_04122*)

**[SWS\_Dem\_00662]** [ Any call of a Dem\_<...>ClearDTC API (e.g. Dem\_ClearDTC) while the clearing process is locked by another instance (e.g. Dem\_DcmClearDTC) shall return DEM\_CLEAR\_BUSY. ](*SRS\_Diag\_04065, SRS\_Diag\_04122*)

**[SWS\_Dem\_00663]** [ If the clearing process is locked, any subsequent call of the same Dem\_<...>ClearDTC API with the same set of parameters shall either return DEM\_CLEAR\_PENDING while DTC clearing is in progress, or return DEM\_CLEAR\_OK in case DTC clearing has finished with no errors. ](*SRS\_Diag\_04065, SRS\_Diag\_04122*)

Note: Also the interface Dem\_ClearDTC is designed to have only one caller. If for example two users are using the same set of clear parameters, the clearing may be performed twice and it is not defined which clear requester gets DEM\_CLEAR\_OK first.

**[SWS\_Dem\_00664]** [ If the clearing process is locked, any subsequent call of any Dem\_<...>ClearDTC API with a different set of parameters shall return DEM\_CLEAR\_BUSY. ](*SRS\_Diag\_04065, SRS\_Diag\_04122*)

**[SWS\_Dem\_00670]** [ If the configuration parameter DemClearDTCLimitation is set to DEM\_ONLY\_CLEAR\_ALL\_DTCS, the APIs Dem\_<...>ClearDTC shall only accept the DTC group DEM\_DTC\_GROUP\_ALL\_DTCS for UDS (DTCFormat equals DEM\_DTC\_FORMAT\_UDS) and shall return DEM\_CLEAR\_WRONG\_DTC in case any other DTC group or DTC value is requested. ](*SRS\_Diag\_04117*)

**[SWS\_Dem\_00569]** [ The Dem module shall provide a configuration parameter DemClearDTCBehavior (refer to DemGeneral) to define the clearing process of diagnostic information concerning volatile and non-volatile memory and the positive response handling for the Dcm/Cdd module. ](*SRS\_Diag\_04065, SRS\_Diag\_04122*)

Note: The interaction with the NvRAM Manager is described in chapter 7.10.5. Due to the hardware design in some cases, the direct access to the non-volatile memory is only possible during shutdown phase.

**[SWS\_Dem\_00570]** [ 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 DEM\_CLEAR\_OK after the volatile memory is cleared. ](*SRS\_Diag\_04065, SRS\_Diag\_04122*)

**[SWS\_Dem\_00571]** [ 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 DEM\_CLEAR\_OK (refer to chapter 8.2.2.12) after the volatile memory is cleared

and clearing of the non-volatile memory is triggered. ](*SRS\_Diag\_04065, SRS\_Diag\_04122*)

**[SWS\_Dem\_00572]** [ 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_OK` (refer to chapter 8.2.2.12) after the volatile memory and the non-volatile memory is cleared. ](*SRS\_Diag\_04065, SRS\_Diag\_04122*)

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_DcmClearDTC`, the Dcm module sends the positive response (refer to [10]).

**[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` (refer to `Dem_ReturnClearDTCType`) if the clearing of the non-volatile memory fails. ](*SRS\_Diag\_04065*)

**[SWS\_Dem\_00573]** [ The Dem module shall provide the configuration parameter `DemTriggerMonitorInitBeforeClearOk`. If `DemTriggerMonitorInitBeforeClearOk` is set to TRUE, the Dem\_<...>ClearDTC shall not return `DEM_CLEAR_OK` until the events initialization is triggered (callback function `InitMonitorForEvent` is called). If `DemTriggerMonitorInitBeforeClearOk` is set to FALSE, `DEM_CLEAR_OK` is not considering Event initialization (`InitMonitorForEvent`). ](*SRS\_Diag\_04065, SRS\_Diag\_04122*)

**[SWS\_Dem\_00514]** [ The Dem module shall be able to suppress the DTC-deletion in Dem\_<...>ClearDTC. The callback `DemClearEventAllowed`, which is configurable/optional per event, shall realize this suppression handling. ](*SRS\_Diag\_04092, SRS\_Diag\_04122*)

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

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]** [ The Dem module shall call the callback `ClearEventAllowed` for each event it is configured for, before clearing this event (via `Dem_<...>ClearDTC`). ]([SRS\\_Diag\\_04092](#), [SRS\\_Diag\\_04122](#))

**[SWS\_Dem\_00667]** [ If the out-parameter ‘Allowed’ of the callback `ClearEventAllowed` returns “false” (and the return value is equal to `E_OK`) after the execution of this callback function, the respective `event memory entry` and Dem-internal data values must not be cleared. ]([SRS\\_Diag\\_04092](#), [SRS\\_Diag\\_04122](#))

**[SWS\_Dem\_00668]** [ If the out-parameter ‘Allowed’ of the callback `ClearEventAllowed` returns “false” and the configuration parameter `DemClearEventAllowedBehavior` (refer to `DemCallbackClearEventAllowed`) is set to `DEM_NO_STATUS_BYTE_CHANGE`, the related event status byte shall not be modified. ]([SRS\\_Diag\\_04092](#), [SRS\\_Diag\\_04122](#))

**[SWS\_Dem\_00669]** [ If the out-parameter ‘Allowed’ of the callback `ClearEventAllowed` returns “false” and the configuration parameter `DemClearEventAllowedBehavior` (refer to `DemCallbackClearEventAllowed`) is set to `DEM_ONLY_THIS_CYCLE_AND_READINESS`, the related event status bits 1 (`TestFailedThisOperationCycle`), 4 (`TestNotCompletedSinceLastClear`), 5 (`TestFailedSinceLastClear`), and 6 (`TestNotCompletedThisOperationCycle`) shall be reset. ]([SRS\\_Diag\\_04092](#), [SRS\\_Diag\\_04122](#))

**[SWS\_Dem\_00516]** [ If the return value of the callback `ClearEventAllowed` is equal to `E_NOT_OK`, the return value shall not be changed and event-deletion is allowed. ]([SRS\\_Diag\\_04092](#), [SRS\\_Diag\\_04122](#))

### 7.6.2.3 Event memory overflow indication

**[SWS\_Dem\_00397]** [ 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`. ]([SRS\\_Diag\\_04093](#))

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]** [ The Dem module shall provide the API `Dem_GetEventMemoryOverflow` (refer to chapter [8.3.3.26](#)) to provide access to the `event memory` overflow indication status of the respective `event memory`. ]([SRS\\_Diag\\_04093](#))

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]** [ The Dem module shall provide the API `Dem_GetNumberOfEventMemoryEntries` (refer to chapter [8.3.3.27](#)) to return the number of `event memory entries` currently stored in the `event memory`. ]([SRS\\_Diag\\_04109](#))

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

**[SWS\_Dem\_00399]** [ The [event memory](#) overflow indication of the respective [event memory](#) shall be reset, if all DTCs of this memory are deleted by Dem\_<...>ClearDTC. ]([SRS\\_Diag\\_04093](#))

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).

#### 7.6.2.4 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]** [ 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](#). ]([SRS\\_Diag\\_04118](#))

Note: If the [event memory](#) size is configured to cover all possible events, no displacement will occur.

**[SWS\_Dem\_00401]** [ The Dem module provides the configuration parameter [DemEventDisplacementStrategy](#) (refer to [DemGeneral](#) defining whether the existing [event memory entry](#) can be displaced or not. ]([SRS\\_Diag\\_04118](#))

**[SWS\_Dem\_00402]** [ If event displacement is disabled ([DemEventDisplacementStrategy](#) selects [DEM\\_DISPLACEMENT\\_NONE](#)), the [Dem](#) module shall not displace existing [event memory entries](#) if the [event memory](#) is full. ]([SRS\\_Diag\\_04118](#))

**[SWS\_Dem\_00406]** [ If event displacement is enabled ([DemEventDisplacementStrategy](#) selects [DEM\\_DISPLACEMENT\\_FULL](#) or [DEM\\_DISPLACEMENT\\_PRIO\\_OCC](#)), the Dem module shall perform the following sequence by combination of the different displacement criteria (refer to figure [7.27](#)):

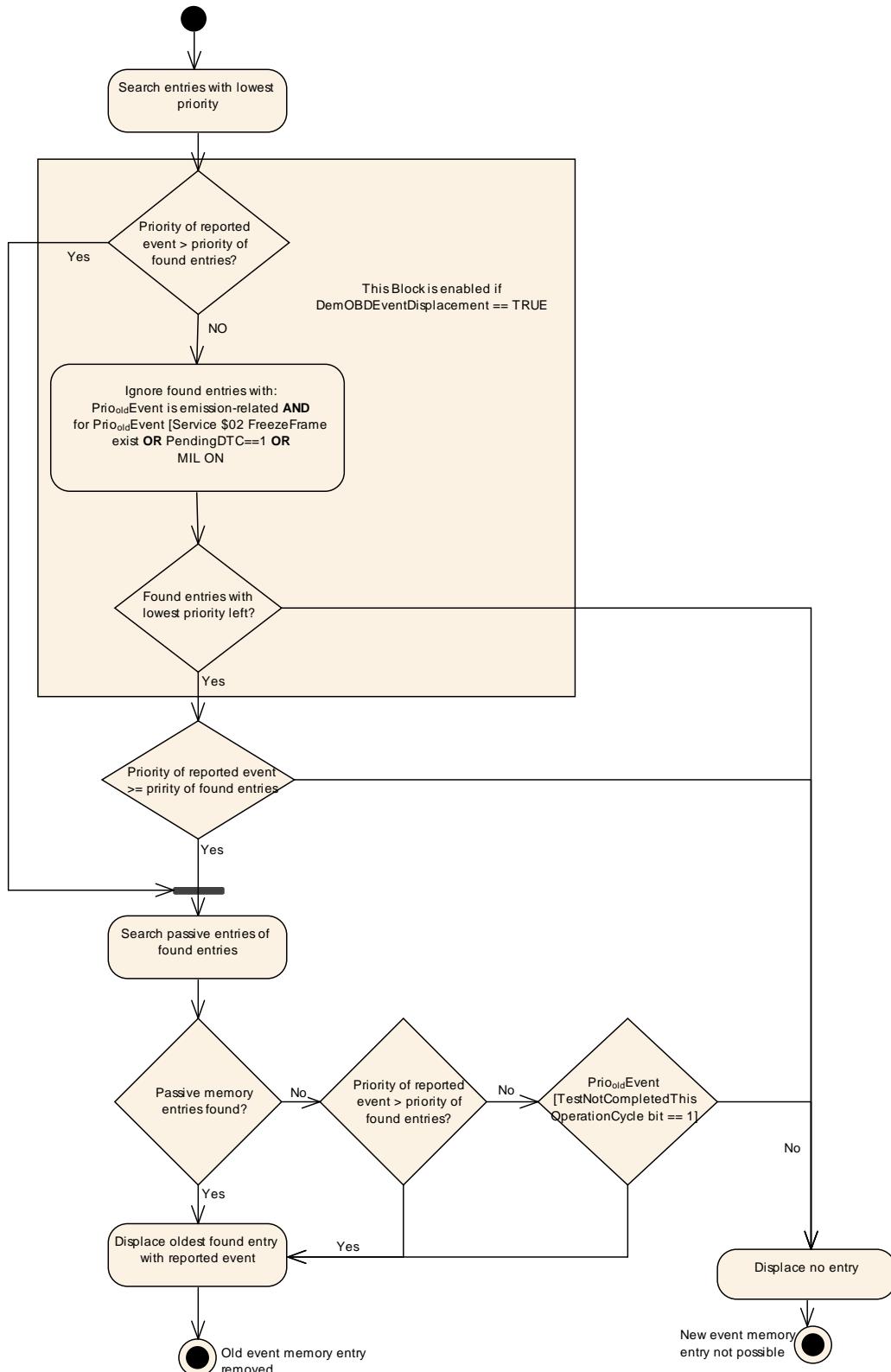
1. Priority (refer [\[SWS\\_Dem\\_00403\]](#))
  2. Active/Passive status (configurable, only for [DEM\\_DISPLACEMENT\\_FULL](#)) (refer [\[SWS\\_Dem\\_00404\]](#))
  3. Occurrence (refer [\[SWS\\_Dem\\_00405\]](#))
- ]([SRS\\_Diag\\_04118](#))

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]** [ If displacement strategy is **DEM\_DISPLACEMENT\_FULL** or **DEM\_DISPLACEMENT\_PRIO\_OCC**, the Dem module shall displace lower prioritized events by higher prioritized events. ](*SRS\_Diag\_04118*)

**[SWS\_Dem\_00404]** [ If displacement strategy is **DEM\_DISPLACEMENT\_FULL**, the Dem module shall displace passive events preferably (refer to chapter **7.6.1.4**). ](*SRS\_Diag\_04118*)

**[SWS\_Dem\_00405]** [ If displacement strategy is **DEM\_DISPLACEMENT\_FULL** or **DEM\_DISPLACEMENT\_PRIO\_OCC**, 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**]). ](*SRS\_Diag\_04118*)



**Figure 7.27: Combined displacement criteria processing**

**[SWS\_Dem\_00407]** If no [event memory entry](#) for displacement was identified, the Dem module shall discard the storage request. ]([SRS\\_Diag\\_04118](#))

**[SWS\_Dem\_00692]** [ If an [event memory entry](#) has the same priority like the reported event and the existing event is not tested in this OperationCycle (TestNotCompletedThisOperationCycle bit == 1) the Dem module shall displace this event. ]([SRS\\_Diag\\_04118](#), [SRS\\_Diag\\_04071](#))

**[SWS\_Dem\_00693]** [ If an [event memory entry](#) has the same priority like the reported event and the existing event is tested in this OperationCycle (TestNotCompletedThisOperationCycle bit == 0) the Dem shall discard the reported event because the TestFailed status of the existing event is not verified. ]([SRS\\_Diag\\_04076](#))

**[SWS\_Dem\_00695]** [ 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 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==1) ]([SRS\\_Diag\\_04102](#))

**[SWS\_Dem\_00696]** [ If the configuration parameter [DemOBDEventDisplacement](#) is set to FALSE the Dem module shall process the displacement without the OBD-behavior (refer to figure 7.27). ]([SRS\\_Diag\\_04102](#))

**[SWS\_Dem\_00408]** [ If the [event memory entry](#) for displacement was identified, the Dem module shall remove this [event memory entry](#) from the [event memory](#). ]([SRS\\_Diag\\_04118](#))

Note: This removed [event memory entry](#) should now used for the pending storage trigger.

**[SWS\_Dem\_00409]** [

If an [event memory entry](#) was removed during displacement, the [Dem](#) module shall reset the UDS status bit 2 (PendingDTC) and UDS status bit 3 (ConfirmedDTC) to 0 if the configuration parameter [DemResetConfirmedBitOnOverflow](#) (refer to [DemGeneral](#)) is set to true. ]([SRS\\_Diag\\_04067](#), [SRS\\_Diag\\_04096](#), [SRS\\_Diag\\_04118](#))

**[SWS\_Dem\_01186]** [ If an [event memory entry](#) was removed during displacement, the UDS status bit 5 (TestFailedSinceLastClear) shall be reset to 0 in case of [DemStatusBitHandlingTestFailedSinceLastClear](#) is set to [DEM\\_STATUS\\_BIT\\_AGING\\_AND\\_DISPLACEMENT](#) and [DemResetConfirmedBitOnOverflow](#) is set to true. ]([SRS\\_Diag\\_04067](#), [SRS\\_Diag\\_04096](#), [SRS\\_Diag\\_04118](#))

### 7.6.2.5 Reporting order of event memory entries

**[SWS\_Dem\_00410]** [ The [Dem](#) module shall report [DTCs](#) in the chronological order of the event storage (refer to API [Dem\\_DcmGetNextFilteredDTC](#)), 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

] (SRS\_Diag\_04070)

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 DTC status mask parameters, additional resources may be necessary.

**[SWS\_Dem\_00787]** [ If an ‘stored event’ gets re-qualified as failed (UDS DTC 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**. ] (SRS\_Diag\_04070)

**[SWS\_Dem\_00411]** [ If the Dem module is requested to report in chronological order, the most recent **event memory entry** shall be reported at first. ] (SRS\_Diag\_04102)

**[SWS\_Dem\_00412]** [ New added event memory entries shall be always stored as the most recent memory entries to keep the chronological order. In case of an update of a event memory location (e.g. event reports again) the chronological order of this **event memory entry** shall remain the same. ] (SRS\_Diag\_04070, SRS\_Diag\_04102)

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..

### 7.6.3 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]** [ The Dem module shall support the event-specific configuration of debounce algorithms by using the configuration container **DemDebounceAlgorithmClass**. ] ([SRS\\_Diag\\_04068](#))

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

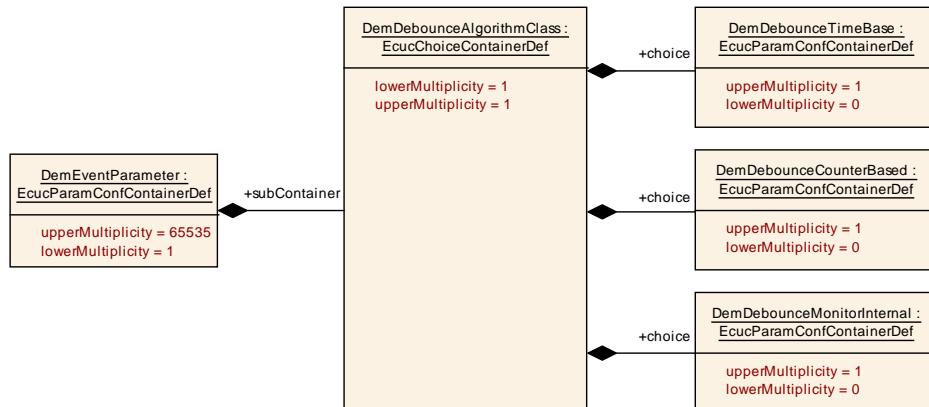


Figure 7.28: Event-specific debounce algorithms

### 7.6.3.1 Counter based debounce algorithm

**[SWS\_Dem\_00526]** [ The Dem module shall provide a configuration parameter **DemDebounceCounterBasedSupport** (refer to [DemGeneral](#)) to enable or disable the Dem-internal counter based debouncing. ] ([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00414]** [ 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. ] ([SRS\\_Diag\\_04068](#))

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

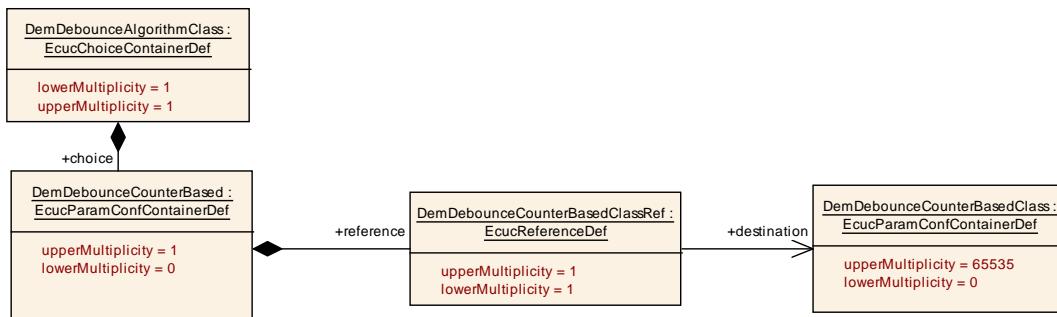


Figure 7.29: Counter based debounce algorithm

**[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 de-

bounce counter to map the internal counter values linearly to the external values.  
]([SRS\\_Diag\\_04010](#))

**[SWS\_Dem\_00416]** [ The Dem module shall provide the configuration parameter `DemDebounceCounterFailedThreshold` used to define the event-specific limit indicating the failed status (active). ]([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00417]** [ The Dem module shall provide the configuration parameter `DemDebounceCounterFailedThreshold` used to define the event-specific limit indicating the passed status (passive). ]([SRS\\_Diag\\_04010](#), [SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00418]** [ 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` (refer to chapter 8.2.1.4). ]([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00419]** [ 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` (refer to chapter 8.2.1.4). ]([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00420]** [ 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. ]([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00421]** [ 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. ]([SRS\\_Diag\\_04096](#), [SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00422]** [ The Dem module shall provide the configuration parameter `DemDebounceCounterJumpDown` for enabling the jump-down behavior. ]([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00423]** [ 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]). ]([SRS\\_Diag\\_04068](#))

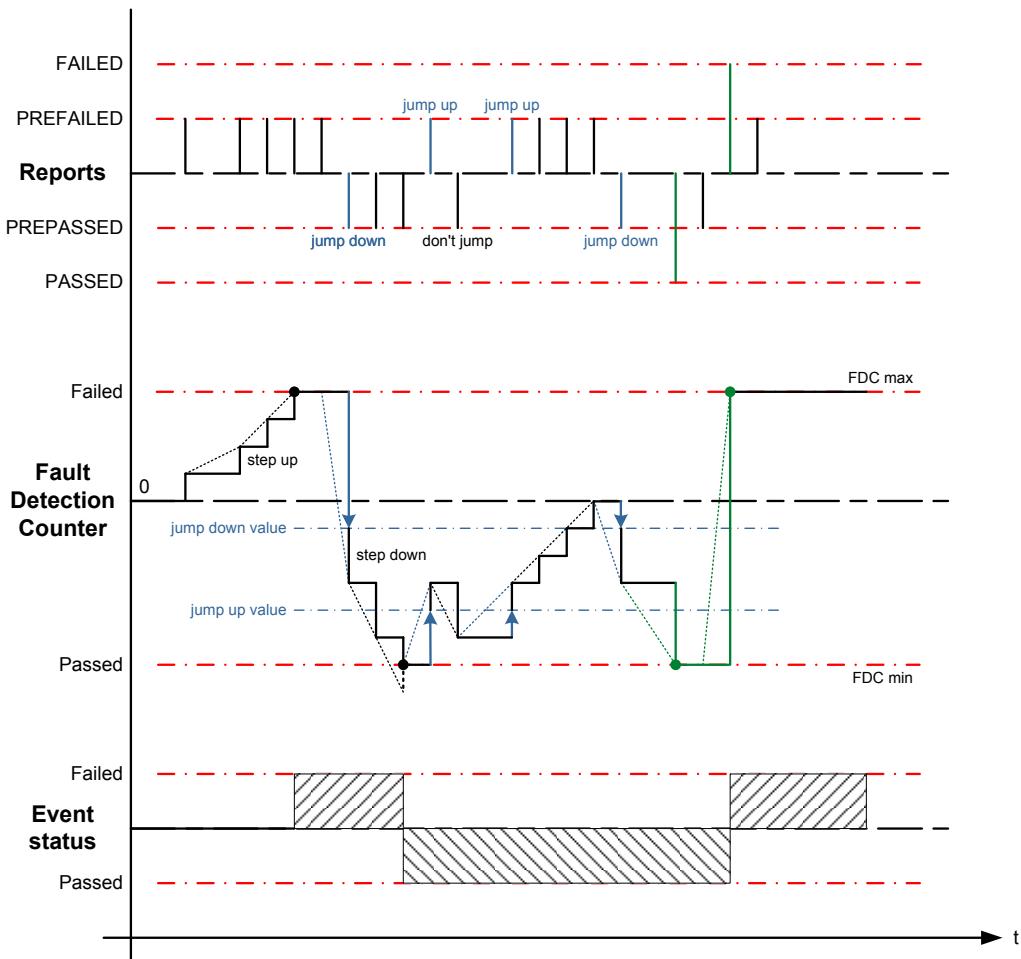
Note: This will only happen if the counting direction changes from incrementing to decrementing.

**[SWS\_Dem\_00424]** [ The Dem module shall provide the configuration parameter `DemDebounceCounterJumpUp` for enabling the jump-up behavior. ]([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00425]** [ 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](#)]).  
 ] ([SRS\\_Diag\\_04068](#))

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



**Figure 7.30: Example of counter based debouncing (including jump behaviour)**

Note: In Figure 28, the value of Fault Detection Counter maps linearly to the internal `debounce counter` value.

The action `DEM_DEBOUNCE_STATUS_FREEZE` of the API `Dem_ResetEventDebounceStatus` 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_FREEZE` is covered in chapter 7.6.6. The DemDebounceBehavior `DEM_DEBOUNCE_RESET` is covered by [[SWS\\_Dem\\_00654](#)] and [[SWS\\_Dem\\_00677](#)]. This is shown in Figure 29.

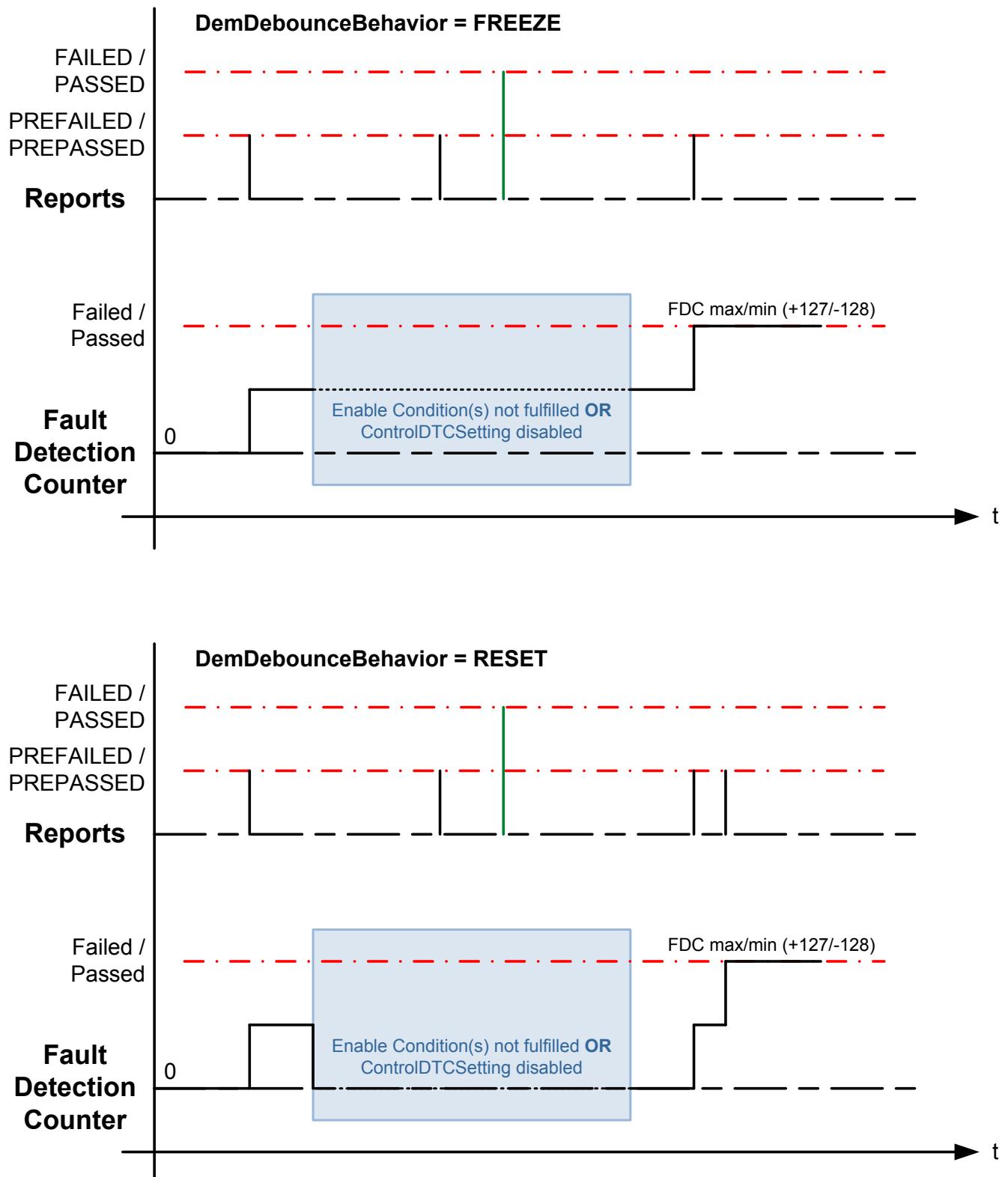


Figure 7.31: Counter based debouncing behavior on enable conditions

**[SWS\_Dem\_00674]** 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]). ](SRS\_Diag\_04124)

Note: DemDebounceCounterStorage is not supported for time based debouncing.

**[SWS\_Dem\_00675]** [ 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`). ] ([SRS\\_Diag\\_04124](#))

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]** [ If development error detection is enabled and `Dem_ReportErrorStatus` 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`. ] ([SRS\\_Diag\\_04124](#))

### 7.6.3.2 Time based debounce algorithm

**[SWS\_Dem\_00527]** [ The Dem module shall provide a configuration parameter `DemDebounceTimeBasedSupport` (refer to `DemGeneral`) to enable or disable the Dem-internal time based debouncing. ] ([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00426]** [ 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. ] ([SRS\\_Diag\\_04068](#))

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

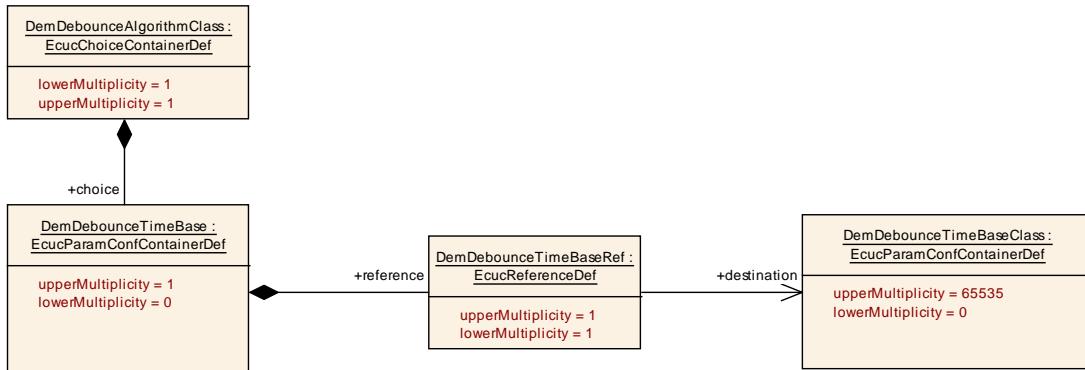


Figure 7.32: Time based debounce algorithm

**[SWS\_Dem\_00427]** [ 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 (refer to Figure 31). ] ([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00428]** [ 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 (refer to Figure 31). ]([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00429]** [ 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. ]([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00430]** [ 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. ]([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00431]** [ 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). ]([SRS\\_Diag\\_04096](#), [SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00432]** [ The Dem module shall start the internal debounce timer to qualify the reported event as passed when the monitor reports **DEM\_EVENT\_STATUS\_PREPASSED** (refer to chapter 8.2.1.4). ]([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00433]** [ 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. ]([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00434]** [ The Dem module shall provide the configuration parameter DemDebounceTimePassedThreshold used to define the event-specific delay indicating the passed status (not active). ]([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00435]** [ 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). ]([SRS\\_Diag\\_04068](#))

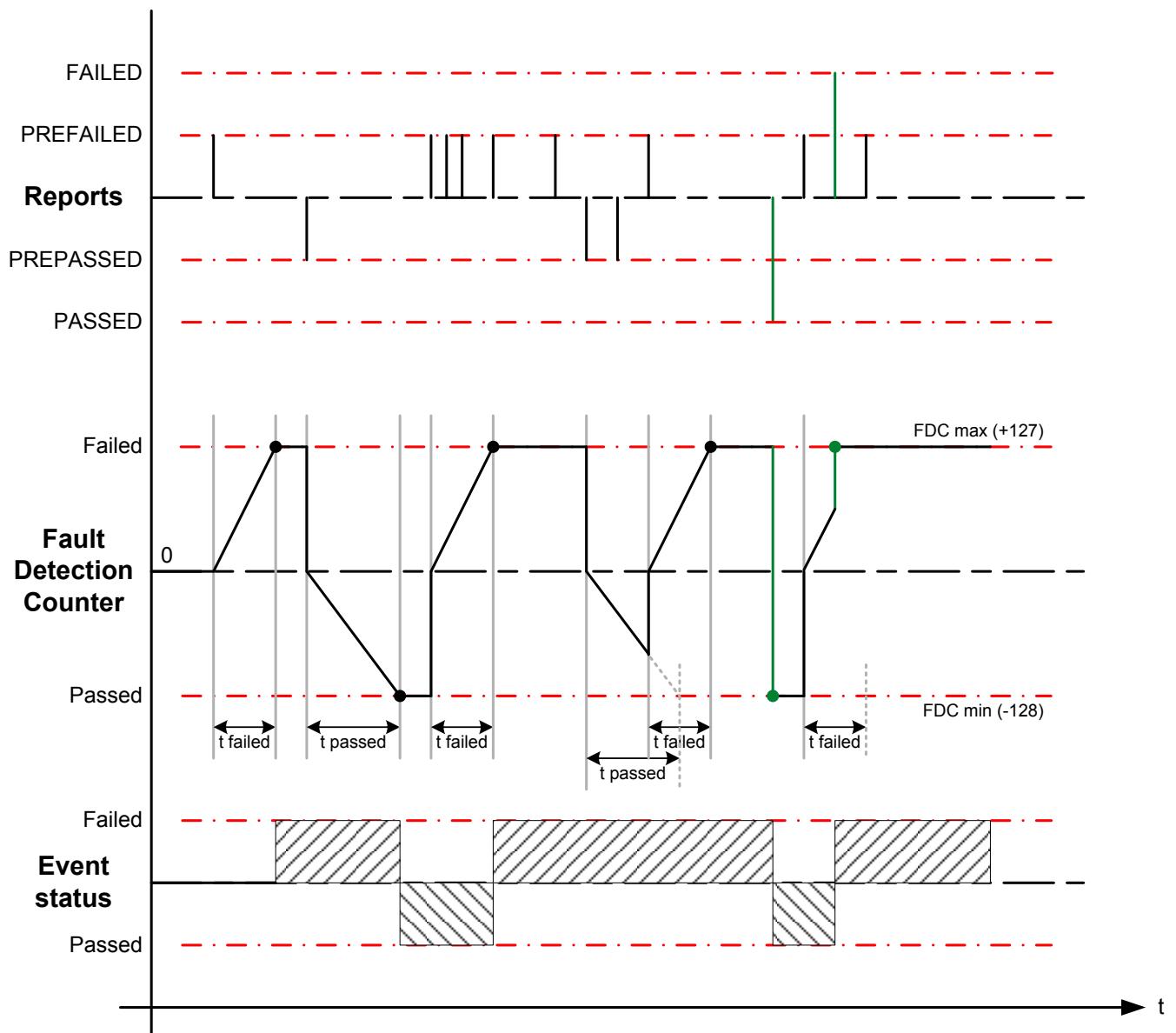


Figure 7.33: Example of time based debouncing

**[SWS\_Dem\_00685]** [ If the API `Dem_ResetEventDebounceStatus` (refer to chapter 8.3.3.4) 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]** [ If the configuration parameter `DemDebounceBehavior` is set to `DEM_DEBOUNCE_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. ]([SRS\\_Diag\\_04125](#))

**[SWS\_Dem\_00678]** [ If the configuration parameter `DemDebounceBehavior` is set to `DEM_DEBOUNCE_FREEZE`, the Dem module shall freeze the internal de-

bounce timer when ControlDTCSetting is set to disabled for the related event.  
]([SRS\\_Diag\\_04125](#))

**[SWS\_Dem\_00656]** [ 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). ]([SRS\\_Diag\\_04125](#))

The DemDebounceBehavior DEM\_DEBOUNCE\_RESET is covered by [[SWS\\_Dem\\_00654](#)] and [[SWS\\_Dem\\_00677](#)].

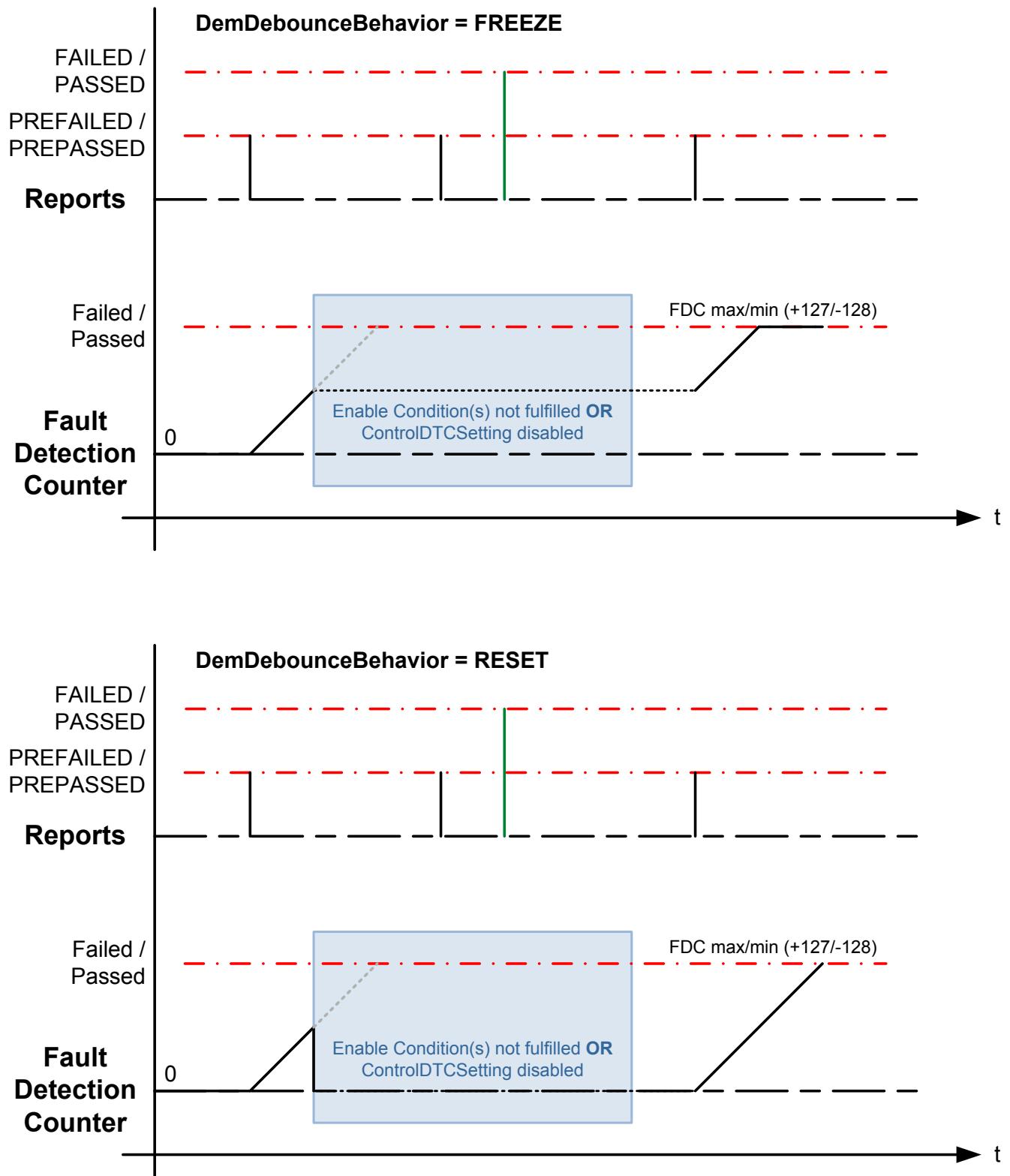


Figure 7.34: Time based debouncing behavior on enable conditions

### 7.6.3.3 Further specific debounce algorithms

The Dem module may be extended with other specific debounce algorithms, if an application requires particular debounce algorithms implemented Dem-internally.

**[SWS\_Dem\_00436]** [ If the Dem module implements application-specific debounce algorithms, the event states `DEM_EVENT_STATUS_PREFAIRED` and `DEM_EVENT_STATUS_PREPASSED` shall be used within `Dem_SetEventStatus`) indicating the maturation process. ] ([SRS\\_Diag\\_04068](#))

### 7.6.3.4 Monitor internal debounce algorithm

**[SWS\_Dem\_00437]** [ If the configuration container `DemDebounceAlgorithmClass` is set to `DemDebounceMonitorInternal`, the Dem module shall not use a Dem-internal debounce mechanism for each individual event, to qualify the reported event. ] ([SRS\\_Diag\\_04068](#))

Note: The monitor is not allowed to report the event states `DEM_EVENT_STATUS_PREFAIRED` and `DEM_EVENT_STATUS_PREPASSED` for monitor internal debouncing.

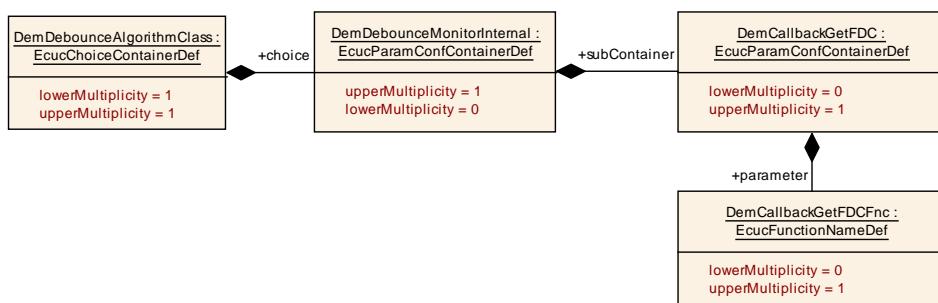


Figure 7.35: Monitor internal debounce algorithm

### 7.6.3.5 Debounce algorithm initialization and reset conditions

**[SWS\_Dem\_00438]** [ If Dem-internal debouncing is configured, the Dem module shall reset the Dem-internal debounce algorithm when `Dem_PrelInit` has been called. ] ([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00343]** [ After receiving a command for clearing the `event memory` (refer to chapter 7.6.2.2), the Dem-internal debounce algorithm shall be reset, presuming `event debouncing` is handled Dem-internally. ] ([SRS\\_Diag\\_04117](#), [SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00684]** [ If the API `Dem_ResetEventDebounceStatus` is called with `DEM_DEBOUNCE_STATUS_RESET`, it shall reset the related fault detection counter. ] ([SRS\\_Diag\\_04068](#), [SRS\\_Diag\\_04068](#))

Note: The internal `debounce counter` is reset to zero. The reset of an internal debounce timer will also stop this timer.

**[SWS\_Dem\_00344]** [ If Dem-internal debouncing is configured, the Dem module shall reset the Dem-internal debounce algorithm upon starting a new operation cycle (refer to chapter 7.6.8). ] ([SRS\\_Diag\\_04068](#))

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

**[SWS\_Dem\_00654]** [ If the configuration parameter DemDebounceBehavior is set to `DEM_DEBOUNCE_RESET` and the function `Dem_SetEnableCondition` (refer to chapter 7.6.6) sets one configured enable condition for the event to not fulfilled, it shall reset the related fault detection counter(s). ] ([SRS\\_Diag\\_04125](#))

**[SWS\_Dem\_00677]** [ If the configuration parameter DemDebounceBehavior is set to `DEM_DEBOUNCE_RESET`, the function `Dem_DcmDisableDTCSetting` (refer to chapter 7.10.2.4) shall reset the related fault detection counter(s). ] ([SRS\\_Diag\\_04125](#))

### 7.6.3.6 Fault detection counter retrieval

**[SWS\_Dem\_00204]** [ The event-specific fault detection counter shall be accessible by using the API `Dem_GetFaultDetectionCounter` (refer to chapter 8.3.3.20). ] ([SRS\\_Diag\\_04068](#), [SRS\\_Diag\\_04010](#))

The fault detection counter can be:

1. located inside the Dem, if Dem-internal debouncing is configured (refer to chapter 7.6.3.1 and chapter 7.6.3.2).
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]** [ 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. ] ([SRS\\_Diag\\_04068](#), [SRS\\_Diag\\_04010](#))

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

**[SWS\_Dem\_00439]** [ If the callback-function `GetFaultDetectionCounter` returns other than `E_OK`, this return value shall also be returned by the API `Dem_GetFaultDetectionCounter`. ] ([SRS\\_Diag\\_04068](#), [SRS\\_Diag\\_04010](#))

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

**[SWS\_Dem\_00671]** [ If the callback-function [GetFaultDetectionCounter](#) is not configured (refer to configuration container [DemCallbackGetFDC](#)) for a given event, the API [Dem\\_GetFaultDetectionCounter](#) shall return DEM\_E\_NO\_FDC\_AVAILABLE. ]([SRS\\_Diag\\_04010](#))

### 7.6.3.7 Fault detection counter reporting

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

**[SWS\_Dem\_00788]** [ 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 for all events referencing a UDS DTC. ]([SRS\\_Diag\\_04125](#))

**[SWS\_Dem\_00789]** [ If the UDS DTC of [[[SWS\\_Dem\\_00788](#)]] is referenced by multiple events or an [combined DTC](#), the event with the highest maximum FDC value shall be used to report. ]([SRS\\_Diag\\_04125](#))

**[SWS\_Dem\_00790]** [ The maximum FDC during current operation cycle shall be reset to zero with each (re-)start operation cycle (refer to [DemOperationCycleRef](#)). ]([SRS\\_Diag\\_04125](#))

**[SWS\_Dem\_00791]** [ 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.]([SRS\\_Diag\\_04125](#))

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

**[SWS\_Dem\_00792]** [ 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 for all events referencing a UDS DTC. ]([SRS\\_Diag\\_04125](#))

**[SWS\_Dem\_00793]** [ If the UDS DTC of [[SWS\\_Dem\\_00793](#)] is referenced by multiple events or an [combined DTC](#), the event with the highest maximum FDC value shall be used to report. ]([SRS\\_Diag\\_04125](#))

**[SWS\_Dem\_00794]** [ The maximum FDC since last clear shall be reset to zero with each clear DTC command affecting this particular event. ]([SRS\\_Diag\\_04125](#))

**[SWS\_Dem\_00795]** [ 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. ]([SRS\\_Diag\\_04125](#))

## 7.6.4 Fault confirmation

After reporting, a fault and entering the pending status, the fault confirmation process begins within the Dem (refer to [Figure 7.25 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 [DemEventFailureCycleCounterThreshold](#) per event (refer to [DemEventParameter](#)) defining the maximum number of tested and failed cycles, before the event becomes “confirmed”, i.e. enters the confirmed state. ]([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_00529]** [ The configuration parameter [DemOperationCycleRef](#) (refer to [DemEventParameter](#)) shall be used for calculating the trip counter according ISO14229-1[1] (see figure [7.18](#)). ]([SRS\\_Diag\\_04067](#))

Note: This cycle type could be for example equal to [DEM\\_OPCYC\\_OBD\\_DCY](#) or to [DEM\\_OPCYC\\_POWER](#) and the combination of handling the counters for a specific cycle type is called “fault confirmation”.

The healing confirmation process (described in chapter [7.6.10](#)) 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 DTC status bit 3 (ConfirmedDTC) is set during the qualification to failed (according to [\[SWS\\_Dem\\_00391\]](#) and [\[SWS\\_Dem\\_00379\]](#)).

### 7.6.4.1 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 representative 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 referencing to the event with the counter value “closest to MIL ON”. ]([SRS\\_Diag\\_04069](#))

**[SWS\_Dem\_00968]** [ The healing / [aging](#) for each event shall to be performed individually. ]([SRS\\_Diag\\_04076](#))

### 7.6.5 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]** [ The following [event combination](#) types are supported:

1. Combination on storage: configuration parameter [DemEventCombinationSupport](#) is set to [DEM\\_EVCOMB\\_ONSTORAGE](#). The [combined DTC](#) is stored and updated in a single [event memory entry](#).
2. Combination on retrieval: configuration parameter [DemEventCombinationSupport](#) is set to [DEM\\_EVCOMB\\_ONRETRIEVAL](#). Each event is stored in a separate [event memory](#) location.
3. Disabled: configuration parameter [DemEventCombinationSupport](#) is set to [DEM\\_EVCOMB\\_DISABLED](#) and [event combination](#) is not used

]()

**[SWS\_Dem\_00024]** [ In case [DemEventCombinationSupport](#) by referencing from each event to the same [DTC](#).] ([SRS\\_Diag\\_04073](#))

Note: For [OBD](#) it is also possible to assign multiple time the same [DTC](#) number via [DemObdDTCRef](#). But as a difference to [UDS](#) and J1939 DTCs the OBD DTCs are reported multiple times to the [Dcm](#).

Based on the configuration of the event related data the Dem module allows two different types of [event combination](#) (assign event related data to the combine event/DTC or to the sub-events).

Note: The primary use-case of [event combination](#) is to map more than one event (represented by an [EventId](#)) to only one DTC number. If combination “on storage” is used, the Dem module uses only one [event memory entry](#) and its related data for all events combined to the DTC. If combination on retrieval is used, each event is stored in a separate [event memory](#) location with its own set of event related data. The Dem module handles the [combined DTC](#) in consideration of its related [event memory entries](#).

[Table 7.1](#) shows an example of a Dem configuration table including [combined DTCs](#). Several events are mapped to the same DTC. The combination on storage is represented by the DTC[1]. The DTC[2] shows an example of the combination on retrieval. The freeze frame data is stored individual for each event.

<b>Unique EventId</b>	<b>Event status</b>	<b>DTC status</b>	<b>Assigned DTC</b>	<b>Freeze frame</b>	....
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	S1 S2 S3	DTC[2]	FF[74]	
Event[5]	S5	S1 S2 S3	DTC[2]	FF[77]	
Event[6]	S6	S1 S2 S3	DTC[2]	FF[75]	
Event[7]	S7	S7	DTC[3]	FF[89]	
Event[8]	S8	S8	DTC[4]	FF[67]	
....	....	....	....	....	....

**Table 7.1: Example of a Dem configuration table including combined DTCs**

**[SWS\_Dem\_00441]** [ The Dem module shall implement the status bit calculations for the combined DTC status byte according to [Table 7.2](#). ]()

<b>DTC status bit description</b>		<b>Combined DTC status information logical equation</b>
0	TestFailed	$CbDTC_{Bit0} = Event[1]_{Bit0} \mid Event[2]_{Bit0} \mid \dots \mid Event[n]_{Bit0}$
1	TestFailedThis OperationCycle	$CbDTC_{Bit1} = Event[1]_{Bit1} \mid Event[2]_{Bit1} \mid \dots \mid Event[n]_{Bit1}$
2	PendingDTC	$CbDTC_{Bit2} = Event[1]_{Bit2} \mid Event[2]_{Bit2} \mid \dots \mid Event[n]_{Bit2}$
3	ConfirmedDTC	$CbDTC_{Bit3} = Event[1]_{Bit3} \mid Event[2]_{Bit3} \mid \dots \mid Event[n]_{Bit3}$
4	TestNotCompleted SinceLastClear	$CbDTC_{Bit4} = ( Event[1]_{Bit4} \mid Event[2]_{Bit4} \mid \dots \mid Event[n]_{Bit4} ) \& ! CbDTC_{Bit5}$
5	TestFailedSince LastClear	$CbDTC_{Bit5} = Event[1]_{Bit5} \mid Event[2]_{Bit5} \mid \dots \mid Event[n]_{Bit5}$
6	TestNotCompleted ThisOperationCycle	$CbDTC_{Bit6} = ( Event[1]_{Bit6} \mid Event[2]_{Bit6} \mid \dots \mid Event[n]_{Bit6} ) \& ! CbDTC_{Bit1} CbDTC$
7	WarningIndicator Requested	$CbDTC_{Bit7} = Event[1]_{Bit7} \mid Event[2]_{Bit7} \mid \dots \mid Event[n]_{Bit7}$

**Table 7.2: Calculation of combined 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 [\[SWS\\_Dem\\_00284\]](#), [\[SWS\\_Dem\\_00986\]](#), [\[SWS\\_Dem\\_00987\]](#) and [\[SWS\\_Dem\\_00828\]](#). ]()

**[SWS\_Dem\_00672]** [ The fault detection counter of the [combined DTC](#) shall be the maximum fault detection counter value of the sub-events. ]()

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

**[constr\_6103]** [ In case the [event combination](#) is disabled, it is not allowed to reference from multiple events to the same dtc. ]()

### 7.6.5.1 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). ]([SRS\\_Diag\\_04073](#))

**[SWS\_Dem\_01051]** [ For combination on storage, the event which allocated (or re-allocate) the [event memory entry](#) shall be the only event, which has the confirmed bit set. ]([SRS\\_Diag\\_04073](#))

**[SWS\_Dem\_01052]** [ For combination on storage, an event that cannot set the confirmed bit due to [\[SWS\\_Dem\\_01051\]](#) shall still generate warningIndicatorOnCriteria according to [\[SWS\\_Dem\\_00501\]](#). ]([SRS\\_Diag\\_04073](#))

**[SWS\_Dem\_01053]** [ If the Dem module is requested to support combination on storage, the [aging](#) shall be calculated based on the [combined DTC](#) status (refer to [\[SWS\\_Dem\\_00408\]](#)). ]([SRS\\_Diag\\_04073](#))

**[SWS\_Dem\_00442]** [ 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 [\[SWS\\_Dem\\_00498\]](#). ]([SRS\\_Diag\\_04073](#), [SRS\\_Diag\\_04133](#))

**[SWS\_Dem\_00443]** [ 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 [\[SWS\\_Dem\\_01186\]](#). ]([SRS\\_Diag\\_04118](#))

### 7.6.5.2 Combination On Retrieval

The section below describes the behavior of combination on retrieval.

**[SWS\_Dem\_00539]** [ If the Dem module is requested to support combination on retrieval , the status bit transition of each event shall trigger the collection, update

and storage of the related data (freeze frame / extended data). ](SRS\_Diag\_04073, SRS\_Diag\_04096)

**[SWS\_Dem\_00540]** [ If the Dem module is requested to report data of a combined DTC (combination on retrieval ), the Dem module shall return the event related data of all assigned events. ]()

**[SWS\_Dem\_00541]** [ Aging of each event of a combined DTC (combination on retrieval ) shall be processed individually according to [SWS\_Dem\_00493] and [SWS\_Dem\_00498]. ](SRS\_Diag\_04073)

**[SWS\_Dem\_00542]** [ For combination on retrieval, the displacement algorithm for each event of a combined DTC shall be treated individually (refer to [SWS\_Dem\_00408]). ]()

## 7.6.6 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\_ReportErrorStatus and 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.

**[SWS\_Dem\_00202]** [ If the Dem module is requested to support enable conditions, the Dem module shall provide the API Dem\_SetEnableCondition (refer to chapter 8.3.3.18) receiving the current status (condition fulfilled or not) of a specific enable condition. ](SRS\_Diag\_04095)

**[SWS\_Dem\_00446]** [ If the Dem module is requested to support enable conditions, the Dem module shall provide the ability to assign one enable condition group (refer to [DemEnableConditionGroup](#)), which includes one or several enable conditions (refer to [DemEnableCondition](#)) to a specific event. ] ([SRS\\_Diag\\_04095](#))

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]** [ 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 APIs [Dem\\_ReportErrorStatus](#) and [Dem\\_SetEventStatus](#)). ] ([SRS\\_Diag\\_04095](#))

**[SWS\_Dem\_00449]** [ If one enable condition is not fulfilled, all status reports from SWCs ([Dem\\_SetEventStatus](#)) and BSW modules ([Dem\\_ReportErrorStatus](#)) for those events being assigned to this condition shall be ignored (no change of UDS DTC status byte) by the Dem. ] ([SRS\\_Diag\\_04095](#))

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]** [ If all event-specific enable conditions are fulfilled, all status reports from SW-Cs ([Dem\\_SetEventStatus](#)) and BSW modules ([Dem\\_ReportErrorStatus](#)) for those events being assigned to these conditions shall be accepted by the Dem from this point in time on. ] ([SRS\\_Diag\\_04095](#))

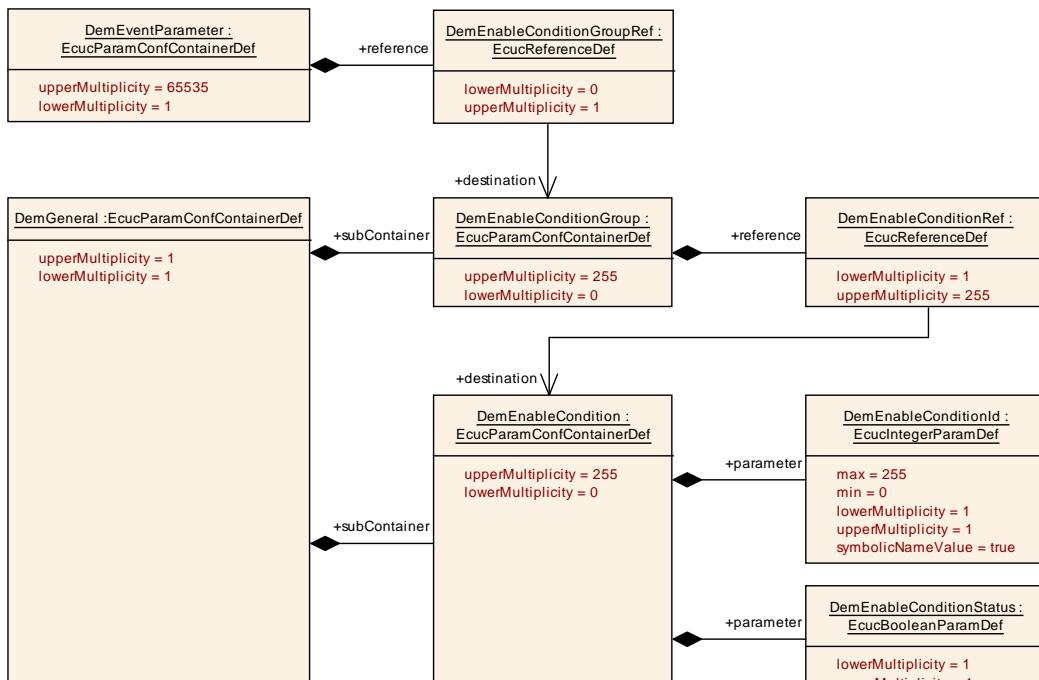


Figure 7.36: Enable condition assignment configuration

The following requirements introduce the handling of the storage conditions.

**[SWS\_Dem\_00543]** [ If the Dem module is requested to support storage conditions, the Dem module shall provide the API [Dem\\_SetStorageCondition](#) (refer to chapter 8.3.3.19) receiving the current status (condition fulfilled or not) of a specific storage condition. ]([SRS\\_Diag\\_04095](#))

**[SWS\_Dem\_00453]** [ If the Dem module is requested to support storage conditions, the Dem module shall provide the ability to assign one storage condition group (refer to [DemStorageConditionGroup](#)), which includes one or several storage conditions (refer to [DemStorageConditionGroup](#)) to a specific event. ]([SRS\\_Diag\\_04095](#))

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]** [ If the Dem module is requested to support storage conditions, the Dem module shall check the assigned storage conditions after the event gets qualified as failed (UDS DTC status bit 0 changes from 0 to 1). ]([SRS\\_Diag\\_04095](#))

**[SWS\_Dem\_00458]** [ If one storage condition is not fulfilled and no respective [event memory entry](#) exists, the Dem module shall not enter the reported event into the [event memory](#). ]([SRS\\_Diag\\_04095](#))

**[SWS\_Dem\_00591]** [ If one storage condition is not fulfilled and a respective [event memory entry](#) exists, the Dem module shall not update the [event memory](#) of the reported event. ]([SRS\\_Diag\\_04095](#))

**[SWS\_Dem\_00459]** [ If all event-specific storage conditions are fulfilled, the Dem module shall permit the storage of the reported event. ]([SRS\\_Diag\\_04095](#))

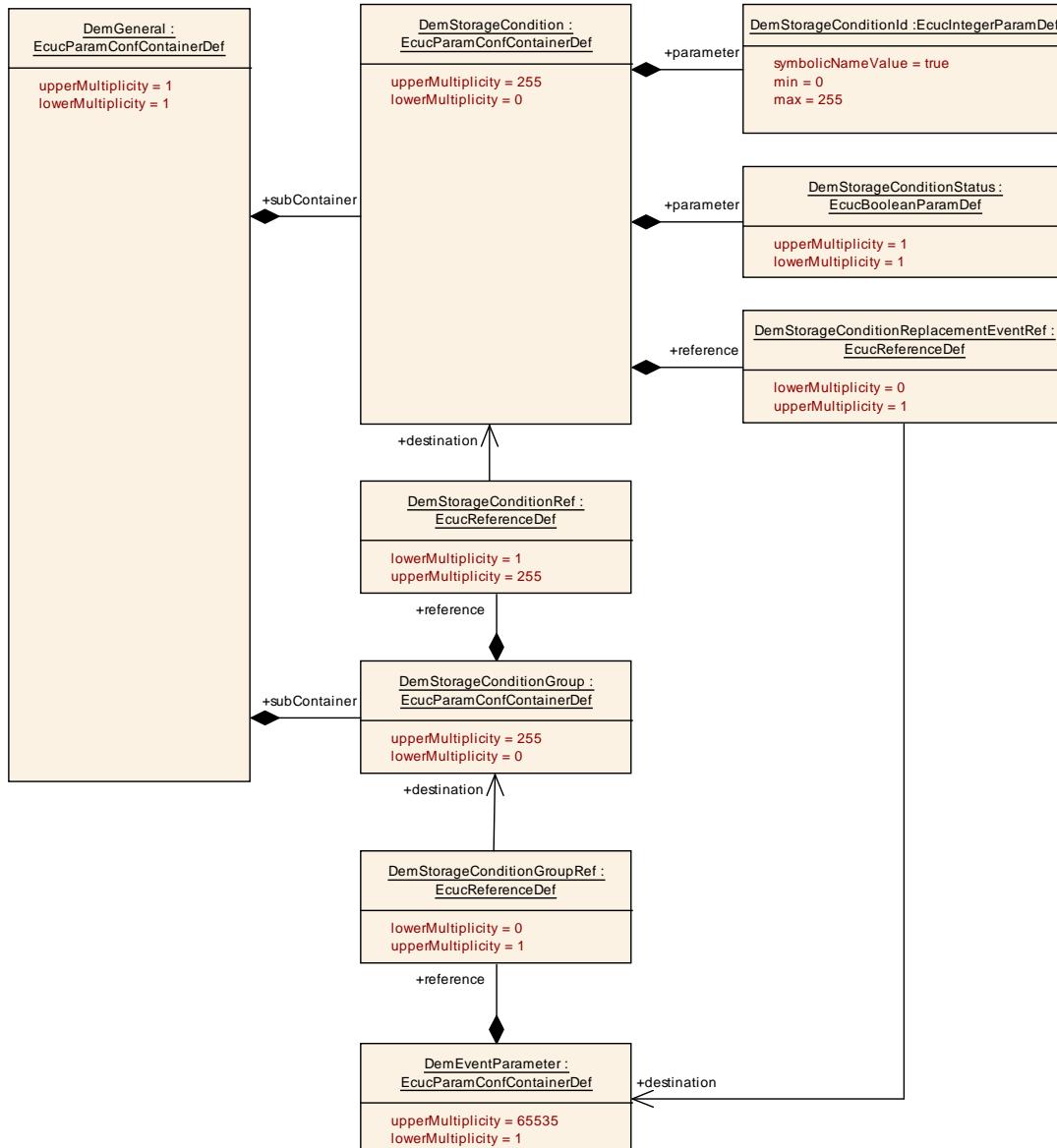


Figure 7.37: Storage condition assignment configuration

**[SWS\_Dem\_01085]** [ For each storage condition a replacement event may be defined (see [DemStorageConditionReplacementEventRef](#)). This replacement event will be used as a replacement failure info. ] ([SRS\\_Diag\\_04137](#))

**[SWS\_Dem\_01086]** [ When a storage condition is deactivated, the replacement event shall be set (status FAILED) as soon as a failure is reported and filtered by the storage condition (storage condition not fulfilled). 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. ] ([SRS\\_Diag\\_04137](#))

**[SWS\_Dem\_01087]** [ The replacement error has an EventStatus ([Dem\\_UdsStatusByteType](#)) as every other event. ] ([SRS\\_Diag\\_04137](#))

**[SWS\_Dem\_01088]** [ The status/behavior of the filtered event (filtered by the storage-condition) is not affected by the replacement event (meaning: the failure which is fil-

tered 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).  
]([SRS\\_Diag\\_04137](#))

**[SWS\_Dem\_01089]** [ The replacement event is reset (status PASSED), when the storage condition is fulfilled (on enabling the storage condition). ]([SRS\\_Diag\\_04137](#))

**[SWS\_Dem\_01090]** [ constraints for the replacement event:

- must not be mapped to an storage condition
- must not be configured to use debouncing
- must not have an InitMForE callback/ clearEventAllowed
- must not be configured for enableConditions
- must be configured as [DEM\\_EVENT\\_KIND\\_BSW](#)

]([SRS\\_Diag\\_04137](#))

**[SWS\_Dem\_01091]** [ Dem\_ReportErrorStatus shall ignore the replacement event.  
]([SRS\\_Diag\\_04137](#))

### 7.6.7 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[1] 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 according to chapter [7.6.7.4](#)). ]([SRS\\_Diag\\_04074](#))

Note: The presence of a Confirmed DTC 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 [7.6.7.4](#)).

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.

### 7.6.7.1 Storage of freeze frame data

**[SWS\_Dem\_00039]** [ The Dem module shall support event-specific freeze frame storage. ](SRS\_Diag\_04074)

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](#)). ](SRS\_Diag\_04074)

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]** [ If the Dem module uses calculated record numbers (refer to [DemTypeOfFreezeFrameRecordNumeration](#)), the Dem module shall be capable to store the configured number of freeze frames (refer to [DemMaxNumberFreezeFrameRecords](#)). ](SRS\_Diag\_04127)

**[SWS\_Dem\_00581]** [ 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. ](SRS\_Diag\_04127)

**[SWS\_Dem\_00582]** [ If the Dem module uses dedicated, configured record numbers (refer to [DemTypeOfFreezeFrameRecordNumeration](#)), the Dem module shall be capable to store per [event memory entry](#) all configured freeze frame records for this particular event (refer to [DemFreezeFrameRecNumClassRef](#) ). ](SRS\_Diag\_04074)

**[SWS\_Dem\_00797]** [ 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](#). ](SRS\_Diag\_04127)

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]** [ 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. ](SRS\_Diag\_04127)

**[SWS\_Dem\_00799]** [ 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 DemEventMemoryEntryFdcThresholdStorageValue) is reached (by a positive increment), but at most once per operation cycle. ](SRS\_Diag\_04127)

**[SWS\_Dem\_01068]** [ 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. ](SRS\_Diag\_04127)

**[SWS\_Dem\_00800]** [ 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 TestFailed UDS DTC status bit 0 is set (changing from 0 to 1). ](SRS\_Diag\_04127)

**[SWS\_Dem\_00801]** [ If the DemFreezeFrameRecordTrigger is set to DEM\_TRIGGER\_ON\_PENDING, the FreezeFrame shall be captured (as allowed by [SWS\_Dem\_00797]) each time the Pending UDS DTC status bit 2 is set (changing from 0 to 1). ](SRS\_Diag\_04127)

**[SWS\_Dem\_00802]** [ 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 DTC status bit 3 is set (changing from 0 to 1). ](SRS\_Diag\_04127)

**[SWS\_Dem\_00803]** [ 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. ](SRS\_Diag\_04127)

**[SWS\_Dem\_00804]** [ 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). ](SRS\_Diag\_04127)

**[SWS\_Dem\_00461]** [ If the configuration parameter DemEnvironmentDataCapture (refer to DemGeneral) is set to DEM\_CAPTURE\_ASYNC\_TO\_REPORTING, event-specific freeze frame data shall be captured latest in the next cycle of the Dem\_MainFunction. ](SRS\_Diag\_04127)

**[SWS\_Dem\_00805]** [ If the configuration parameter DemEnvironmentDataCapture (refer to DemGeneral) is set to DEM\_CAPTURE\_SYNCHRONOUS\_TO\_REPORTING, event-specific freeze

`frame` data shall be captured within the reporting function (i.e. in the context of the SetEventStatus/ReportErrorStatus function). ](SRS\_Diag\_04127)

Note: [SWS\_Dem\_00461] and [SWS\_Dem\_00805] does not apply if a pre-stored `freeze frame` is available; see also SWS\_Dem\_00464.

[SWS\_Dem\_01081] [ If the configuration parameter `DemEnvironmentDataCapture` (refer to DemGeneral) is set to `DEM_CAPTURE_ASYNCNCHRONOUS_TO_REPORTING`, event-specific extended data shall be captured latest in the next cycle of the Dem\_MainFunction. ](SRS\_Diag\_04127)

[SWS\_Dem\_01082] [ If the configuration parameter `DemEnvironmentDataCapture` (refer to DemGeneral) is set to `DEM_CAPTURE_SYNCHRONOUS_TO_REPORTING`, event-specific extended data shall be captured within the reporting function (i.e. in the context of the SetEventStatus/ReportErrorStatus function). ](SRS\_Diag\_04127)

If `DemEnvironmentDataCapture` is equal to `DEM_CAPTURE_SYNCHRONOUS_TO_REPORTING`, a queue could be used to store the environment data until the data is transferred to the `event memory`. To not loose any reporting, the queue size needs to be configured to store all reporting between two cycles of Dem\_MainFunction.

[SWS\_Dem\_01083] [ If `DemEnvironmentDataCapture` is equal to `DEM_CAPTURE_SYNCHRONOUS_TO_REPORTING`, the Dem module shall queue the environmental data from the reporting (see [SWS\_Dem\_00805] and [SWS\_Dem\_01082]) till the next call of the Dem\_MainFunction. The Dem shall at minimum queue the configured size (refer `DemMaxNumberEventEntryEventBuffer`). In case the queue is full, the reporting function shall return E\_NOT\_OK (SetEventStatus) or ignore the reporting (ReportErrorStatus). ](SRS\_Diag\_04127)

[SWS\_Dem\_00261] [ The Dem module shall use the C-callback `ReadDataElement` (refer to chapter 8.4.3.3.4) respectively the operation `ReadData` of the interface `DataServices_<Data>` (refer to chapter 8.6.2.13) to collect all configured data elements of the respective `freeze frame`. ]()

[SWS\_Dem\_00806] [ The Dem module may invoke the data collection according [SWS\_Dem\_00261] from Dem\_MainFunction and not within the reporting functions. Note: To ensure a synchronous data collection the prestore freeze frame feature should be used. ](SRS\_Diag\_04127)

[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, reports the development error `DEM_E_NODATAAVAILABLE` to the Det and continues its normal operation. ](SRS\_Diag\_04085)

[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. ]([SRS\\_Diag\\_04074](#))

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

#### 7.6.7.2 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]** [ The Dem module shall provide the configuration parameter [DemFFPrestorageSupported](#) (refer to [DemEventParameter](#)) to enable or disable pre-storage handling of freeze frames per event. ]([SRS\\_Diag\\_04127](#))

Note: If [DemMaxNumberPrestoredFF](#) (refer to [DemGeneral](#)) is set to 0, [DemFFPrestorageSupported](#) can not be enabled for any DTC.

**[SWS\_Dem\_00334]** [ 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. ]([SRS\\_Diag\\_04127](#))

**[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 DTC status bit changes.]([SRS\\_Diag\\_04074](#))

Note: If a request to restore a [freeze frame](#) ([Dem\\_PrestoreFreezeFrame](#)) is interrupted by reporting an event status ([Dem\\_SetEventStatus](#)/[Dem\\_ReportErrorStatus](#)) or a call to [Dem\\_ClearPrestoredFreezeFrame](#) of the same EventId the detailed behavior is implementation specific. The [freeze frame](#) which is requested to be restored might be used immediately or might be stored and used with next upcoming event report or vice versa.

**[SWS\_Dem\_00807]** [ The capture (sampling) of data (using [\[SWS\\_Dem\\_00261\]](#)) shall be synchronous in the call of [Dem\\_PrestoreFreezeFrame](#). ]([SRS\\_Diag\\_04074](#))

**[SWS\_Dem\_00808]** [ The API [Dem\\_PrestoreFreezeFrame](#) shall return E\_NOT\_OK if no memory is available (see [DemMaxNumberPrestoredFF](#)). ]([SRS\\_Diag\\_04127](#))

**[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 data is captured (refer to [\[SWS\\_Dem\\_00461\]](#). ]([SRS\\_Diag\\_04074](#))

**[SWS\_Dem\_00969]** [ The pre-stored freeze frame shall be released after the data event has been stored (event retention was successfull) or discarded (event retention failed). ](SRS\_Diag\_04074)

**[SWS\_Dem\_00191]** [ If no pre-stored freeze frame is available, the Dem module shall behave according to chapter 7.6.7.1 Storage of freeze frame data. ](SRS\_Diag\_04074)

Note: The captured data while using pre-stored freeze frames can differ from the data, which is collected using the UDS DTC 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.

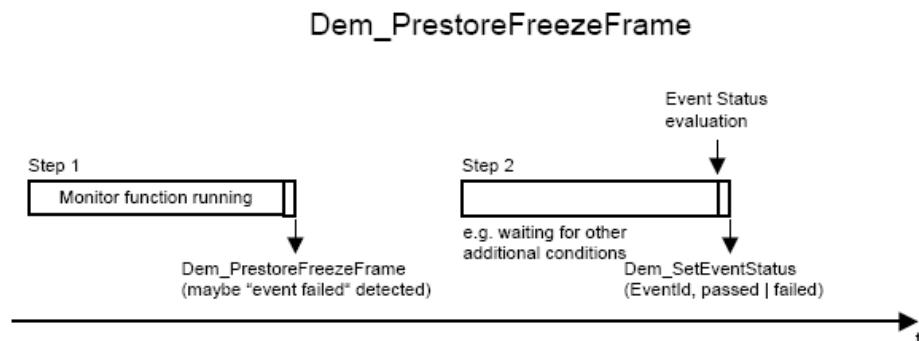


Figure 7.38: Example to use `Dem_PrestoreFreezeFrame` to prestore freeze frame data

**[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 DTC status bit 0 changes from 1 to 0) the Dem module shall release the pre-stored freeze frame for the specific event. ](SRS\_Diag\_04074)

### 7.6.7.3 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]** [ 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`. ](SRS\_Diag\_04127)

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]** [ 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. ]([SRS\\_Diag\\_04127](#))

**[SWS\_Dem\_00811]** [ 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 [DemEventMemoryEntryFdcThresholdStorageValue](#) is reached (by a positive increment), but at most once per operation cycle. ]([SRS\\_Diag\\_04127](#))

**[SWS\_Dem\_01069]** [ 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. ]([SRS\\_Diag\\_04127](#))

**[SWS\_Dem\_00812]** [ 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 TestFailed UDS DTC status bit 0 is set (changing from 0 to 1). ]([SRS\\_Diag\\_04127](#))

**[SWS\_Dem\_00813]** [ If the [DemExtendedDataRecordTrigger](#) is set to [DEM\\_TRIGGER\\_ON\\_PENDING](#), the ExtendedDataRecord shall be captured (as allowed by [\[SWS\\_Dem\\_00810\]](#)) each time the Pending UDS DTC status bit 2 is set (changing from 0 to 1). ]([SRS\\_Diag\\_04127](#))

**[SWS\_Dem\_00814]** [ If the [DemExtendedDataRecordTrigger](#) is set to [DEM\\_TRIGGER\\_ON\\_CONFIRMED](#), the ExtendedDataRecord shall be captured (as allowed by [\[SWS\\_Dem\\_00810\]](#)) each time the Confirmed UDS DTC status bit 3 is set (changing from 0 to 1). ]([SRS\\_Diag\\_04127](#))

**[SWS\_Dem\_01070]** [ 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. ]([SRS\\_Diag\\_04127](#))

**[SWS\_Dem\_01071]** [ If the [DemExtendedDataRecordTrigger](#) is set to [DEM\\_TRIGGER\\_ON\\_MIRROR](#), the ExtendedDataRecord shall be captured with the transfer of the memory entry to the mirror memory. ]([SRS\\_Diag\\_04127](#))

**[constr\_6101]** [ DemExtendedDataRecordTrigger needs to be configured. [DemExtendedDataRecordTrigger](#) shall always be configured, except for internal data elements like occurrence counters. ]()

**[SWS\_Dem\_00815]** [ 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](#). ]([SRS\\_Diag\\_04127](#))

**[SWS\_Dem\_00816]** [ If the configuration parameter [DemExtendedDataRecordUpdate](#) is set to [DEM\\_UPDATE\\_RECORD\\_YES](#), the ExtendedDataRecord shall be updated with each trigger (refer to [DemExtendedDataRecordTrigger](#)). ]([SRS\\_Diag\\_04127](#))

**[SWS\_Dem\_00282]** [ The [Dem](#) module shall use the C-callback [ReadDataElement](#) respectively the operation [ReadData](#) of the interface [DataServices\\_<Data>](#) 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]** [ 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](#). ]()

#### 7.6.7.4 Configuration of Event related data

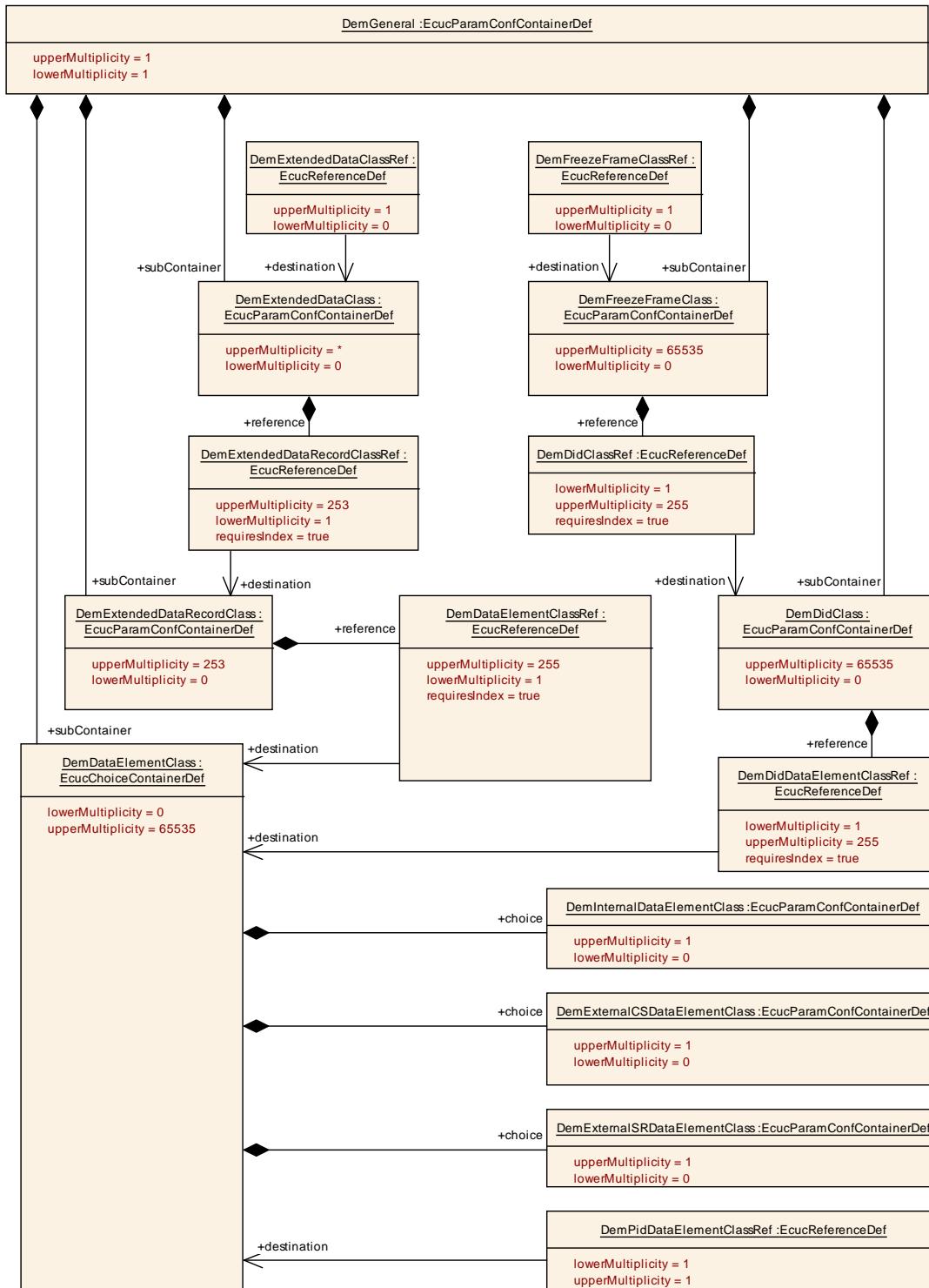
This section describes the configuration of event related data and the access of event related data from the SW-Cs/BSW modules.

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.



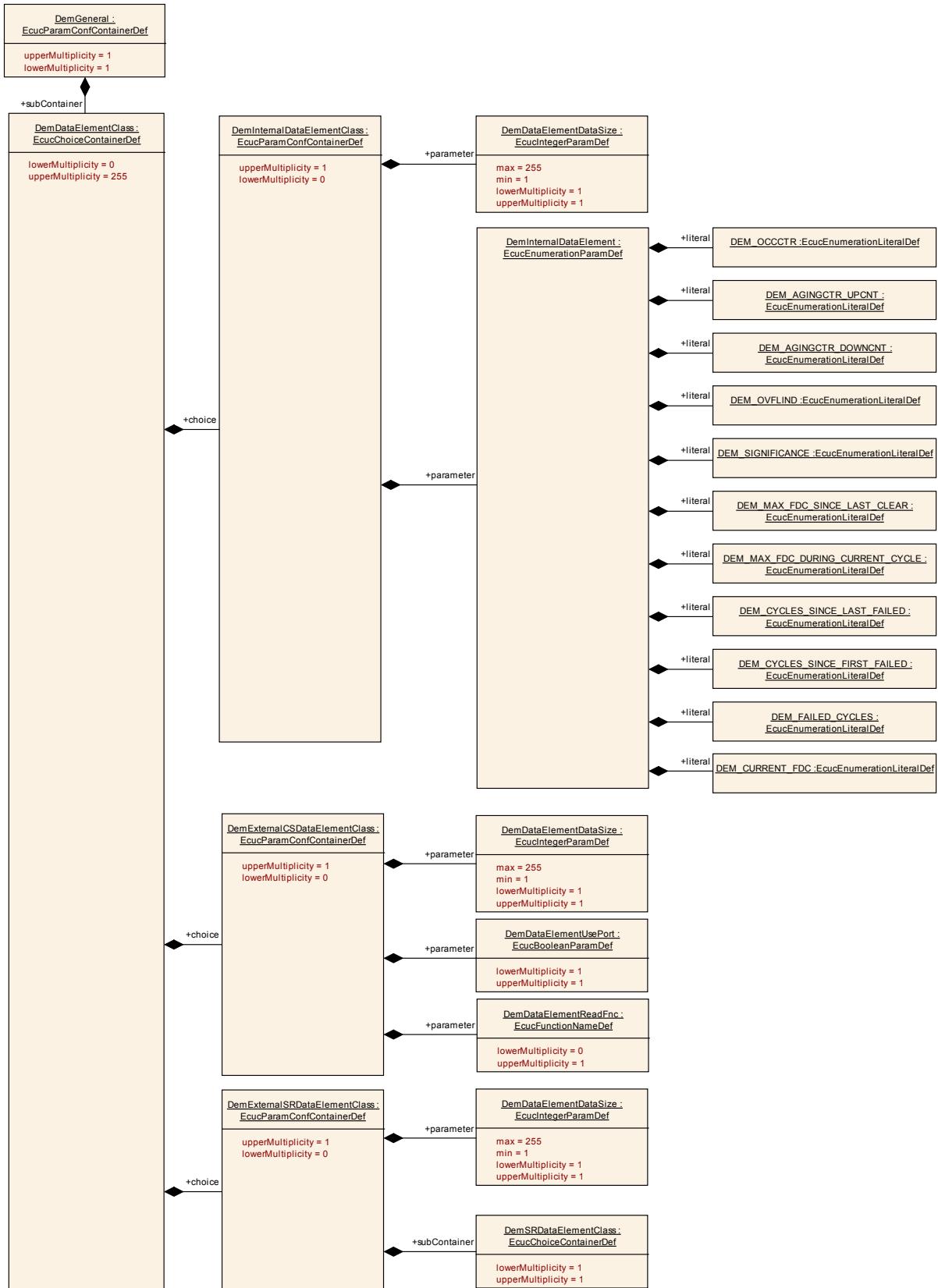
**Figure 7.39: Event related data configuration**

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 [DemExternalCSDDataElementClass](#) and [DemExternalSRDataEle-](#)

**mentClass.** 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 7.40: Data element configuration**

**[SWS\_Dem\_00469]** [ 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. ] (*SRS\_Diag\_04104, SRS\_Diag\_04134*)

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]** [ Internal data elements (refer to *DemInternalDataElementClass*) shall not be stored, but the current value shall be used instead. ] (*SRS\_Diag\_04127, SRS\_Diag\_04134*)

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\_00471]** [ If the configuration parameter *DemInternalDataElement* is set to *DEM\_OCCCTR*, then the Dem-internal value of the occurrence counter (refer to chapter 7.3.2) shall be mapped to the respective data element. ] (*SRS\_Diag\_04127, SRS\_Diag\_04134*)

**[SWS\_Dem\_00472]** [ 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* (refer to chapter 7.6.9) shall be mapped to the respective data element (based on **[SWS\_Dem\_00643]** or **[SWS\_Dem\_00673]**). ] (*SRS\_Diag\_04134*)

**[SWS\_Dem\_00643]** [ If the data element *DEM\_AGINGCTR\_UPCNT* is configured, the *aging counter* mapping shall be based on a count-up mechanism from 0 to *DemAgingCycleCounterThreshold* (refer to ISO-14229-1[1], Annex D). ] (*SRS\_Diag\_04068, SRS\_Diag\_04134*)

**[SWS\_Dem\_00673]** [ If the data element *DEM\_AGINGCTR\_DOWNCNT* is configured, the *aging counter* mapping shall be based on a count-down mechanism from *DemAgingCycleCounterThreshold* to 0 (refer to ISO-14229-1[1], Annex D). ] (*SRS\_Diag\_04076, SRS\_Diag\_04134*)

**[SWS\_Dem\_01043]** [ If *DemAgingAllowed* is set to ‘false’ the internal data element *DEM\_AGINGCTR\_DOWNCNT* shall be *DemAgingCycleCounterThreshold* if configured or ‘255’. ] (*SRS\_Diag\_04134*)

**[SWS\_Dem\_01044]** [ If Dem *DemAgingAllowed* is set to ‘false’ the internal data element *DEM\_AGINGCTR\_UPCNT* shall be ‘0’. ] (*SRS\_Diag\_04134*)

**[SWS\_Dem\_00473]** [ If the configuration parameter *DemInternalDataElement* is set to *DEM\_OVFLIND*, then the Dem-internal value of the overflow indication (refer to chapter 7.6.2.3) shall be mapped to the respective data element as boolean (0 = False, 1 = True). ] (*SRS\_Diag\_04134*)

**[SWS\_Dem\_00592]** [ If the configuration parameter [DemInternalDataElement](#) is set to [DEM\\_SIGNIFICANCE](#), then the (static) Dem-internal value of the DTC significance (refer to chapter [7.4.6](#)) shall be mapped to the respective data element with 0 = OC-CURRENCE and 1 = FAULT. ]([SRS\\_Diag\\_04134](#))

**[SWS\_Dem\_01084]** [ If the configuration parameter [DemInternalDataElement](#) is set to [DEM\\_CURRENT\\_FDC](#), then the Dem-internal value of the current fault detection counter (refer to [subsubsection 7.6.3.4](#) and [subsubsection 7.6.3.5](#)) shall be mapped to the respective data element with the following value-translation:  
[0..127]->[0x00..0x7F] and [-1..-128]->[0xFF..0x80]. ]([SRS\\_Diag\\_04134](#))

**[SWS\_Dem\_00818]** [ 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 since last clear (refer to chapter [7.6.3.7](#)) shall be mapped to the respective data element. ]([SRS\\_Diag\\_04068](#), [SRS\\_Diag\\_04134](#))

**[SWS\_Dem\_00819]** [ 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 (refer to chapter [7.6.3.7](#)) shall be mapped to the respective data element. ]([SRS\\_Diag\\_04127](#), [SRS\\_Diag\\_04134](#))

**[SWS\_Dem\_00820]** [ 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 (refer to chapter [7.6.9.1](#)) shall be mapped to the respective data element. ]([SRS\\_Diag\\_04127](#), [SRS\\_Diag\\_04134](#))

**[SWS\_Dem\_00821]** [ 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 (refer to chapter [7.6.9.2](#)) shall be mapped to the respective data element. ]([SRS\\_Diag\\_04104](#), [SRS\\_Diag\\_04134](#))

**[SWS\_Dem\_00822]** [ If the configuration parameter [DemInternalDataElement](#) is set to [DEM\\_FAILED\\_CYCLES](#), then the Dem-internal value of the failed operation cycle counter (refer to chapter [7.6.9.3](#)) shall be mapped to the respective data element. ]([SRS\\_Diag\\_04079](#), [SRS\\_Diag\\_04134](#))

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\_00470]** [ If the Dem module implements customer-specific Dem-internal data elements, the configuration parameter [DemInternalDataElement](#) shall be extended with the respective enumeration values. ]([SRS\\_Diag\\_04134](#))

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]** [ If the configuration parameter [DemInternalDataElement](#) is set to [DEM\\_CURRENT\\_FDC](#), then the fault detection counter (refer to chapter [7.6.3](#)) shall be mapped to the respective data element. ]([SRS\\_Diag\\_04134](#))

**[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 `Dem_DataEndianness` 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 defined in [17] Chapter 2.4 to signed data types. ]()

Note: In [17] Chapter 2.4 defines the endianness conversion for unsigned data types.

#### 7.6.7.5 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 [8.4.3.3.2 EventDataChanged](#)).

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 `DemEventParameter`) is used to specify the related port or c-callback per event.

**[SWS\_Dem\_00475]** [ If 'event related data' of an '[event memory entry](#)' is added or updated (triggering the collection of new 'event related data') AND notifications on data changes are configured via '[DemCallbackEventDataChanged](#)' AND the `DemDataElementClass` of the 'event related data' is of type '[DemExternalCSDDataElementClass](#)' or '[DemExternalSRDataElementClass](#)', the [Dem](#) shall trigger these configured event-specific notifications as well as the general notification '[GeneralCallbackEventDataChanged](#)'. ]()

Note: Deleted, displaced or aged '[event memory entries](#)' do not trigger `EventDataChanged` callbacks.

Note: 'event related data' of type [DemInternalDataElementClass](#) do not trigger `EventDataChanged` 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` ([DemMemoryDestinationRef](#)).

**[SWS\_Dem\_00479]** [ 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 [DemTypeOfFreezeFrameRecordNumeration](#) is set to [DEM\\_FF\\_RECNUM\\_CALCULATED](#) the most recent record shall be used, otherwise E\_NOT\_OK shall be returned. ]([SRS\\_Diag\\_04024](#))

**[SWS\_Dem\_01194]** [ 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\\_E\\_WRONG\\_RECORDNUMBER](#) instead (refer to [DemOBDSupport](#)). ]([SRS\\_Diag\\_04024](#))

**[SWS\_Dem\_00991]** [ 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. ]([SRS\\_Diag\\_04127](#))

**[SWS\_Dem\_00477]** [ The function [Dem\\_GetEventExtendedDataRecordEx](#) shall report the data of the extended data record of the requested diagnostic event. ]([SRS\\_Diag\\_04102](#))

**[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\\_DcmGetExtendedDataRecordByDTC](#) instead of the function [Dem\\_GetEventExtendedDataRecordEx](#). The Dlt uses the function [Dem\\_DLtGetAllExtendedDataRecords](#) instead.

**[SWS\_Dem\_00995]** [ If the interfaces [Dem\\_GetEventFreezeFrameDataEx](#) and [Dem\\_GetEventExtendedDataRecordEx](#) are called in the context of [GeneralCallback-EventDataChanged](#) or [CallbackEventDataChanged](#), the data of the triggering event shall be retrievable (return value E\_NOT\_OK is not allowed). ]([SRS\\_Diag\\_04074](#), [SRS\\_Diag\\_04104](#))

**[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). ]([SRS\\_Diag\\_04074](#))

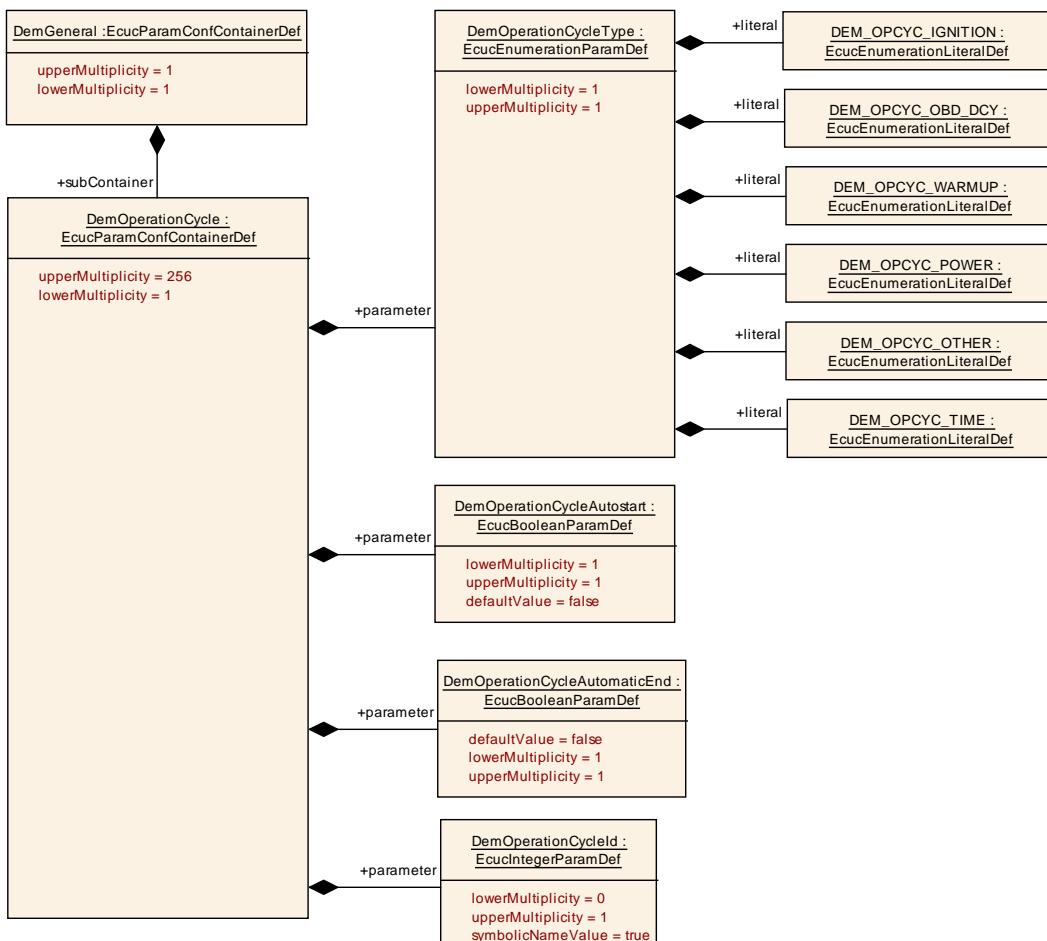
**[SWS\_Dem\_00997]** [ [Dem\\_GetEventExtendedDataRecordEx](#) shall return E\_NOT\_OK if the requested data is currently not accessible (e.g. in case of asynchronous preempted data retrieval from application). ]([SRS\\_Diag\\_04074](#))

## 7.6.8 Operation cycle management

The Dem module uses different operation cycles (ref. to ISO-14229-1[1]). Those cycles could either be provided by other BSW modules and SW-C or generated by the Dem module itself.

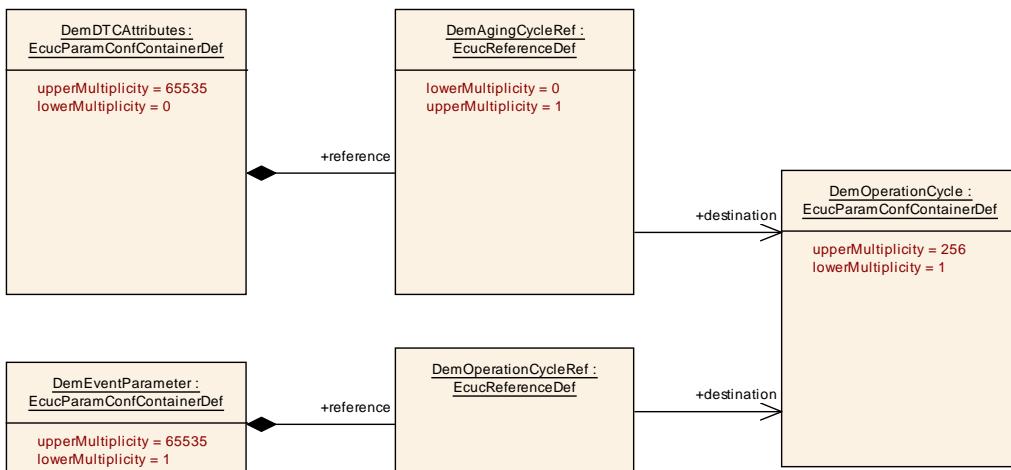
Examples of operation cycles are: -ignition on/off cycle - power up/power down cycle - OBD driving cycle - engine warm up cycle - operation active/passive cycle - accumulated operating time

The Dem operation cycle management processes these different types of operation cycle definitions to create Dem-specific operation cycle state information, or accepts cycle state changes via the API [Dem\\_SetOperationCycleState](#) (refer to chapter [8.3.3.8](#)). The OBD driving cycle ([DEM\\_OPCYC\\_OBD\\_DCY](#)) behaves differently than other driving cycles. Some requirements differ for “OBD” and “non-OBD” cycle.



**Figure 7.41: Operation cycle configuration**

**[SWS\_Dem\_00577]** 「 The Dem module shall provide the configuration parameter [DemOperationCycleStatusStorage](#) (refer to [DemGeneral](#)) to define if the operation cycle state shall be available over the power cycle (stored non-volatile) or not. 」([SRS\\_Diag\\_04076](#))



**Figure 7.42: Operation and aging cycle assignment**

**[SWS\_Dem\_00853]** [ For all operation cycles with [DemOperationCycleAutostart](#) set to true [Dem\\_Init](#) shall start the operation cycle (identical to [Dem\\_SetOperationCycleState](#) is called with parameter [DEM\\_CYCLE\\_STATE\\_START](#)) with the exception of executing all callbacks resulting from UDS status byte changes. ]()

Note: Each call [Dem\\_SetOperationCycleState](#) with parameter [DEM\\_CYCLE\\_STATE\\_START](#) would re-start the operation cycle, wherefore the start-up sequence should not explicitly start auto-started cycles.

Note: The [DemOperationCycleType](#) [DEM\\_OPCYC\\_POWER](#) should use the [DemOperationCycleAutostart](#) feature.

**[SWS\_Dem\_00481]** [ If an operation cycle has started, all status reports from SW-Cs ([Dem\\_SetEventStatus](#)) and BSW modules ([Dem\\_ReportErrorStatus](#)) for those events, being assigned to this cycle, shall be accepted by the Dem from this point in time on. ]([SRS\\_Diag\\_04076](#))

**[SWS\_Dem\_00699]** [ At the moment the OBD driving cycle is started, the Dem shall execute computations required to reach the confirmed states of Events collected during the “ended” phase of the cycle. ]([SRS\\_Diag\\_04123](#))

**[SWS\_Dem\_00482]** [ If an non-OBD operation cycle has ended, all status reports from SW-Cs ([Dem\\_SetEventStatus](#)) and BSW modules ([Dem\\_ReportErrorStatus](#)) for those events being assigned to this cycle shall be ignored (no change of UDS DTC status byte) by the Dem. ]([SRS\\_Diag\\_04076](#))

The operation cycle status has no impact on [Dem\\_ResetEventDebounceStatus](#) and [Dem\\_ResetEventStatus](#).

**[SWS\_Dem\_00700]** [ If the OBD driving cycle has ended, status reports shall be processed, except update of the confirmed status. ]([SRS\\_Diag\\_04067](#), [SRS\\_Diag\\_04129](#))

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

**[SWS\_Dem\_01184]** [ If the OBD driving cycle has ended and the Warm up conditions have been fulfilled (WUC cycle started) during this cycle the aging shall be processed. ]()

**[SWS\_Dem\_00338]** [ The operation cycle management of the Dem module shall use the reported state (DEM\_CYCLE\_STATE\_START / DEM\_CYCLE\_STATE\_END) of the API [Dem\\_SetOperationCycleState](#) (refer to chapter [8.3.3.8](#)) to set the Dem specific operation cycle state (started / ended). ]([SRS\\_Diag\\_04076](#))

Note: This API is called by the SW-Cs / BSW modules, as soon as those detect the status change of the monitored operation cycles. The part of this API, which will be handled asynchronously, is implementation-specific.

The operation cycle reporting can be Dem-internal for Dem module self-calculated operation cycles.

**[SWS\_Dem\_00697]** [ Dem shall provide a configuration parameter DemOperationCycleAutomaticEnd (refer to [ECUC\\_Dem\\_00837](#)) to automatically END the OBD driving cycle at initialization time. The implementation may either END the cycle either at the beginning of [Dem\\_Shutdown\(\)](#) or during [Dem\\_Init\(\)](#). If the cycle is ENDed at [Dem\\_Init\(\)](#), any event status change received before (after [Dem\\_PrelInit\(\)](#)) shall be processed after the cycle is ENDed, i.e., be considered for the new cycle. ]([SRS\\_Diag\\_04001](#))

Note: If the control unit is restarted (e.g., by turning the ignition key from the OFF to the ON position) before [Dem\\_Shutdown\(\)](#) is processed, the Dem will usually receive a END from the OBD driving cycle defining SW-C.

**[SWS\_Dem\_00483]** [ If the API [Dem\\_SetOperationCycleState](#) is called with DEM\_CYCLE\_STATE\_START and the respective operation cycle was already started, the operation cycle shall be restarted (started again). ]([SRS\\_Diag\\_04076](#))

Note: This will be the case, if the end- and start-criteria of an operation cycle is fulfilled at exactly the same point in time. Therefore, the caller of [Dem\\_SetOperationCycleState](#) needs only to report "start".

**[SWS\_Dem\_00484]** [ If the API [Dem\\_SetOperationCycleState](#) is called with DEM\_CYCLE\_STATE\_END and the respective operation cycle was already ended, the API shall perform no further action. ]([SRS\\_Diag\\_04076](#))

Note: The operation cycle state may not be started during [Dem\\_PrelInit](#) via interface [Dem\\_SetOperationCycleState](#). Also persistently stored cycle states cannot be ensured to be restored until end of initialization.

**[SWS\_Dem\_01078]** [ If [DemOperationCycleStatusStorage](#) is set to FALSE, a call of [Dem\\_Shutdown](#) shall end all operation cycles which are not stored in non-volatile memory. ]([SRS\\_Diag\\_04076](#))

Note: If Power cycle ([DEM\\_OPCYC\\_POWER](#)) should not be ended/stopped during "ECU Sleep" then it should be configured with [DemOperationCycleStatusStorage == TRUE](#).

## 7.6.9 Operation Cycle Counters

### 7.6.9.1 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]** [ 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. ]([SRS\\_Diag\\_04134](#))

**[SWS\_Dem\_00771]** [ If internal [debounce counter](#) reach [DemDebounceCounter-FailedThreshold](#) (latest UDS DTC status bit 0 changes from 0 to 1) and this counter is not stored in [event memory](#) and there are available [event memory entry](#), new entry shall be allocated and the counter shall be started and initialized to zero. ]()

**[SWS\_Dem\_00772]** [ If internal [debounce counter](#) reach [DemDebounceCounter-FailedThreshold](#) (latest UDS DTC status bit 0 changes from 0 to 1) and this counter is stored in [event memory](#) the counter shall initialized to zero. ]([SRS\\_Diag\\_04068](#))

**[SWS\_Dem\_00773]** [ In case the counter is available and started, it shall be incremented at the end of the referenced operation cycle (refer to [DemOperationCycleRef](#)). ]([SRS\\_Diag\\_04076](#))

**[SWS\_Dem\_00774]** [ The counter shall be implemented as one byte. If any count operation occurs which would cause a counter to roll over past 0xFF then the count value shall instead be maintained at 0xFF. ]([SRS\\_Diag\\_04068](#))

### 7.6.9.2 Cyles since first failed

The counter Cyles 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]** [ If the counter ‘Cyles 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. ]([SRS\\_Diag\\_04074](#), [SRS\\_Diag\\_04134](#))

**[SWS\_Dem\_00776]** [ If internal [debounce counter](#) reach [DemDebounceCounter-FailedThreshold](#) (latest UDS DTC status bit 0 changes from 0 to 1) and this counter is not stored in [event memory](#) and there are available [event memory entry](#), new entry shall be allocated and the counter shall be started and initialized to zero. ]([SRS\\_Diag\\_04125](#))

**[SWS\_Dem\_00777]** [ In case the counter is available and started, it shall be incremented at the end of the referenced operation cycle (refer to [DemOperationCycleRef](#)). ]()

**[SWS\_Dem\_00778]** [ The counter shall be implemented as one byte. If any count operation occurs which would cause a counter to roll over past 0xFF then the count value shall instead be maintained at 0xFF. ]([SRS\\_Diag\\_04068](#))

### 7.6.9.3 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]** [ 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. ]([SRS\\_Diag\\_04068](#), [SRS\\_Diag\\_04104](#), [SRS\\_Diag\\_04134](#))

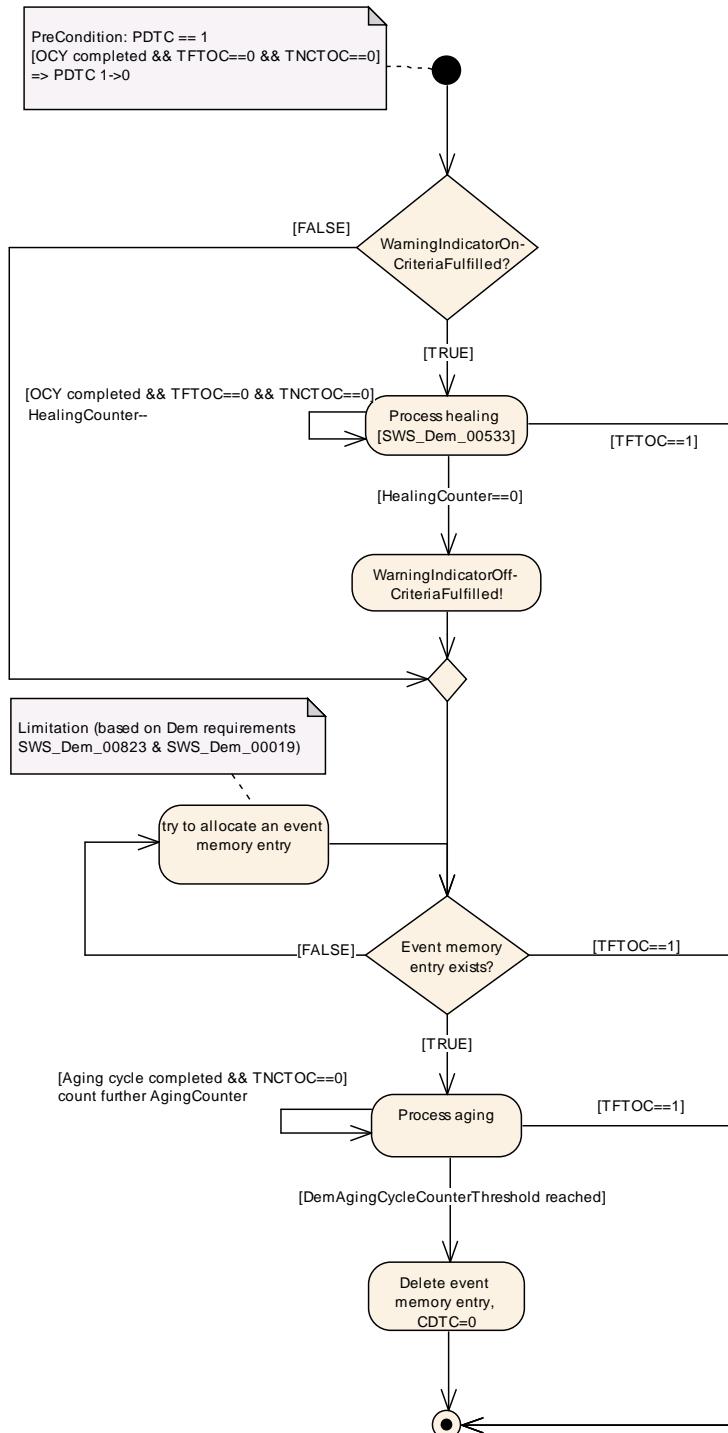
**[SWS\_Dem\_00780]** [ If internal [debounce counter](#) reach [DemDebounceCounter-FailedThreshold](#) (latest UDS DTC status bit 0 changes from 0 to 1) and this counter is not stored in [event memory](#) and there are available [event memory entry](#), new entry shall be allocated and the counter shall be started and initialized to zero. ]()

**[SWS\_Dem\_00781]** [ In case the counter is available and started, it shall be incremented at the end of the referenced operation cycle (refer to [DemOperationCycleRef](#)) in case the UDS DTC status bit 1 (TestFailedThisOperationCycle) is set to 1. ]()

**[SWS\_Dem\_00782]** [ The counter shall be implemented as one byte. If any count operation occurs which would cause a counter to roll over past 0xFF then the count value shall instead be maintained at 0xFF. ]([SRS\\_Diag\\_04124](#))

### 7.6.10 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 7.43: General diagnostic event deletion processing**

**[SWS\_Dem\_00698]** ┌ The process of [aging](#) (counting of [aging counter](#)) starts when healing is completed (WarningIndicatorRequested bit == 0 , refer to [Figure 7.43](#)).  
└([SRS\\_Diag\\_04133](#), [SRS\\_Diag\\_04076](#))

**[SWS\_Dem\_00019]** ┌ The Dem module shall support at least an [aging counter](#) for each [event memory entry](#). ┘([SRS\\_Diag\\_04133](#), [SRS\\_Diag\\_04068](#), [SRS\\_Diag\\_04076](#))

**[SWS\_Dem\_00985]** [ The `aging counter` shall be calculated based on the referenced aging/operation cycle (refer to configuration parameter `DemAgingCycleRef`), if `aging` is enabled (refer to `DemAgingAllowed`) for this event. ](*SRS\_Diag\_04133, SRS\_Diag\_04068, SRS\_Diag\_04076*)

**[SWS\_Dem\_00492]** [ 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.](*SRS\_Diag\_04133*)

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 7.6.7.3 Storage of extended data).

**[SWS\_Dem\_00493]** [ `DemAgingCycleCounterThreshold` or `DemAgingCycleCounterThresholdForTFSLC` (depending on which value is higher) defines the number of completed aging cycles, after which the `event memory entry` shall be deleted(aged) from the `event memory`. ](*SRS\_Diag\_04133*)

Note: For completed aging cycles definition see [[SWS\\_Dem\\_00489](#)]

**[SWS\_Dem\_00823]** [ If configuration parameter `DemResetConfirmedBitOnOverflow` is set to false and in case an event has UDS status bit 3 (ConfirmedDTC) 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]** [ 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 (ConfirmedDTC) set to 1 and UDS status bit 0 (TestFailed) set to 0. If such an event is found, the Dem shall allocate an `event memory` location to get an `aging counter`. ](*SRS\_Diag\_04133*)

Note: The prioritization which event is choosen is implementation specific.

**[SWS\_Dem\_00498]** [ Upon event `aging counter` reach threshold `DemAgingCycleCounterThreshold`, the UDS status bit 3 (ConfirmedDTC) shall be set to 0. ](*SRS\_Diag\_04133*)

Note: All other UDS status bits are not modified by `aging` of corresponding `event memory entry`.

**[SWS\_Dem\_00161]** [ The Dem module shall handle the reoccurrence of unlearned events like new events, since they were previously deleted from the `event memory` by `aging`. ](*SRS\_Diag\_04070*)

**[SWS\_Dem\_00489]** [ 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. ](*SRS\_Diag\_04133*)

Note: [Aging](#) is independent of the UDS status bit 3 (ConfirmedDTC) and therefore independent of the fault confirmation (refer to chapter [7.6.4](#)).

**[SWS\_Dem\_01054]** [ Upon event [aging counter](#) reach threshold [DemAgingCycleCounterThresholdForTFSLC](#), the UDS status bit 5 (TestFailedSinceLastClear) shall be set to 0 if [DemStatusBitHandlingTestFailedSinceLastClear](#) is set to [DEM\\_STATUS\\_BIT\\_AGING\\_AND\\_DISPLACEMENT](#). ]()

In case parameter [DemAgingCycleCounterThresholdForTFSLC](#) is not configured, [TestFailedSinceLastClear](#) will not be aged. ]([SRS\\_Diag\\_04133](#))

**[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. ]([SRS\\_Diag\\_04065](#))

**[SWS\_Dem\_01185]** [ 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. ]([SRS\\_Diag\\_04065](#), [SRS\\_Diag\\_04133](#))

**[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 [7.6.8](#) for the handling of the operation cycle.

**[SWS\_Dem\_00490]** [ If the configuration parameter [DemAgingRequiresTestedCycle](#) (refer to [DemGeneral](#)) 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]** [ If the configuration parameter [DemAgingRequiresTestedCycle](#) (refer [DemGeneral](#)) 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 (TestNotCompleteThisOperationCycle) is set to zero. ]([SRS\\_Diag\\_04076](#))

### 7.6.11 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 7.43](#)).

**[SWS\_Dem\_01056]** [ The Dem shall process Healing only on passed events using the Failure cycle reference [DemEventFailureCycleRef](#). ]()

### 7.6.11.1 Warning indicator handling

The detailed configuration of the warning indicator handling within the Dem is shown in Figure 7.44.

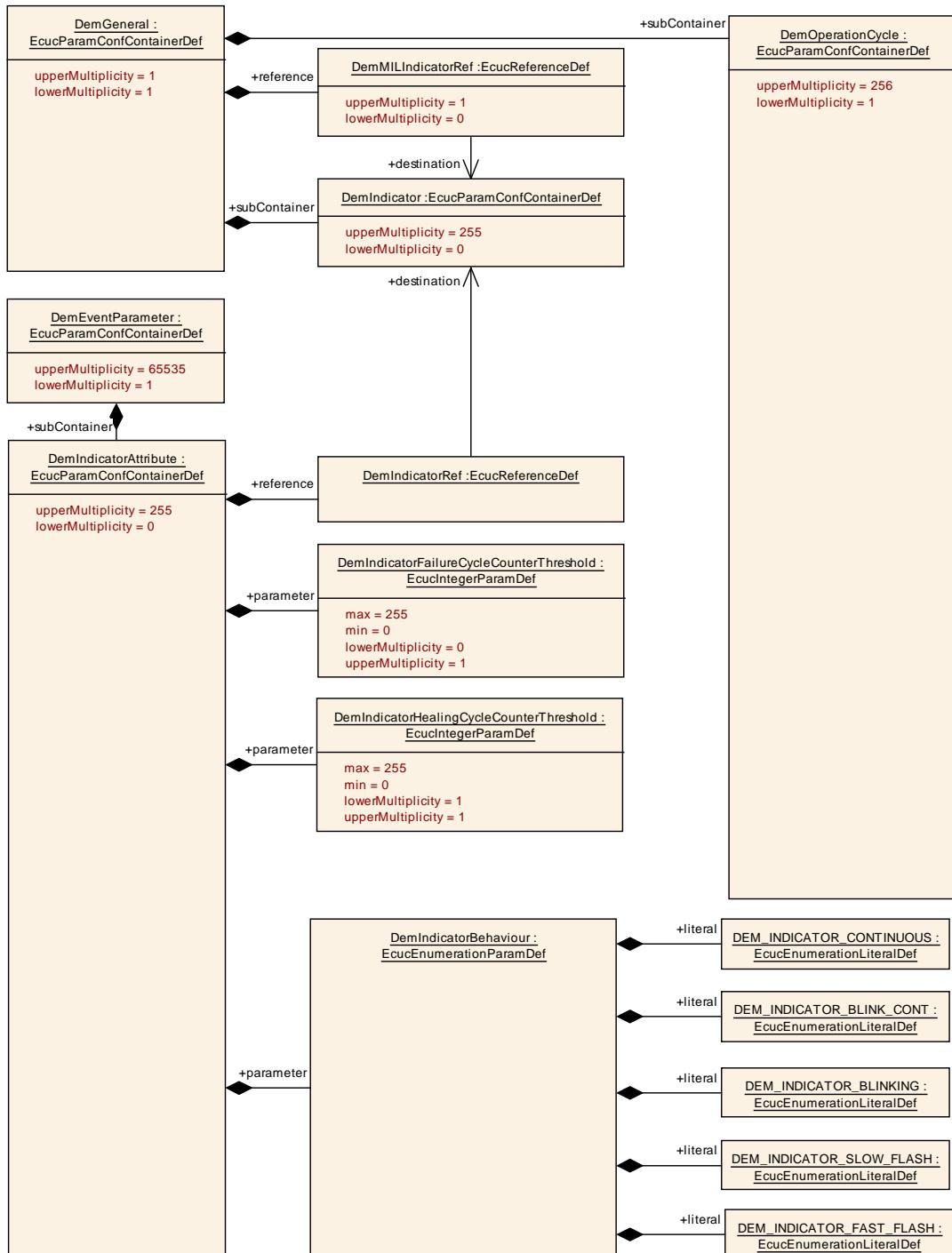


Figure 7.44: Warning indicator configuration

**[SWS\_Dem\_00499]** [ The Dem module shall support event specific counters to activate and deactivate indicators, which are calculated based on the configured failure

and healing cycles (e.g. to turn on the MIL upon fault confirmation, and turn off the MIL after subsequent healing over three OBD-driving cycles). ]()

**[SWS\_Dem\_00509]** [ The Dem module shall provide the configuration parameter [DemIndicatorId](#) to cover the identifier of a specific indicator. ]()

**[SWS\_Dem\_00510]** [ The Dem module shall be able to assign no indicator, one indicator, or several indicators to a specific event (refer to [DemIndicatorAttribute](#)). ]([SRS\\_Diag\\_04069](#))

**[SWS\_Dem\_00511]** [ The Dem module shall provide the configuration parameter [DemIndicatorBehaviour](#) (refer to [DemIndicatorAttribute](#)) to assign a specific behavior (e.g. illuminate continuously or blinking) to each indicator and per event. ]([SRS\\_Diag\\_04069](#))

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]** [ 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 DTC status bit 7 (WarningIndicator). ]([SRS\\_Diag\\_04067](#))

**[SWS\_Dem\_00500]** [ The [Dem](#) module shall provide the configuration parameter [DemIndicatorFailureCycleCounterThreshold](#) per indicator per event (refer to [DemIndicatorAttribute](#)) to define the maximum number of tested and failed cycles, before the respective indicator is activated (i.e. generates a specific condition [WarningIndicatorOnCriteriaFulfilled](#), refer to [Figure 7.22](#)). ]([SRS\\_Diag\\_04069](#))

**[SWS\_Dem\_00501]** [ The [Dem](#) module shall generate the condition [WarningIndicatorOnCriteriaFulfilled](#) specific for the assigned warning indicator, if the respective indicator [failure counter](#) of the event entry has been processed (counted further) [DemIndicatorFailureCycleCounterThreshold](#) times. ]()

Note: For emission related events the fault confirmation given by the DemDTCAttributes also leads to the [MIL](#) on i.e. in that case, the [IndicatorOnCriteria](#) bases on the same failure cycle settings (trigger and threshold) of the event. Then the source would be [DEM\\_FAILURE\\_CYCLE\\_EVENT](#) and the indicator specific settings would be ignored (grayed out). If an additional indicator needs to be activated e.g. via time for this particular event, then the [DemIndicatorSource](#) would be [DEM\\_FAILURE\\_CYCLE\\_INDICATOR](#). That means the trigger and the threshold are defined for this indicator.

**[SWS\_Dem\_00502]** [ The Dem module shall provide the configuration parameter [DemIndicatorHealingCycleCounterThreshold](#) per indicator per event (refer to [DemIndicatorAttribute](#)) to define the maximum number of tested and passed healing cycles, before the respective indicator is deactivated (i.e. generates a specific condition [WarningIndicatorOffCriteriaFulfilled](#), refer to [Figure 7.22](#)). ]()

**[SWS\_Dem\_00503]** [ The Dem module shall generate the condition [WarningIndicatorOffCriteriaFulfilled](#) specific for the assigned warning indicator, if the respective indica-

tor healing counter of the event entry has been processed (counted further) [DemIndicatorHealingCycleCounterThreshold](#) times. ]()

**[SWS\_Dem\_00533]** [ If more than one indicator is configured for a specific event and each assigned indicator healing counter has been processed (counted further) [DemIndicatorHealingCycleCounterThreshold](#) times, the Dem module shall reset the UDS DTC status bit 7 (WarningIndicatorRequested). ]([SRS\\_Diag\\_04069](#))

**[SWS\_Dem\_00506]** [ The Dem module shall process all indicator failure or indicator healing counters, if the cycle is started (refer to chapter [7.6.8](#)) according to the current status of the stored events. ]([SRS\\_Diag\\_04069](#))

Note: This cycle type could be for example equal to [DEM\\_OPCYC\\_OBD\\_DCY](#) or to [DEM\\_OPCYC\\_POWER](#).

It is possible to define a failure cycle or healing cycle, which does not correspond to an operation cycle in the sense, e.g. of a period of time with a beginning and an end. For that reason, the operation cycles can be extended with new types, e.g. [DEM\\_OPCYC\\_TIME\\_200MS](#) interpreted as triggers. In this example, the trigger is set every 200ms, i.e. is started every 200ms, to initiate the indicator counters to be processed.

### 7.6.11.2 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[1] 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 EventStatusChanged).

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. fail-safe state. ](SRS\_Diag\_04128)

**[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). ](SRS\_Diag\_04128)

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 . ](SRS\_Diag\_04128)

**[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. ](SRS\_Diag\_04128)

**[SWS\_Dem\_00835]** [ The WIR-bit of the corresponding event shall be set to "0" if all referenced Dem Indicator(s) are not active and WIRbit is not controlled by the API Dem\_SetWIRStatus. ](SRS\_Diag\_04128)

**[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. ](SRS\_Diag\_04128)

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.

### 7.6.11.3 Handling of the warning indicator lamp (MIL)

**[SWS\_Dem\_00546]** [ For OBD-relevant ECUs, the Dem module shall provide the configuration parameter DemMILIndicatorRef (refer to DemGeneral) 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 [DemEventFailureCycleCounterThreshold](#) and [DemEventFailureCycleRef](#)) has to be consitant 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 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. ]([SRS\\_Diag\\_04069](#))

**[SWS\_Dem\_00535]** [ In case of OBD-relevant events the indicator cycles shall be based on cycles defined by OBD legislation. ]([SRS\\_Diag\\_04069](#))

#### 7.6.11.4 Notification and Set of the warning indicator status

**[SWS\_Dem\_00046]** [ The [Dem](#) module shall provide the API [Dem\\_GetIndicatorStatus](#) (refer to chapter [8.3.3.21](#)) that a software component can get information about the calculated indicator status. ]([SRS\\_Diag\\_04069](#))

### 7.7 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 [12] for further details). For these errors (only a small number compared to application specific events), some additional aspects apply:

- Errors can be detected 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)

Since BSW events are treated as normal SW-C events in the [event memory](#), they can also be classified (availability in workshop tester), aged (refer to chapter [7.6.9](#)) and prioritized (displacement handling).

[Dem\\_ReportErrorStatus](#) 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\\_ReportErrorStatus](#) is necessary to process the reported fault within the main function of the Dem module.

During initialization of the `Dem` module, the API `Dem_ReportErrorStatus` supports debouncing (pre-failed and pre-passed). The corresponding internal `debounce counters/timers` will be initialized by calling `Dem_PrelInit` and can be reset by `Dem_ResetEventDebounceStatus`.

**[SWS\_Dem\_00107]** [ The `Dem` module shall provide the interface `Dem_ReportErrorStatus` (refer to chapter 8.3.3.1) to the BSW modules, to report BSW events which are processed like event reports by `Dem_SetEventStatus`). ](SRS\_BSW\_00417, SRS\_BSW\_00339)

**[SWS\_Dem\_00167]** [ The `Dem` module shall provide a buffer mechanism (FIFO) to queue events which are reported before `Dem_Init` via `Dem_ReportErrorStatus` as qualified (Failed/Passed) or reaching the qualification (the `Dem` module cannot access the `event memory`). ](SRS\_BSW\_00339)

**[SWS\_Dem\_00852]** [ The `Dem` module shall provide a buffer mechanism (FIFO) to queue events which are reported via `Dem_ReportErrorStatus` during normal operation (after the `Dem` module has been already initialized). ](SRS\_Diag\_04070, SRS\_BSW\_00339)

**[SWS\_Dem\_00207]** [ The size of the queue of the function `Dem_ReportErrorStatus` is configurable by the configuration parameter `DemBswErrorBufferSize` (refer to `Dem-General`). ](SRS\_BSW\_00339)

**[SWS\_Dem\_01079]** [ 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 `DemEvent Kind = DEM_EVENT_KIND_BSW` and undefined `DemReportBehavior` or `DemReportBehavior = REPORT_BEFORE_INIT`. In case `DemBswErrorBufferSize` is configured, it overwrites the calculated size of the queue. ](SRS\_BSW\_00339)

Note: During startup phase, not all event-related data might be available. SW-C events cannot be stored before complete initialization of the `Dem`.

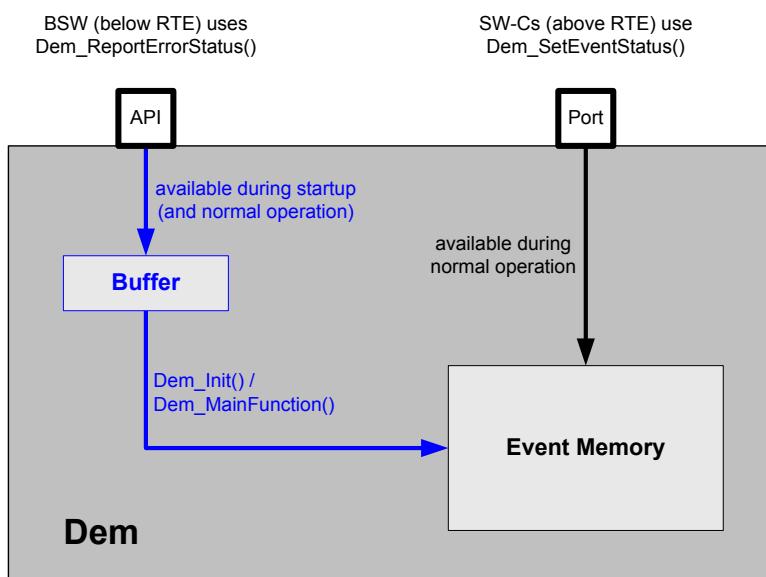


Figure 7.45: `Dem_ReportErrorStatus` buffering behavior

**[SWS\_Dem\_00851]** [ All events which are reported via [Dem\\_ReportErrorStatus](#) before [Dem\\_Init](#) shall not consider the state of the operation cycles (refer to [\[SWS\\_Dem\\_00481\]](#)). ] ([SRS\\_BSW\\_00339](#))

**[SWS\_Dem\_00854]** [ The operation cycle state shall be evaluated, when processing the events from the queue. If the operation cycle related to that event is started, the event shall be processed regularly (similar to [Dem\\_SetEventStatus](#)); otherwise in case the operation cycle is not started the event shall be dropped and the [debounce counter](#) shall be reset to zero. ] ()

Note: All events which are reported via [Dem\\_ReportErrorStatus](#) before [Dem\\_Init](#) should use a operation cycle which is auto started ([DemOperationCycleAutostart](#) set to true) or persistently stored ([DemOperationCycleStatusStorage](#) set to true).

## 7.8 OBD-specific functionality

### 7.8.1 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 \[2\]](#) 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 handled 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](#):

<b>Functionality</b>	<b>Master ECU</b>	<b>Primary ECU</b>	<b>Dependent / Secondary ECU</b>
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
<b>Calulation and provision of General Information (DYC, General Nominator, PFC cycle,..)</b>	Yes	No	No
<b>Reception and execution of General Information (DYC, General Nominator, PFC cycle,..)</b>	(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 7.3: 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 differentiate 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 [13] and [18]).

**[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. ] ([SRS\\_Diag\\_04010](#))

**Calculation and provision of General Information Data:**

The Master ECU shall calculate and provide the following General Information Data via the Busystem 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)

The information about the actual state (started / ended) can be accessed via the API [Dem\\_GetOperationCycleState](#).

**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 “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 [7.8.5.8](#))**Note: Calibration Identification (CAL-ID) and Calibration Verification Number (CVN)**

The Calculation of the Calibration Identification (CAL-ID) and Calibration Verification Number (CVN) is not a BSW Task and will not be handled within the Dem.

## 7.8.2 PIDs provided by Dem

**[SWS\_Dem\_00293]** [ For the following PIDs the data layout shall be compliant to SAE J1979DA [19].

- PID \$01 monitor status since DTCs cleared (4 byte)
- PID \$02<sup>1</sup> DTC that caused required freeze frame storage (2 byte)
- PID \$1C OBD requirements to which vehicle or engine is certified (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)
- 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)

] ([SRS\\_Diag\\_04082](#))

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\_00351]** [ The Dem module shall compute and provide the number of confirmed faults (PID \$01, Byte A). ] ([SRS\\_Diag\\_04082](#), [SRS\\_Diag\\_04141](#))

**[SWS\_Dem\_01137]** [ For [WWH-OBD](#) PID\$01 shall not provide the number of confirmed faults. ] ([SRS\\_Diag\\_04082](#), [SRS\\_Diag\\_04141](#))

**[SWS\_Dem\_01138]** [ The Dem module shall compute and provide the ECU MIL status (PID \$01, Byte A). ] ([SRS\\_Diag\\_04082](#), [SRS\\_Diag\\_04141](#))

**[SWS\_Dem\_01139]** [ For [WWH-OBD](#) PID\$01 shall not provide the ECU MIL status. ] ([SRS\\_Diag\\_04082](#), [SRS\\_Diag\\_04141](#))

<sup>1</sup>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 [\[SWS\\_Dem\\_00623\]](#))

**[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 [13], e.g. OBD, OBDII, JOBD etc. The value PID1Cvalue to return is in configuration parameter DemOBDCCompliance defined. ]([SRS\\_Diag\\_04082](#), [SRS\\_Diag\\_04141](#))

### 7.8.2.1 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. ]([SRS\\_Diag\\_04082](#))

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. ]([SRS\\_Diag\\_04082](#))

`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 (refer to chapter [7.10.8](#)) to calculate PID \$21 and PID \$31. ]([SRS\\_Diag\\_04082](#))

**[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. ]([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_00347]** [ If PID \$1E (auxiliary input status) is supported the PTO (Power Take Off) related event status handling shall implemented inside the Dem module (refer to [20]). ]([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_00377]** [ The Dem module shall provide the interface `Dem_SetPtoStatus` allowing a SWC implementing the PTO functionality to notify the Dem module if PTO is active or inactive (refer to section 8). ]([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_00378]** [ The Dem module shall support the configuration parameter [DemConsiderPtoStatus](#) indicating that a certain event is affected by the Dem PTO handling. ]([SRS\\_Diag\\_04082](#))

The Dem module provides the configuration switch [DemPTOSuppor](#) to enable or disable the usage of PID \$1E.

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. ]([SRS\\_Diag\\_04082](#))

According to SAEJ1979 [18], the group AirCondition Component shall not be supported anymore. However, it is still included in ISO 15031-5 [13]. 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. ]([SRS\\_Diag\\_04082](#))

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]). ]([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_01097]** [ On invocation of [Dem\\_GetDataOfPID21](#) the internal calculated [PID](#) \$21 value shall be returned. ]([SRS\\_Diag\\_04082](#))

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. ]([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_01099]** [ If [Dem\\_DcmReadDataOfPID21](#) is called before [Dem\\_SetDataOfPID21](#), the Dem shall return 0xFFFF as [PID \\$21](#) value. ]([SRS\\_Diag\\_04082](#))

### 7.8.3 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 (refer to chapter [7.4.8](#)) shall be ignored for this computation. ]([SRS\\_Diag\\_04082](#))

Note: [DemEventOBDReadinessGroup](#) has a regulation background and special computations based on ModelYear (MY) and Market.

**[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 (refer to chapter [7.4.8](#)) shall be ignored for this computation. ]([SRS\\_Diag\\_04082](#))

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 (refer to chapter [7.4.8](#)) shall be ignored for this computation. ]([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_00348]** [ The Dem module shall provide the disabling of events (refer to chapter [8.3.8.1](#)). OBD Events Suppression (refer to chapter [7.4.8](#)) shall be ignored for this computation. ]([SRS\\_Diag\\_04082](#))

**[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](#). ]([SRS\\_Diag\\_04082](#))

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](#)

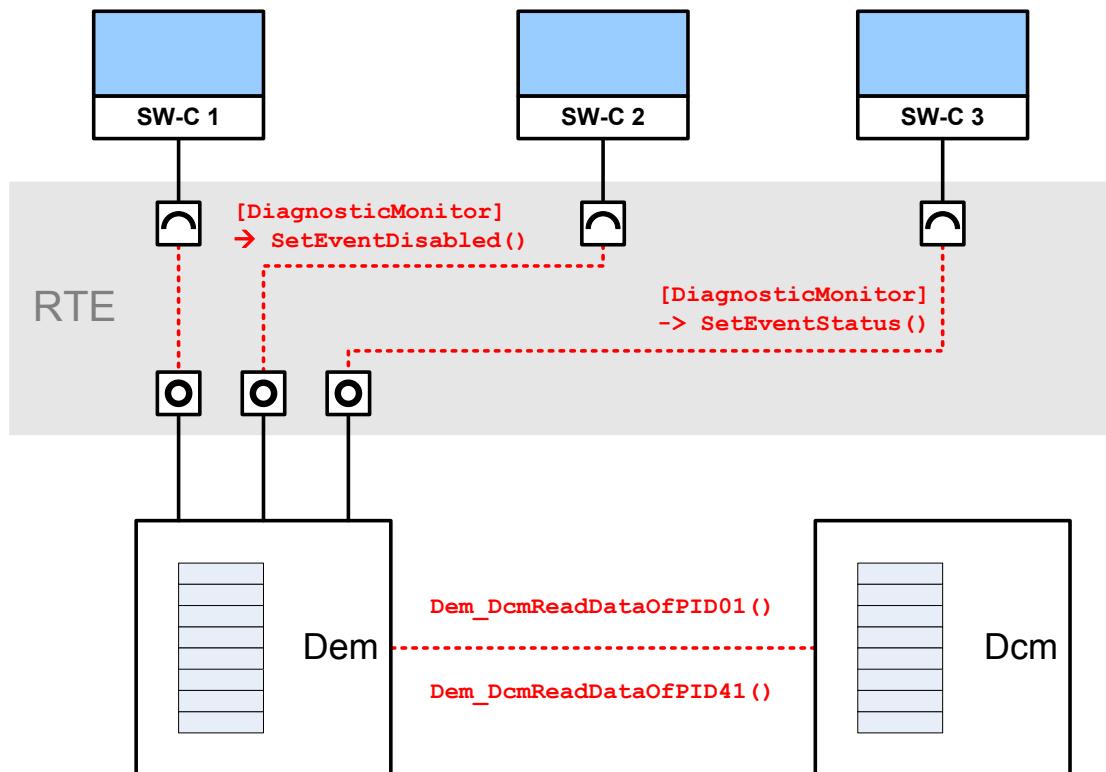


Figure 7.46: Dem calculates PID\$01 and PID\$41 data based on specific port operations

#### 7.8.4 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.  
]([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_00710]** [ The numerator shall be calculated in the ECU (Master and Primary). ]([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_00711]** [ The [Dem](#) of the Master ECU shall calculate the [IUMPR](#)-Cycle Flag.  
]([SRS\\_Diag\\_04082](#))

**[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. ]([SRS\\_Diag\\_04082](#))

The Master ECU (software component) will provide the IUMPR-Cycle Flag (included in the General Denominator signal) via the bus system.

**[SWS\_Dem\_00712]** [ The ECU internal denominator incrementation (in a Master / Primary ECU) shall be done depended on the IUMPR-Cycle Flag (included in the General Denominator signal). ](SRS\_Diag\_04082)

**[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: APIs to lock and unlock the denominator under special conditions are defined in [\[SWS\\_Dem\\_00362\]](#).

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 [13] and [18]).

**[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. ](SRS\_Diag\_04082)

**[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. ](SRS\_Diag\_04082)

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. ](SRS\_Diag\_04082)

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 Performance 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 Ratiold (refer to DemRatio). Thus, the container DemRatio contains the EventId, primary FID, secondary FIDs, IUMPR-group and an interface option to configure “APluse” 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). ] ([SRS\\_Diag\\_04001](#))

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\_00360]** [ For OBD relevant systems the Dem module shall provide the API [Dem\\_ReplIUMPRFaultDetect](#). ] ([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_00296]** [ The Dem module shall provide the API [Dem\\_ReplIUMPRFaultDetect](#) for the asymmetric monitor to report that a malfunction could have been found. ] ([SRS\\_Diag\\_04082](#))

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. ] (*SRS\_Diag\_04082*)

**[SWS\_Dem\_00361]** [ The Dem module shall provide the configuration parameter **DemRatioKind**, to indicate per Ratiold if the numerator is calculated based on the TESTED-status or the API call. ] (*SRS\_Diag\_04082*)

#### **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 to lock the denominator per API at the beginning of the driving cycle and will unlock it when the additional conditions on the denominator are met.

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 Ratiold. ] (*SRS\_Diag\_04082*)

**[SWS\_Dem\_00362]** [ The Dem module shall provide the APIs for locking (refer to chapter [8.3.8.5](#)) and unlocking (refer to chapter [8.3.8.6](#)) of the denominator under special conditions if the **DemIUMPRDenGroup** is configured as **DEM\_IUMPR\_DEN\_PHYS\_API**. ] (*SRS\_Diag\_04082*)

**[SWS\_Dem\_00297]** [ The Dem shall provide the API **Dem\_ReplIUMPRDenLock** to IUMPR-relevant SWC, to control the denominator specific to the respective Ratiold.] (*SRS\_Diag\_04082*)

Note: This service shall be used by the monitor to report, that a denominator of a specific monitor (represented by FID and EventId) is locked for physical reasons.

**[SWS\_Dem\_00308]** [ The Dem shall provide the API **Dem\_ReplIUMPRDenRelease** to IUMPRrelevant SW-C, to control the denominator specific to the respective Ratiold.] (*SRS\_Diag\_04082*)

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](#) (refer to chapter [8.3.8.3](#)) to set the status of a particular condition (mainly in depending ECUs). ]([SRS\\_Diag\\_04082](#))

Note: To communicate 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\_00299]** [ As long as an event has Pending status, the Dem module shall stop increasing the numerator and denominator. ]([SRS\\_Diag\\_04001](#))

Based on the related FID (FiM access) and Ratiod, the Dem module can determine which numerator and denominator has to be stopped.

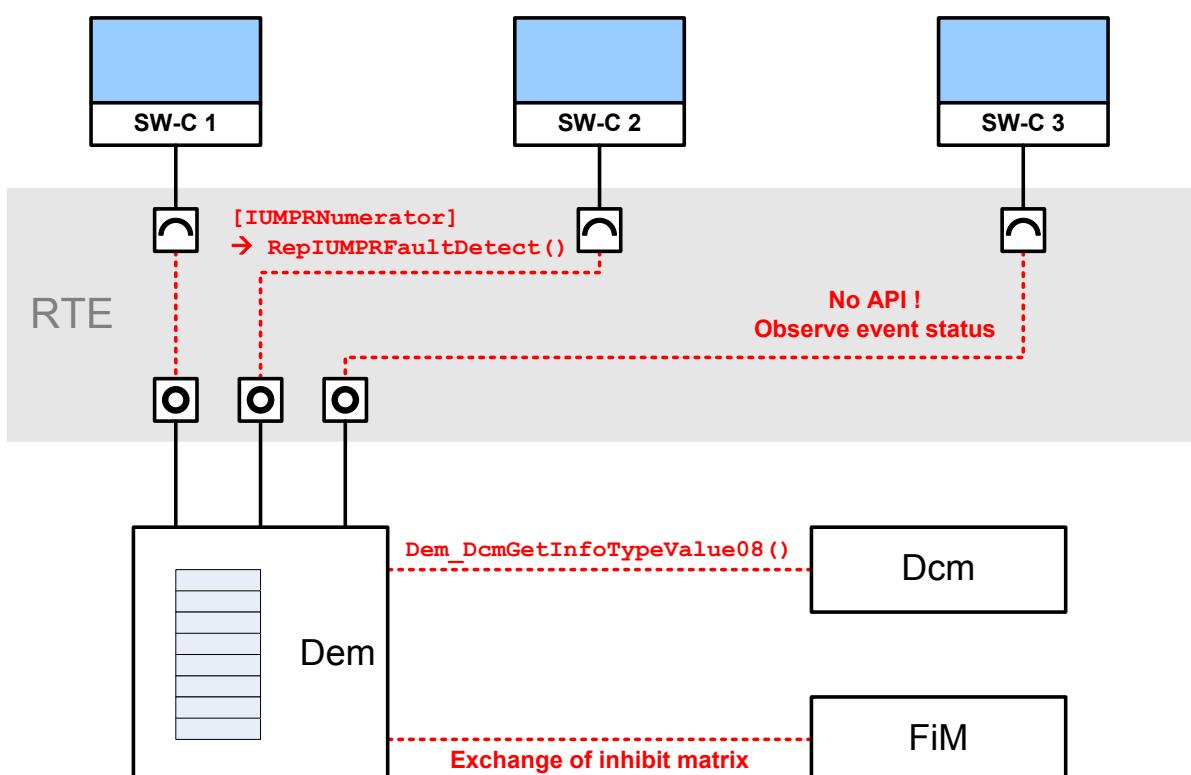


Figure 7.47: Dem calculates IUMPR data based on specific interface operations

## 7.8.5 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 [18]. 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.)

### 7.8.5.1 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 TBD is provided.

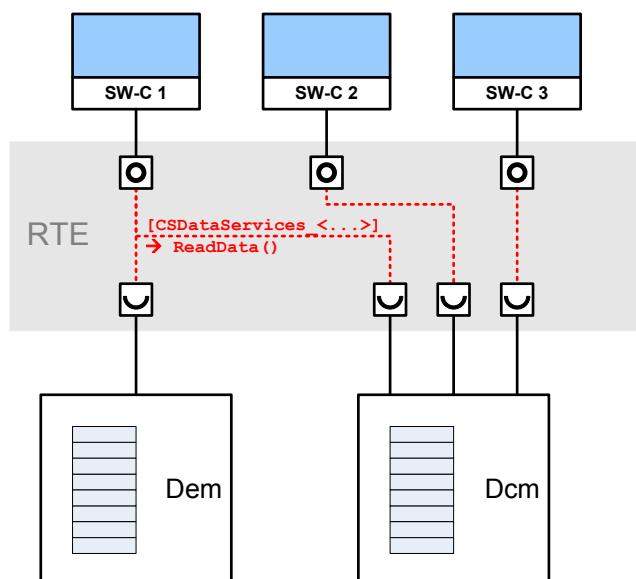
### 7.8.5.2 Service \$02 Read Powertrain Freeze Frame Data

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](#)). ] ([SRS\\_Diag\\_04082](#))

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 7.48: 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](#)). ] ([SRS\\_Diag\\_04082](#))

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 zero-based 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. ] ([SRS\\_Diag\\_04082](#))

Note: This index is not configured explicitly (to be able to avoid resource overhead for e.g. mapping tables in the implementation).

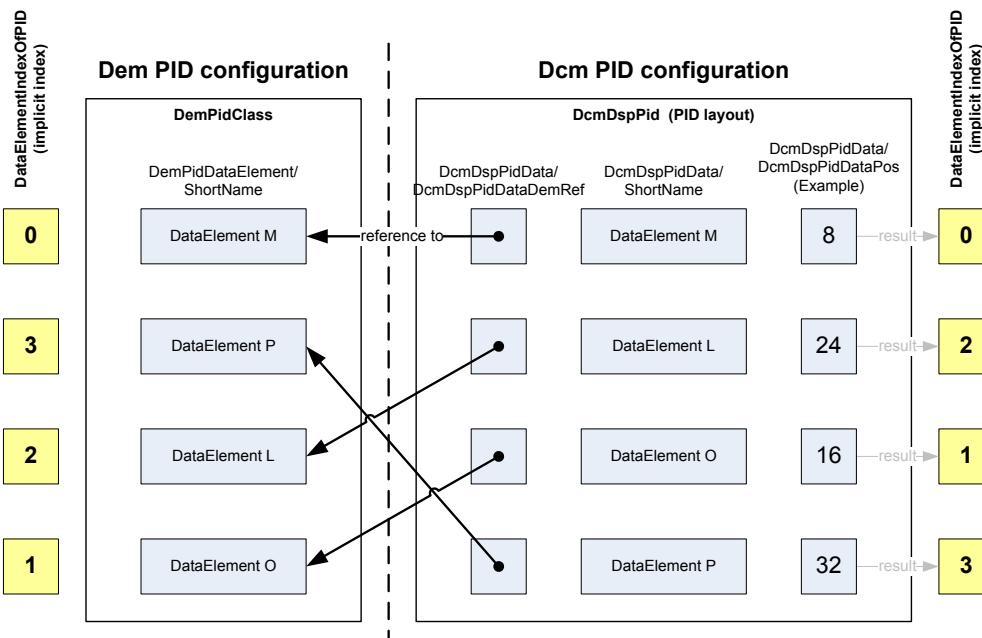


Figure 7.49: PID layout configuration within Dcm (master) and Dem

**[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). ] ([SRS\\_Diag\\_04082](#))

### 7.8.5.3 Service \$03 - Read Emission-Related Powertrain Diagnostic Trouble Codes

The diagnostic service provides the OBD DTCs where the DTC Status bit 3 (ConfirmedDTC) is currently set. Therefore the interfaces described in chapter [8.3.4.1](#) will be used by [Dcm](#).

### 7.8.5.4 Service \$04 - Clear Reset Emission-Related Diagnostic Information

**[SWS\_Dem\_00718]** [ In Master, Primary and [Dependent / Secondary ECUs](#) executing service \$04 shall clear all emission-related diagnostic information in primary and user defined (event) memory (if configured) as defined in SAE J1979 [18]. ]([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_00719]** [ Executing service \$04 shall include non emissionrelated diagnostic information (e.g. Debouncing counter, Event/DTC related status information as TestFailedSinceLastClear) of primary and all user defined [event memories](#) (if configured) in Master, Primary and [Dependent / Secondary ECUs](#). ]([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_00720]** [ The [Master ECU](#) shall inform the [Dependent / Secondary ECUs](#) about the execution of service \$04 via the bussystem. ]()

**[SWS\_Dem\_00721]** [ The Dem (of the [Master ECU](#)) shall execute the software component API [<Module>\\_SetClearDTC](#) to inform a software component about service \$04 execution. The software component shall send this information via the bussystem. ]([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_00722]** [ The Dem (of a [Dependent / Secondary ECU](#)) shall provide an API [Dem\\_SetClearDTC](#) to get informed about the reception of service \$04 execution by a software component. ]([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_00723]** [ In [Dependent / Secondary ECU](#) service \$04 execution shall lead to reset / erase to all (non emissionrelated and emission-related) DTCs in primary and all user defined [event memories](#) (if configured). ]([SRS\\_Diag\\_04082](#))

### 7.8.5.5 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 [13] and SAE J1979 [18]) 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 Event 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 CompuRational-Coeffs 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]** [ The [DTRId](#) is assigned by the [Dem](#) during the [BSW](#) configuration step. ]([SRS\\_Diag\\_04129](#))

Note: The Dem refers to a BSW configuration [DemDtr](#) which associates

- a [DTRId](#) 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](#). ]([SRS\\_Diag\\_04001](#))

**[SWS\_Dem\_00754]** [ The Dem shall use the [DemDtrUpdateKind](#) for the evaluation and processing of reported [DTR](#)-values, i.e. [DTRs](#) are only forwarded to the [Dcm](#), if the related event (if it is configured) has a certain state.

- [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

] ([SRS\\_Diag\\_04082](#))

**[SWS\_Dem\_00755]** [ The Dem shall only support a linear mapping, i.e. only support values for the coefficients b, c, and f. ]([SRS\\_Diag\\_04082](#))

Note: ASAM formula coefficients:  $(ax^2 + bx + c) / (dx^2 + ex + f)$  supporting on b, c and f.

**[SWS\_Dem\_00756]** [ 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 [DTRId](#), the test value, the minimum and maximum limit, and a state indicating whether the monitor provides a valid minium limit, maximum limit, or both, or whether the reported values shall be reset to zero. ]([SRS\\_Diag\\_04030](#))

**[SWS\_Dem\_00757]** [ Based on [DemDtrUpdateKind](#) and the status of the associated event (taking enable/storage conditions into account), the Dem shall either process or ignore the reported [DTR](#) values. ]([SRS\\_Diag\\_04082](#))

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 event is 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]** [ 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. ]([SRS\\_Diag\\_04030](#))

**[SWS\_Dem\_00760]** [ Upon request by the [Dcm](#), the [Dem](#) shall compute "Supported-OBDMID" information, i.e. the "availability OBDMIDs". Since the [Dem\\_DcmGetAvailableOBDMIDs](#) reports only value per standardized OBDMID (\$00, \$20, \$40...), the [Dcm](#) has to call this API iteratively until no further availability OBDMID is supported. ]()

**[SWS\_Dem\_00761]** [ Upon request by the Dcm, the Dem shall respond 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. ]([SRS\\_Diag\\_04082](#))

**[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. ]([SRS\\_Diag\\_04001](#))

**[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\\_DcmClearDTC](#) with:

**DTC** = [DEM\\_DTC\\_GROUP\\_ALL\\_DTCS](#)

**DTCFormat** = [DEM\\_DTC\\_FORMAT\\_OBD](#)

**DTCOrigin** = DEM\_DTC\_ORIGIN\_PRIMARY\_MEMORY

](*SRS\_Diag\_04082*)

#### 7.8.5.6 Service \$07 - Read emission-related diagnostic trouble codes detected 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 [8.3.4.1](#) will be used by **Dcm**.

#### 7.8.5.7 Service \$0A - Read Emission-Related Diagnostic Trouble Codes with Permanent Status

Permanent DTCs are stored in the permanent fault memory.

An OBD related DTC entering the permanent fault memory with illuminating the MIL and leaving it with deactivating the MIL.

The permanent fault memory is robust against ClearDTC and Power Fail. After a ClearDTC an event can only be removed from the permanent fault memory under special conditions. This special conditions are defined by the OBD regulation.

**[SWS\_Dem\_00300]** [ The Dem shall be able to handle Permanent DTCs according to regulations (refer to [21]) within the specific **event memory** type "permanent" (refer to chapter [7.3.4](#)). ](*SRS\_Diag\_04082*)

**[SWS\_Dem\_00590]** [ Events that have been confirmed and activate the MIL (refer to OBDrelevant DTCs, chapter [7.4.1](#)), shall be stored robustly against ClearDTC or powerfail (for details confer [21]). ](*SRS\_Diag\_04082*)

**[SWS\_Dem\_00301]** [ The **Dem** shall provide the ability to access the non-volatile stored Permanent DTC by filtering for permanent DTCs to the **Dcm** (refer to chapter [8.3.4.1.6 Dem\\_DcmSetDTCFilter](#) and chapter [8.3.4.1.8 Dem\\_DcmGetNextFilteredDTC](#)). ](*SRS\_Diag\_04082*)

Note: For service \$0A, the **Dcm** uses the **DTCOrigin** DEM\_DTC\_ORIGIN\_PERMANENT\_MEMORY.

**[SWS\_Dem\_01076]** [ New permanent faults of the current driving cycle shall be reported to the **Dcm** service \$0A request latest after the next **OBD** driving cycle starts. ](*SRS\_Diag\_04001*)

**[SWS\_Dem\_01077]** [ New permanent faults shall be stored in non-volatile memory (Next Driving Cycle after WriteAll or after positive Callback from NvM) latest at **ECU** shutdown. ](*SRS\_Diag\_04001*)

**[SWS\_Dem\_00725]** [ The [Dem](#) of the [MIL Master ECU](#) shall provide the interface [Dem\\_GetPfcCycleQualified](#) (returning a boolean) (refer to chapter [8.3.8.15](#)) to read the current state of the [PFC cycle](#). TRUE indicates that during the current [OBD](#) driving cycle the conditions for the [PFC cycle](#) have been met. ]([SRS\\_Diag\\_04082](#))

Note: A SWC on the MIL Master Ecu forwards the information "Minimum trip conditions satisfied" (PFC cycle) via Bussystem to the primary ECUs.

Note: A software component in the Master ECU optionally polls the API [Dem\\_GetPfcCycleQualified](#) and send the information via the bussystem.

**[SWS\_Dem\_00728]** [ The Dem (of a Primary ECU) shall provide the interface [Dem\\_GetPfcCycleQualified](#) to get informed that the current OBD driving cycle has met the criteria for the PFC cycle by a software component. ]([SRS\\_Diag\\_04082](#))

### 7.8.5.8 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.

[Primary ECUs](#) must report their Indicator ([MIL](#)) request (when the criterion for triggering the [MIL](#) is fulfilled) to the “MIL Master functionality“.

[Primary ECUs](#) could use the function [Dem\\_GetIndicatorStatus](#) to poll the [Dem](#) to get information about an event with a related Indicator ([MIL](#)) to inform the “MIL Master functionality“.

## 7.8.6 WWH-OBD

### 7.8.6.1 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 [3], Annex D.3 “DTCSeverityMask and DTC class information bit definitions” (refer to chapter [8.3.4.1.4](#) and chapter [8.3.4.1.10](#)), only if configured for at least one DTC. ]([SRS\\_Diag\\_04141](#))

**[constr\_6109]** [ The DTC class is only available for ISO 14229-1 [3]DTCs. It is configurable per DTC optionally (refer to [DemWWHOBDDTCClass](#)). ]()

**[constr\_6110]** [ The [WWH-OBD](#) DTC priority shall be according table [Table 7.4](#). ]()

<b>WWH-OBD DTC class</b>	<b>Priority</b>
A	> B1 DTC
B1	> B2 DTC
B2	> C DTC
C	> no OBD DTC

---

**Table 7.4: WWH-OBD DTC priority**

Note : ISO 27145-3 [22] defines the following DTC Classes utilizing the severity levels: No Class, Class A, Class B1, Class B2 and Class C.

### 7.8.6.2 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. ]([SRS\\_Diag\\_04141](#))

**[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. ]([SRS\\_Diag\\_04141](#))

**[SWS\_Dem\_01143]** [ The counter shall halt its value when the Continuous-MI is no longer commanded to be on. ]([SRS\\_Diag\\_04141](#))

**[SWS\_Dem\_01144]** [ The counter shall continue its incrementation if the Continuous-MI is again commanded on within three operation sequences. ]([SRS\\_Diag\\_04141](#))

**[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. ]([SRS\\_Diag\\_04141](#))

**[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. ]([SRS\\_Diag\\_04141](#))

**[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. ]([SRS\\_Diag\\_04141](#))

**[SWS\_Dem\_01148]** [ The counter shall be stored non-volatile. ]([SRS\\_Diag\\_04141](#))

### 7.8.6.3 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, ]([SRS\\_Diag\\_04141](#))

**[SWS\_Dem\_01150]** [ The counter shall halt its value when the Continuous-MI is no longer commanded to be on, ](SRS\_Diag\_04141)

**[SWS\_Dem\_01151]** [ The counter shall never be reset. ](SRS\_Diag\_04141)

**[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. ](SRS\_Diag\_04141)

**[SWS\_Dem\_01153]** [ The counter shall be stored non-volatile. ](SRS\_Diag\_04141)

#### 7.8.6.4 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. ](SRS\_Diag\_04141)

**[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. ](SRS\_Diag\_04141)

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. ](SRS\_Diag\_04141)

**[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. ](SRS\_Diag\_04141)

**[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. ](SRS\_Diag\_04141)

**[SWS\_Dem\_01159]** [ The B1 counter shall be stored non-volatile. ](SRS\_Diag\_04141)

#### 7.8.6.5 Activation Mode

##### Activation Mode Escalation

**[SWS\_Dem\_01160]** [ If the B1 counter exceeds 200 engine operating hours the [Activation Mode 4](#) shall be active. ](SRS\_Diag\_04141)

**[SWS\_Dem\_01161]** [ If at least one Class A event is Confirmed and TestFailed the Activation Mode 4 shall be active. ]([SRS\\_Diag\\_04141](#))

### 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. ]([SRS\\_Diag\\_04141](#))

**[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. ]([SRS\\_Diag\\_04141](#))

**[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. ]([SRS\\_Diag\\_04141](#))

**[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. ]([SRS\\_Diag\\_04141](#))

**[SWS\_Dem\_01166]** [ The function Dem\_GetIndicatorStatus shall return the current "indicator status"(refer to [Table 7.5](#)) of those events referenced by the DemMILIndicatorRef. The conditions for the activation modes are defined in [Table 7.5](#). ]([SRS\\_Diag\\_04141](#))

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 7.5: Mapping of conditions to the activation mode**

### 7.8.6.6 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 DTC status bit 2 (PendingDTC) from 0 to 1 (via DemFreezeFrameRecordTrigger set to DEM\_TRIGGER\_ON\_PENDING). ]([SRS\\_Diag\\_04141](#))

**[SWS\_Dem\_01171]** [ The update behavior if the Freeze Frame 0x00 is updated on transition of DTC status bit 3 (ConfirmedDTC) from 0 to 1, can be configured using DemFreezeFrameRecordUpdate. ]([SRS\\_Diag\\_04141](#))

**[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. ]([SRS\\_Diag\\_04141](#))

**[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. ]([SRS\\_Diag\\_04141](#))

**[SWS\_Dem\_01174]** [ The Freeze Frame 0x00 shall be erased if the event was removed from fault memory due to aging. ]([SRS\\_Diag\\_04141](#))

**[SWS\_Dem\_01175]** [ The legislative Freeze Frame 0x00 is configured analogues to the non-emission related freeze frames described in chapter 7.3.7.1. ]([SRS\\_Diag\\_04141](#))

#### 7.8.6.7 Extended Data Record 0x90

Extended Data Record 0x90 is not supported due to a single B1 counter only supported in Dem.

#### 7.8.6.8 Aging

**[SWS\_Dem\_01176]** [ Additional to event aging described in chapter 7.6.10 the event shall be aged after 200 hours of engine operation while the aging conditions are met. ]([SRS\\_Diag\\_04141](#))

**[constr\_6111]** [ An [OBD](#) related DTC shall have an aging counter threshold of 40. ]()

**[constr\_6112]** [ An [OBD](#) related DTC shall have the Warm-Up cycle as aging cycle. ]()

#### 7.8.6.9 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\\_DcmGetDTCSeverityAvailabilityMask](#) shall provide the DtcSeverityAvailabilityMask to the DCM. ]([SRS\\_Diag\\_04141](#))

Note: The DTCSeverityAvailabilityMask can be derived from the configured DTC severity and the [WWH-OBD](#) DTC class.

### 7.8.6.10 Service \$14 FFFF33 - Clear Emission Related DTCs

**[SWS\_Dem\_01178]** [ In addition to chapter [7.6.2.2](#) 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

] ([SRS\\_Diag\\_04141](#))

**[SWS\_Dem\_01179]** [ Service \$14 FFFF33 shall clear all DTCs ([WWH-OBD](#) DTCs and UDS DTCs) ] ([SRS\\_Diag\\_04141](#))

## 7.9 J1939 specific functionality

The Dem module is capable to support J1939 networks. On these, it uses a J1939 specific DTC.

**[SWS\_Dem\_00845]** [ The J1939 support shall be available in case the configuration container [DemGeneralJ1939](#) (see [DemGeneral](#)) is present. ] ([SRS\\_Diag\\_04112](#))

### 7.9.1 Read DTC

**[SWS\_Dem\_00855]** [ The function [Dem\\_J1939DcmSetDTCFilter](#) (refer to chapter [8.3.6.1.1](#)) shall set the filter mask attributes to be used for the subsequent calls of [Dem\\_J1939DcmGetNumberOfFilteredDTC](#) and [Dem\\_J1939DcmGetNextFilteredDTC](#), and reset an internal counter to the first event. ] ()

**[SWS\_Dem\_00856]** [ The filter mask attributes set via [Dem\\_J1939DcmSetDTCFilter](#) shall be used until the next call of [Dem\\_J1939DcmSetDTCFilter](#) or Dem initialization. ] ([SRS\\_Diag\\_04112](#))

**[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 7.6](#). ] ()

<i>SetFilter Parameter</i>	<i>DTC Status filter</i>	<i>further filter criteria</i>	<i>DM representation</i>	<i>AllDTCs</i>	<i>EmissionDTCs</i>
----------------------------	--------------------------	--------------------------------	--------------------------	----------------	---------------------

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 7.6: 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

### 7.9.1.1 Composite Malfunction Indicator Lamp Status

**[SWS\_Dem\_00858]** [ 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”. ]([SRS\\_Diag\\_04110](#))

**[SWS\_Dem\_00859]** [ 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](#). ]([SRS\\_Diag\\_04110](#))

**[SWS\_Dem\_00860]** [ 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](#). ]([SRS\\_Diag\\_04110](#))

**[SWS\_Dem\_00861]** [ 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](#). ]([SRS\\_Diag\\_04110](#))

### 7.9.1.2 Composite Red Stop Lamp Status

**[SWS\_Dem\_00862]** [ 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”. ]([SRS\\_Diag\\_04110](#))

**[SWS\_Dem\_00863]** [ 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](#). ]([SRS\\_Diag\\_04110](#))

**[SWS\_Dem\_00864]** [ 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](#). ]([SRS\\_Diag\\_04112](#))

**[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](#). ]([SRS\\_Diag\\_04069](#))

### 7.9.1.3 Composite Amber Warning Lamp Status

**[SWS\_Dem\_00866]** [ 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”. ]([SRS\\_Diag\\_04110](#))

**[SWS\_Dem\_00867]** [ 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]** [ 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](#). ]([SRS\\_Diag\\_04110](#))

**[SWS\_Dem\_00869]** [ 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](#). ]()

### 7.9.1.4 Composite Protect Lamp Status

**[SWS\_Dem\_00870]** [ 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 **DemProtectLamplIndicatorRef** has an IndicatorStatus **DEM\_INDICATOR\_OFF** or **DEM\_INDICATOR\_CONTINUOUS**. ]()

**[SWS\_Dem\_00872]** [ The composite “Flash Protect Lamp” shall be set to “Slow Flash” (0x00) in case the Indicator referenced by **DemProtectLamplIndicatorRef** has an IndicatorStatus **DEM\_INDICATOR\_SLOW\_FLASH**. ](SRS\_Diag\_04110)

**[SWS\_Dem\_00873]** [ The composite “Flash Protect Lamp” shall be set to “Fast Flash” (0x01) in case the Indicator referenced by **DemProtectLamplIndicatorRef** has an IndicatorStatus **DEM\_INDICATOR\_FAST\_FLASH**. ](SRS\_Diag\_04110)

### 7.9.1.5 DTC data acquisition

**[SWS\_Dem\_00874]** [ The function **Dem\_J1939DcmGetNumberOfFilteredDTC** shall return the number of J1939 DTCs matching the defined filter criteria of Table 4 for a specific node defined by the function call of **Dem\_J1939DcmSetDTCFilter**.

- In case the function has calculated the total number of matching DTCs, the return value shall be **DEM\_NUMBER\_OK**.
- In case the search needs to be interrupted due to internal implementations or limitations, the return value can be set to **DEM\_NUMBER\_PENDING**. The out parameter needs not to be valid in this case. The next call of **Dem\_J1939DcmGetNumberOfFilteredDTC** should continue after the interrupted element.

] (SRS\_Diag\_04112)

**[SWS\_Dem\_00875]** [ Each call of the function **Dem\_J1939DcmGetNextFilteredDTC** shall search for the next event having a J1939DTC assigned and matching the filter criteria of Table 4 for the node (see **DemJ1939DTC\_J1939NodeRef**) 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\_FILTERED\_NO\_MATCHING\_ELEMENT**. The out parameters need not to be valid in this case. ]()

**[SWS\_Dem\_00877]** [ In case the function **Dem\_J1939DcmGetNextFilteredDTC** has found an event matching the filter criteria it shall return the corresponding J1939DTC (refer to **DemDTCAttributes** for details) and the corresponding occurrence counter. The return value shall be **DEM\_FILTERED\_OK**. In case the search needs to be interrupted due to internal implementations or limitations, the return value can be set to **DEM\_FILTERED\_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 (0x7F), the returned values shall be set to +126 (0x7F). ](SRS\_Diag\_04111)

### 7.9.2 Clear DTCs

For definition of locking the clearing process, refer to chapter 7.6.2.2.

**[SWS\_Dem\_00878]** [ The Dem module shall provide the Dem\_J1939DcmClearDTC (refer to chapter 8.3.6.2.1) 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 [15] are requested, all active (DM11) or previously active (DM03) DTCs have to be cleared, along with additional information like freeze frames.

**[SWS\_Dem\_00879]** [ 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). ](SRS\_Diag\_04117)

Note: Exceptions may apply due to [SWS\_Dem\_00514].

### 7.9.3 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\_J1939DcmGetNextDTCwithLampStatus.

**[SWS\_Dem\_00880]** [ Each call to Dem\_J1939DcmFirstDTCwithLampStatus shall set the internal counter to the first event having a J1939DTC for this particular “node” assiged. ](SRS\_Diag\_04113)

**[SWS\_Dem\_00881]** [ Each call to Dem\_J1939DcmGetNextDTCwithLampStatus shall search for the next event having a J1939DTC assigned for “node” set by the function Dem\_J1939DcmFirstDTCwithLampStatus. In case no more events are available that have a J1939DTC assigend, the function return value shall be DEM\_FILTERED\_NO\_MATCHING\_ELEMENT. The out parameter needs not to be valid in this case. ](SRS\_Diag\_04113)

**[SWS\_Dem\_00882]** [ 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 DEM\_FILTERED\_OK. In case the search needs to be interrupted due to internal implementations or limitations, the return value can be set to DEM\_FILTERED\_PENDING. The out parameter needs not to be valid in this case. The next call of Dem\_J1939DcmGetNextDTCwithLampStatus should continue after the in-

terrupted element. In case the occurrence counter is above +126 (0x7F), the returned values shall be set to +126 (0x7F). ]([SRS\\_Diag\\_04113](#))

### 7.9.3.1 Malfunction Indicator Lamp

**[SWS\_Dem\_00883]** [ The DTC-specific “Malfunction Indicator Lamp” returned in the function [Dem\\_J1939DcmGetNextDTCwithLampStatus](#) shall be set to “Lamp On” in case the corresponding event has assigned the Indicator referenced by DemMILIndicatorRef and the UDS DTC status bit 7 (WarningIndicator) is active (set to 1), otherwise it shall be set to ‘Lamp Off’. ]([SRS\\_Diag\\_04110](#))

**[SWS\_Dem\_00884]** [ The DTC-specific “Flash Malfunction Indicator Lamp” shall be set to “Unavailable / Do Not Flash” (0x03) in case the corresponding event has set “Lamp On” and the [DemIndicatorBehaviour](#) of the Indicator referenced by DemMILIndicatorRef is set to [DEM\\_INDICATOR\\_CONTINUOUS](#). ]([SRS\\_Diag\\_04110](#))

**[SWS\_Dem\_00885]** [ 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](#). ]([SRS\\_Diag\\_04110](#))

**[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](#). ]([SRS\\_Diag\\_04069](#))

### 7.9.3.2 Red Stop Lamp

**[SWS\_Dem\_00887]** [ The DTC-specific “Red Stop Lamp” returned in the function [Dem\\_J1939DcmGetNextDTCwithLampStatus](#) shall be set to “Lamp On” in case the corresponding event has assigned the Indicator referenced by DemRedStopIndicatorRef and the UDS DTC status bit 7 (WarningIndicator) is active (set to 1), otherwise it shall be set to “Lamp Off”. ]([SRS\\_Diag\\_04110](#))

**[SWS\_Dem\_00888]** [ The DTC-specific “Flash Red Stop Lamp” shall be set to “Unavailable / Do Not Flash” (0x03) in case the corresponding event has set “Lamp On” and the [DemIndicatorBehaviour](#) of the Indicator referenced by DemRedStopIndicatorRef is set to [DEM\\_INDICATOR\\_CONTINUOUS](#). ]()

**[SWS\_Dem\_00889]** [ 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 DemRedStopIndicatorRef is set to [DEM\\_INDICATOR\\_SLOW\\_FLASH](#). ]([SRS\\_Diag\\_04110](#))

**[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 [De-](#)

`mIndicatorBehaviour` of the Indicator referenced by `DemRedStopIndicatorRef` is set to `DEM_INDICATOR_FAST_FLASH`. ]()

### 7.9.3.3 Amber Warning Lamp

**[SWS\_Dem\_00891]** [ The DTC-specific “Amber Warning Lamp” returned in the function `Dem_J1939DcmGetNextDTCwithLampStatus` shall be set to “Lamp On” in case the corresponding event has assigned the Indicator referenced by `DemAmberWarningIndicatorRef` and the UDS DTC status bit 7 (WarningIndicator) is active (set to 1), otherwise it shall be set to “Lamp Off”. ]()

**[SWS\_Dem\_00892]** [ The DTC-specific “Flash Amber Warning Lamp” shall be set to “Unavailable / Do Not Flash” (0x03) in case the corresponding event has set “Lamp On” and the `DemIndicatorBehaviour` of the Indicator referenced by `DemAmberWarningIndicatorRef` is set to `DEM_INDICATOR_CONTINUOUS`. ]()

**[SWS\_Dem\_00893]** [ 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 `DemAmberWarningIndicatorRef` is set to `DEM_INDICATOR_SLOW_FLASH`. ](*SRS\_Diag\_04110*)

**[SWS\_Dem\_00894]** [ 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 `DemAmberWarningIndicatorRef` is set to `DEM_INDICATOR_FAST_FLASH`. ](*SRS\_Diag\_04110*)

### 7.9.3.4 Protect Lamp

**[SWS\_Dem\_00895]** [ The DTC-specific “Protect Lamp” returned in the function `Dem_J1939DcmGetNextDTCwithLampStatus` shall be set to “Lamp On” in case the corresponding event has assigned the Indicator referenced by `DemProtectLamplIndicatorRef` and the UDS DTC status bit 7 (WarningIndicator) is active (set to 1), otherwise it shall be set to “Lamp Off”. ](*SRS\_Diag\_04110*)

**[SWS\_Dem\_00896]** [ The DTC-specific “Flash Protect Lamp” shall be set to “Unavailable / Do Not Flash” (0x03) in case the corresponding event has set “Lamp On” and the `DemIndicatorBehaviour` of the Indicator referenced by `DemProtectLamplIndicatorRef` is set to `DEM_INDICATOR_CONTINUOUS`. ](*SRS\_Diag\_04069*)

**[SWS\_Dem\_00897]** [ 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 `DemProtectLamplIndicatorRef` is set to `DEM_INDICATOR_SLOW_FLASH`. ](*SRS\_Diag\_04110*)

**[SWS\_Dem\_00898]** [ 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 `DemProtectLamplIndicatorRef` is set to `DEM_INDICATOR_FAST_FLASH`. ](*SRS\_Diag\_04110*)

catorBehaviour of the Indicator referenced by [DemProtectLamIndicatorRef](#) is set to [DEM\\_INDICATOR\\_FAST\\_FLASH](#). ]()

#### 7.9.4 FreezeFrame

**[SWS\_Dem\_00899]** [ 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 criteria are [DEM\\_J1939DCM\\_FREEZEFRAME](#) (filtering the data in [DemJ1939FreezeFrameClassRef](#)) or [DEM\\_J1939DCM\\_EXPANDED\\_FREEZEFRAME](#) (filtering the data in [DemJ1939ExpandedFreezeFrameClassRef](#)). ]([SRS\\_Diag\\_04111](#))

**[SWS\_Dem\_00900]** [ Each function call of [Dem\\_J1939DcmGetNextFreezeFrame](#) shall step to the next (Expanded-)FreezeFrame for a specific node defined by the filter criteria (FreezeFrameKind and node) of [Dem\\_J1939DcmSetFreezeFrameFilter](#). In case no more (Expanded-)FreezeFrames are matching the filter criteria, the function shall return [DEM\\_FILTERED\\_NO\\_MATCHING\\_ELEMENT](#). The out parameters need not to be valid in this case. ]([SRS\\_Diag\\_04111](#))

**[SWS\_Dem\_00901]** [ 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\\_FREEZE FRAME](#). ]([SRS\\_Diag\\_04111](#))

**[SWS\_Dem\_00902]** [ 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\\_FILTERED\\_PENDING](#). The out parameter needs not to be valid in this case. The next call of [Dem\\_J1939DcmGetNextFreezeFrame](#) should continue after the interrupted element. ]([SRS\\_Diag\\_04111](#))

**[SWS\_Dem\_00903]** [ In case the function [Dem\\_J1939DcmGetNextFreezeFrame](#) has found a FreezeFrame, the Dem shall:

- Check if the buffer in the BufSize parameter is big enough to hold the (Expanded-)FreezeFrame. If not, [DEM\\_FILTERED\\_BUFFER\\_TOO\\_SMALL](#) shall be returned without any further actions. The out parameters need not to be valid in this case.
- Copy the (Expanded-)FreezeFrame data into the buffer provided by the parameter DestBuffer (in case of Expanded FreezeFrames without any SPN informations). Unused bits shall be filled with “0”.

- Set the parameter J1939DTC to the corresponding J1939DTC value (refer [DemDTCAttributes](#) for details) and the parameter OccurrenceCounter to the corresponding occurrence counter value.
- Return with DEM\_FILTERED\_OK. In case the occurrence counter is above +126 (0x7F), the returned value shall be set to +126 (0x7F).

]([SRS\\_Diag\\_04111](#))

#### 7.9.4.1 SPNs in ExpandedFreezeFrame

**[SWS\_Dem\_00904]** [ Each call to Dem\_J1939DcmGetNextSPNInFreezeFrame shall step to the next SPN of the ExpandedFreezeFrame definition for a specific node. In case no more SPN of the ExpandedFreezeFrame definition is available, the function return value shall be [DEM\\_FILTERED\\_NO\\_MATCHING\\_ELEMENT](#). The out parameter needs not to be valid in this case. ]([SRS\\_Diag\\_04111](#))

**[SWS\_Dem\_00905]** [ 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\\_FILTERED\\_PENDING](#). The out parameter needs not to be valid in this case. The next call of Dem\_J1939DcmGetNextSPNInFreezeFrame should continue after the interrupted element. ]([SRS\\_Diag\\_04111](#))

**[SWS\_Dem\_00906]** [ The function Dem\_J1939DcmGetNextSPNInFreezeFrame shall trigger the [Det](#) error [DEM\\_E\\_WRONG\\_CONDITION](#) in case of not supported FreezeFrameKind. Valid value is Dem\_SPNsInExpandedFreezeFrame. ]([SRS\\_Diag\\_04057](#), [SRS\\_Diag\\_04111](#))

**[SWS\_Dem\_00907]** [ In case Dem\_J1939DcmGetNextSPNInFreezeFrame has found an SPN of the ExpandedFreezeFrame definition, the Dem shall:

- Check if the buffer in the BufSize parameter is big enough to hold the (Expanded-)FreezeFrame. If not, [DEM\\_FILTERED\\_BUFFER\\_TOO\\_SMALL](#) shall be returned without any further actions. The out parameters need not to be valid in this case.
- Copy the (Expanded-)FreezeFrame data into the buffer provided by the parameter DestBuffer (in case of Expanded FreezeFrames without any SPN informations). Unused bits shall be filled with "0".
- Set the parameter J1939DTC to the corresponding J1939DTC value (refer [DemDTCAttributes](#) for details) and the parameter OccurrenceCounter to the corresponding occurrence counter value.
- Return with DEM\_FILTERED\_OK. In case the occurrence counter is above +126 (0x7F), the returned value shall be set to +126 (0x7F).

]([SRS\\_Diag\\_04111](#))

### 7.9.5 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]** [ 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]** [ A call of [Dem\\_J1939DcmReadDiagnosticReadiness1](#) shall report a response message based on the J1939-73 [15] 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\\_PreviosulyActive](#), [DEM\\_DTC\\_KIND\\_ALL\\_DTCS](#)).
- “OBD Compliance” shall be based on the configuration parameter [DemOBDCCompliance](#) (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]** [ A call of [Dem\\_J1939DcmReadDiagnosticReadiness2](#) shall report a response message based on the J1939-73 [15] 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]** [ A call of [Dem\\_J1939DcmReadDiagnosticReadiness3](#) shall report a response message based on the J1939-73 [15] 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 J193973 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 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).

]()

### 7.9.6 Monitor Performance Ratio

For J1939 the In-Use-Monitor Performance Ratio (IUMPR) requires a reporting for each supported Applicable System Monitor only.

**[SWS\_Dem\_00912]** [ 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]** [ Each call of the function [Dem\\_J1939DcmGetNextFilteredRatio](#) shall skip to the next valid SPN for a specific node with DemRatioid defined.

- The events referenced by this SPN via DemRatioid shall be used to calculate the return values SPN (for "SPN of Applicable System Monitor"), Numerator (for "Applicable System Monitor Numerator"), and Denominator (for "Applicable System Monitor Denominator"). The return value shall be [DEM\\_FILTERED\\_OK](#).
- In case the calculation needs to be interrupted due to internal implementations or limitations, the return value can be set to [DEM\\_FILTERED\\_PENDING](#). The out parameter needs not to be valid in this case. The next call of [Dem\\_J1939DcmGetNextFilteredRatio](#) should continue after the interrupted element.
- In case no more SPNs are available, the function return value shall be [DEM\\_FILTERED\\_NO\\_MATCHING\\_ELEMENT](#). The out parameter needs not to be valid in this case.

] ([SRS\\_Diag\\_04113](#))

## 7.10 Interaction with other Software Modules

### 7.10.1 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 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]** [ 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 chapters chapter [8.6.2.8](#), chapter [8.6.2.21](#), and chapter [8.6.2.3](#)), 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.

### 7.10.2 Interaction with Diagnostic Communication Manager (Dcm)

The [Dcm](#) accesses the [Dem](#) module in order to delegate those 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 [10, SWS Dcm]. There, especially the handling of freeze frame data is described.

**[SWS\_Dem\_00171]** [ 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. ]([SRS\\_Diag\\_04058](#))

**[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. ]([SRS\\_Diag\\_04010](#))

**[SWS\_Dem\_00827]** [ The [Dem](#) shall provide the API [Dem\\_DcmControlDTCStatusChangedNotification](#). ]([SRS\\_Diag\\_04010](#))

**[SWS\_Dem\_00828]** [ If the API [Dem\\_DcmControlDTCStatusChangedNotification](#) with state value TRUE and the parameter [DemTriggerDcmReports](#) is en-

abled, the `Dem` starts to report changes of the `DTC` status to the `Dcm` via `Dcm_DemTriggerOnDTCStatus`. ]()

**[SWS\_Dem\_00829]** [ If the API `Dem_DcmControlDTCStatusChangedNotification` with state value FALSE is enabled, the `Dem` stops to report changes of the `DTC` status to the `Dcm` via `Dcm_DemTriggerOnDTCStatus`. ](SRS\_Diag\_04105)

**[SWS\_Dem\_00830]** [ If the DTC status changed by clearing the `event memory` `Dcm_DemTriggerOnDTCStatus` shall not be called. ]()

Note: The configuration parameter `DemTriggerDcmReports` enables this function. The function `Dcm_DemTriggerOnDTCStatus` uses the same prototype as `<Module>_DemTriggerOnDTCStatus`. This configuration parameter is for simplicity (so the generic parameter `DemCallbackDTCStatusChangedFnc` needs not to be used).

### 7.10.2.1 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]** [ The function `Dem_DcmGetTranslationType` shall provide the capability to get the configured translation format of the ECU (refer to `DemTypeOfDTC-Supported` in `DemGeneral`). ]()

**[SWS\_Dem\_00060]** [ The function `Dem_DcmGetDTCStatusAvailabilityMask` shall provide the DTC status availability mask (refer to `DemDtcStatusAvailabilityMask` in `DemGeneral`), that means the DTC status information (according to ISO-14229-1[1]) supported by the Dem module with respect to the tester client with UDS service 0x19. ](SRS\_Diag\_04067, SRS\_Diag\_04010)

**[SWS\_Dem\_01181]** [ The Dem module shall be able to return the DTC severity availability mask (refer to `Dem_DcmGetDTCSeverityAvailabilityMask`) in accordance to ISO 14229-1 [3]. ](SRS\_Diag\_04141)

**[SWS\_Dem\_01182]** [ The function `Dem_DcmGetDTCSeverityAvailabilityMask` shall calculate the DTC severity availability mask in accordance to ISO 14229-1 [3] from the severity configuration of each supported DTC. ](SRS\_Diag\_04141)

**[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). ](SRS\_Diag\_04010)

**[SWS\_Dem\_00059]** [ The function `Dem_DcmGetStatusOfDTC` (refer to chapter 8.3.4.1.3) shall copy the status of a DTC to the API parameter `DTCStatus` according to ISO-14229-1[1]. ](SRS\_Diag\_04067)

It is possible, that a `DTC` with different states exists depending on the location. If the mirror memory is used as a kind of protocol stack that gives information which services have been performed on the primary memory, different `DTC` states might appear (e.g.

DTC has been deleted from the primary memory and is written to the mirror memory with its latest status).

**[SWS\_Dem\_00057]** [ The function [Dem\\_DcmSetDTCFilter](#) (refer to chapter [8.3.4.1.6](#)) shall set the filter criteria attributes to be used for the subsequent calls of [Dem\\_DcmGetNumberOfFilteredDTC](#), [Dem\\_DcmGetNextFilteredDTC](#), [Dem\\_DcmGetNextFilteredDTCAndFDC](#), as well as [Dem\\_DcmGetNextFilteredDTCAndSeverity](#). The filter criteria attributes shall be used until the next call of [Dem\\_DcmSetDTCFilter](#) or Dem initialization. ]([SRS\\_Diag\\_04057](#))

**[SWS\_Dem\_00649]** [ Each call of [Dem\\_DcmSetDTCFilter](#) shall lead to a reset of the sequence. ]([SRS\\_Diag\\_04057](#))

**[SWS\_Dem\_01058]** [ The function [Dem\\_DcmSetDTCFilter](#) shall ignore unsupported bits (refer to configuration parameter [DemDtcStatusAvailabilityMask](#)) retrieved in DTC-StatusMask. ]()

**[SWS\_Dem\_01065]** [ The Det error [DEM\\_E\\_WRONG\\_CONFIGURATION](#) shall be reported if the function [Dem\\_DcmSetDTCFilter](#) is called with a value of the parameter [DTCKind](#) that is not supported per configuration (e.g. if [DEM\\_DTC\\_KIND\\_EMISSION\\_REL\\_DTCS](#) is passed in [DTCKind](#), but no emission related DTCs are configured). ]()

**[SWS\_Dem\_01066]** [ The Det error [DEM\\_E\\_WRONG\\_CONFIGURATION](#) shall be reported if the function [Dem\\_DcmSetDTCFilter](#) 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\\_DcmSetDTCFilter](#) is called with a value of the parameter [DTCOrigin](#) that is not supported per configuration (e.g. if [DTCOrigin](#) "DEM\_DTC\_ORIGIN\_MIRROR\_MEMORY" is passed, but no mirror memory is configured). ]()

**[SWS\_Dem\_00061]** [ The function [Dem\\_DcmGetNumberOfFilteredDTC](#) (refer to chapter [8.3.4.1.7](#)) shall get the number of DTCs matching the filter criteria defined by the function call of [Dem\\_DcmSetDTCFilter](#) [[SWS\\_Dem\\_00057](#)]. ]([SRS\\_Diag\\_04000](#))

[Dem\\_DcmSetDTCFilter](#) has to be called prior to [Dem\\_DcmGetNumberOfFilteredDTC](#).

**[SWS\_Dem\_00216]** [ With each call to the function [Dem\\_DcmGetNextFilteredDTC](#) the Dem module shall return the next DTC and its associated status matching the filter criteria defined by the function call of [Dem\\_DcmSetDTCFilter](#) [[SWS\\_Dem\\_00057](#)]. ]([SRS\\_Diag\\_04010](#))

The Dcm calls the function [Dem\\_DcmGetNextFilteredDTC](#) continuously until the return value of the function is [DEM\\_FILTERED\\_NO\\_MATCHING\\_ELEMENT](#), to receive all DTCs matching the filter criteria.

**[SWS\_Dem\_00653]** [ The API [Dem\\_DcmGetNextFilteredDTC](#) shall not return the value [DEM\\_FILTERED\\_PENDING](#) when being used for emissionrelated services, indicated by any of these parameter combination:

- [DTCKind = DEM\\_DTC\\_KIND\\_EMISSION\\_REL\\_DTCS](#) and [DTCStatusMask = DEM\\_UDS\\_STATUS\\_CDTC](#)
- [DTCKind = DEM\\_DTC\\_KIND\\_EMISSION\\_REL\\_DTCS](#) and [DTCStatusMask = DEM\\_UDS\\_STATUS\\_PDTC](#)
- [DTCKind = DEM\\_DTC\\_KIND\\_EMISSION\\_REL\\_DTCS](#) and [DTCOrigin = DEM\\_DTC\\_ORIGIN\\_PERMANENT\\_MEMORY](#)

]()

Rationale: The API [Dem\\_DcmGetNextFilteredDTC](#) 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\\_DcmGetNextFilteredDTCAndFDC](#) 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\\_DcmSetDTCFilter](#) (refer to [SWS\_Dem\_00057]). ]([SRS\\_Diag\\_04010](#))

**[SWS\_Dem\_00513]** [ If the callback-function [GetFaultDetectionCounter](#) returns other than [E\\_OK](#) or is not configured (refer to configuration container [DemCallbackGetFDC](#)) for a [DTC](#) filtered by the function [Dem\\_DcmGetNextFilteredDTCAndFDC](#), this [DTC](#) shall not match the filter criteria (i.e. assume [FDC](#) is 0). ]()

The Dcm calls the function [Dem\\_DcmGetNextFilteredDTCAndFDC](#) continuously until the return value of the function is [DEM\\_FILTERED\\_NO\\_MATCHING\\_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 [GetFaultDetectionCounter](#), chapter 8.3.3.20).

**[SWS\_Dem\_00287]** [ With each call to the function [Dem\\_DcmGetNextFilteredDTCAndSeverity](#) (refer to chapter 8.3.4.1.10) the Dem module shall return the next DTC and its associated fault severity matching the filter criteria defined by the function call [Dem\\_DcmSetDTCFilter](#) (refer to [SWS\_Dem\_00057]). ]()

The Dcm calls the function [Dem\\_DcmGetNextFilteredDTCAndSeverity](#) continuously until the return value of the function is [DEM\\_FILTERED\\_NO\\_MATCHING\\_ELEMENT](#), to receive all DTCs matching the filter criteria.

For information about the functions [Dem\\_DcmGetSeverityOfDTC](#) and [Dem\\_DcmGetFunctionalUnitOfDTC](#), refer to chapter 7.4.

**[SWS\_Dem\_00595]** [ The function [Dem\\_DcmSetFreezeFrameRecordFilter](#) 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\\_DcmGetNextFilteredRecord](#). The filter criteria attributes shall be used until the next call of [Dem\\_DcmSetFreezeFrameRecordFilter](#) or Dem initialization. ]()

**[SWS\_Dem\_00650]** [ Each call of [Dem\\_DcmSetFreezeFrameRecordFilter](#) shall lead to a reset of the sequence. ]([SRS\\_Diag\\_04079](#))

**[SWS\_Dem\_00210]** [ The function [Dem\\_DcmSetFreezeFrameRecordFilter](#) shall retrieve the number of filtered freeze frame records. This filter always belongs to primary memory. ]([SRS\\_Diag\\_04010](#))

**[SWS\_Dem\_00225]** [ With each call to the function [Dem\\_DcmGetNextFilteredRecord](#) 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\\_DcmSetFreezeFrameRecordFilter](#). ]([SRS\\_Diag\\_04074](#))

The Dcm calls the function [Dem\\_DcmGetNextFilteredRecord](#) continuously until the return value of the function is [DEM\\_FILTERED\\_NO\\_MATCHING\\_ELEMENT](#), to receive all records matching the filter criteria.

**[SWS\_Dem\_00219]** [ The function [Dem\\_DcmGetDTCByOccurrenceTime](#) shall provide the capability to get one DTC stored in the primary [event memory](#) according to the API parameter DTCRequest, which specifies the relevant occurrence time. ]([SRS\\_Diag\\_04070](#))

**[SWS\_Dem\_00221]** [ The function [Dem\\_DcmGetDTCByOccurrenceTime](#) shall return [DEM\\_OCCURR\\_NOT\\_AVAILABLE](#), if no DTC is matching the requested occurrence time (DTCRequest). ]([SRS\\_Diag\\_04070](#))

### 7.10.2.2 Access event related data

This section defines the APIs, to get access to the event related data (refer to chapter [7.6.7.1](#) and chapter [7.6.7.3](#)) 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[1] and ISO 27145-3[22].

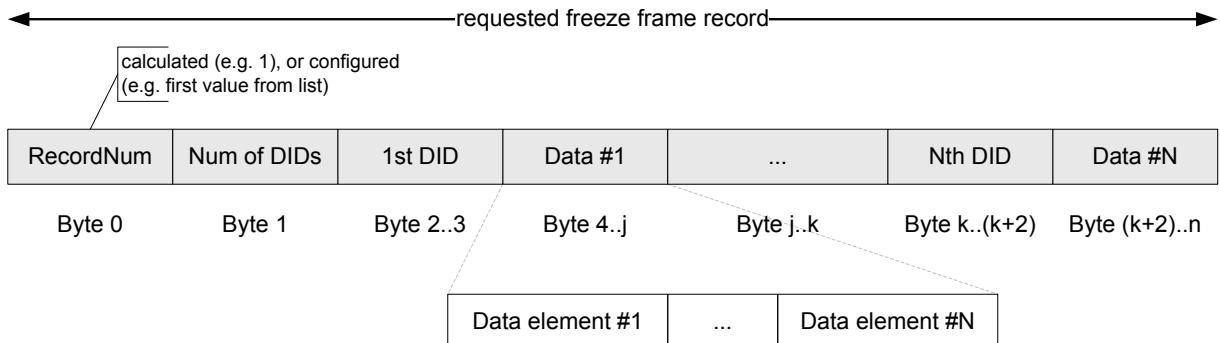
Example for the mapping of a PID to a DID: PID \$04 → DID 0xF404

Note: The data returned include the [DTCSnapshotRecordNumberOfIdentifiers](#) (Num of DIDs) as defined by ISO-14229-1[1] for the response message to service 0x19 0x05.

**[SWS\_Dem\_00071]** [ The function [Dem\\_DcmGetFreezeFrameDataByDTC](#) (refer to chapter [8.3.4.2.3](#)) shall copy the specific data identifiers (DIDs) of the requested freeze frame record to the destination buffer (DestBuffer). The function shall transmit these data as a complete record with the format shown in [Figure 7.50](#). ]([SRS\\_Diag\\_04074](#))

**[SWS\_Dem\_00576]** [ If the API parameter RecordNumber of [Dem\\_DcmGetFreezeFrameDataByDTC](#) is set to 0x00, the Dem module shall provide the event/DTC specific WWH-OBD equivalent freeze frame record. ]([SRS\\_Diag\\_04074](#))

Note: Retrieving record 0x00 with [Dem\\_DcmGetFreezeFrameDataByDTC](#) is only supported if the [Dem](#) module supports WWH-OBD (refer to [DemOBDSupport](#))



**Figure 7.50: Buffer format used by [Dem\\_DcmGetFreezeFrameDataByDTC](#)**

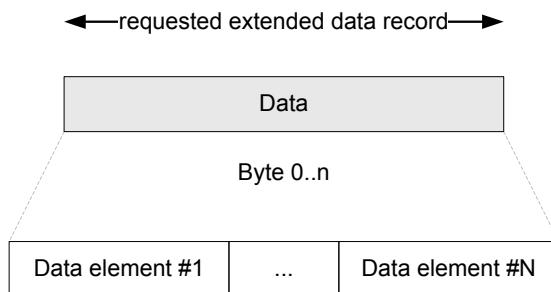
Note: The data returned include the DTCsnapshotRecordNumber (RecordNum) and DTCsnapshotRecordNumberofIdentifiers (Num of DIDs) as defined by ISO-14229-1[1] for the response message to service 0x19 0x04. In contradiction to the Dcm PIDstructure handling, the DIDstructure is handled within the Dem module. 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 [Dem\\_DcmGetFreezeFrameDataByDTC](#) is called with a valid DTC and a valid freeze frame record number which is not stored, the Dem shall return DEM\_GET\_FFDATABYDTC\_OK and BufSize 0 (empty buffer). ]([SRS\\_Diag\\_04074](#))

**[SWS\_Dem\_00074]** [ The function [Dem\\_DcmGetSizeOfFreezeFrameByDTC](#) (refer to chapter [8.3.4.2.4](#)) shall return the size of the requested freeze frame record(s), which represents the number of user data bytes, including any freeze frame headerinformation (according to the format defined in [\[SWS\\_Dem\\_00071\]](#)). ]([SRS\\_Diag\\_04074](#), [SRS\\_Diag\\_04079](#))

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\_00075]** [ The function [Dem\\_DcmGetExtendedDataRecordByDTC](#) (refer to chapter [8.3.4.2.5](#)) shall copy the complete data of the requested extended data record to the destination buffer (DestBuffer). The function shall transmit these data as a complete record with the format shown in Figure 7.51. ]([SRS\\_Diag\\_04074](#))



**Figure 7.51: Buffer format used by Dem\_DcmGetExtendedDataRecordByDTC**

**[SWS\_Dem\_00631]** [ If Dem\_DcmGetExtendedDataRecordByDTC is called with a valid DTC and a valid extended data record number which is not stored, the Dem shall return DEM\_RECORD\_OK and BufSize 0 (empty buffer). ] ([SRS\\_Diag\\_04074](#))

**[SWS\_Dem\_00076]** [ The function [Dem\\_DcmGetSizeOfExtendedDataRecordByDTC](#) (refer to chapter [8.3.4.2.6](#)) shall return the size of the requested extended data record(s), 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 [\[SWS\\_Dem\\_00075\]](#)). ] ([SRS\\_Diag\\_04074](#))

Note: If the record number value 0xFE is requested, the Dem considers the size of all OBD stored extended data records in the range of 0x90 to 0xEF according to [\[SWS\\_Dem\\_00076\]](#).

Note: If the record number value 0xFF is requested, the Dem considers the size of all stored extended data records (in the range of 0x01 to 0xEF) according to [\[SWS\\_Dem\\_00076\]](#).

The Dcm uses the function [Dem\\_DcmDisableDTCRecordUpdate](#), if the freeze frame or extended data of a specific DTC is about to be accessed by subsequent APIcalls. 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\\_DcmDisableDTCRecordUpdate](#) (refer to chapter [8.3.4.2.1](#)) shall protect the event related data of the specified DTC within the specified origin from updating or deleting, to allow a consistent read for the following subsequent APIcalls:

- Dem\_DcmGetSizeOfFreezeFrameByDTC and [Dem\\_DcmGetFreezeFrameDataByDTC](#)
- [Dem\\_DcmGetSizeOfExtendedDataRecordByDTC](#) and [Dem\\_DcmGetExtendedDataRecordByDTC](#).

] ()

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. Deminternal data elements).

Note: The function [Dem\\_DcmDisableDTCRecordUpdate](#) does not affect the DTC status information update.

**[SWS\_Dem\_00271]** [ The function [Dem\\_DcmEnableDTCRecordUpdate](#) (refer to chapter [8.3.4.2.2](#)) shall release the currently disabled DTC which has been protected by the function [Dem\\_DcmDisableDTCRecordUpdate](#), so that the data can be updated again. ]([SRS\\_Diag\\_04074](#))

The function [Dem\\_DcmEnableDTCRecordUpdate](#) is the counterpart to the function [Dem\\_DcmDisableDTCRecordUpdate](#).

The Dcm calls the function [Dem\\_DcmEnableDTCRecordUpdate](#) after the freeze frame and extended data of the specific DTC were protected by the function [Dem\\_DcmDisableDTCRecordUpdate](#), after the access by subsequent APIcalls is finished.

**[SWS\_Dem\_00648]** [ If development error detection is enabled and the function [Dem\\_DcmDisableDTCRecordUpdate](#) is called while another DTC is locked, the Dem module shall set the error code [DEM\\_E\\_WRONG\\_CONDITION](#) (refer also to [\[SWS\\_Dem\\_00370\]](#)). ]()

### 7.10.2.3 Clear diagnostic information

For definition of locking the clearing process, refer to chapter [7.6.2.2](#).

**[SWS\_Dem\_00009]** [ The [Dem](#) module shall provide the API [Dem\\_DcmClearDTC](#) to the [Dcm](#) for deleting single DTCs, DTC groups, and all DTCs from the [event memory](#). This shall trigger also initializing of related SWCs / BSW modules according to [\[SWS\\_Dem\\_00003\]](#). ]([SRS\\_Diag\\_04065](#))

The service ClearDiagnosticInformation (0x14) of ISO-14229-1[1] defines and covers the required actions and the deletion of related memory areas like freeze frames. One DTC value is transmitted, which can represent either a single DTC, or a group of DTCs. The groups are defined in ISO 14229-1, Annex D.1.

Note: [Dem\\_DcmClearDTC](#) typically triggers further callbacks through the RTE. To indicate the respective calltree for these runnables, a workaround is used: The Dcm triggers the DTC deletion using the Dem interface DcmIf (operation DcmClearDTC, refer to chapter chapter [8.6.2.14](#)) instead of a direct C call.

**[SWS\_Dem\_00077]** [ The function [Dem\\_DcmClearDTC](#) 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). ]([SRS\\_Diag\\_04065](#))

Note: Exceptions may apply due to [\[SWS\\_Dem\\_00514\]](#).

Note: This ensures that [Dem\\_DcmClearDTC](#) can react to calls from different services in a safe way (e.g. preemptive protocol changes from ISO-14229-1[1] service 0x14 to ISO 15031 service \$04).

**[SWS\_Dem\_01042]** [ The Dem module shall release the locked clearing process as soon as the internal processing is finished and the Dem is able to process the next clear request. ]([SRS\\_Diag\\_04130](#))

Note: The caller might not receive a final E\_OK from call of Dem\_<...>ClearDTC API in case it was aborted by another client. In this case E\_BUSY is finally returned and can be used as end criterion.

#### 7.10.2.4 Control DTC setting

**[SWS\_Dem\_00035]** [ The Dem module shall be capable of disabling (refer to [Dem\\_DcmDisableDTCSetting](#) ) and enabling (refer to [Dem\\_DcmEnableDTCSetting](#)) the [DTC](#) setting of a specific DTC group. ]()

Note: [Dem\\_DcmEnableDTCSetting](#) typically triggers further InitMonitorForEvent callbacks through the [RTE](#). To indicate the respective calltree for these runnables, a workaround is used: The Dcm triggers the enabling of DTC setting using the Dem interface Dcmlf (operation [DcmEnableDTCSetting](#)) instead of a direct C call.

**[SWS\_Dem\_00626]** [ When DTC setting is disabled, all status reports from SW-Cs (refer to [Dem\\_SetEventStatus](#)) and BSW modules (refer to [Dem\\_ReportErrorStatus](#)) for those events being assigned to this specific DTC group shall be ignored (no change of UDS DTC status byte) by the Dem. ]([SRS\\_Diag\\_04115](#))

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 7.31 and Figure 7.34).

**[SWS\_Dem\_00079]** [ The function [Dem\\_DcmDisableDTCSetting](#) shall disable the storage of all events assigned to a specific DTC group in the [event memory](#) including the respective UDS DTC status byte updates. ]([SRS\\_Diag\\_04010](#))

The function [Dem\\_DcmDisableDTCSetting](#) 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\\_DcmDisableDTCSetting](#) / [Dem\\_DcmEnableDTCSetting](#)) 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 DTC status byte updates, as these are also suppressed during disabled DTC setting.

**[SWS\_Dem\_00080]** [ The function [Dem\\_DcmEnableDTCSetting](#) shall enable the storage of all events assigned to specific a DTC group in the [event memory](#) including the respective UDS DTC status byte updates. ]([SRS\\_Diag\\_04000](#), [SRS\\_Diag\\_04115](#))

Note : If a DTC belongs to multiple DTC groups [e.g. EMISSION and NETWORK], the DTC has to be consider when processing any of the group.

#### 7.10.2.5 Asynchronous Dcm operations

Most of the APIs of the Dem/Dcm interface may depend on NVRAM data. Therefore, an asynchronous processing of these APIs can be realized by using the additional “PENDING” value of the respective API return types (refer to chapter [8.2.1.21](#)).

#### 7.10.3 Interaction with J1939 Diagnostic Manager (J1939)

**[SWS\_Dem\_00971]** [ In case the TestFailed bit in the event status changes and a J1939 DTC number is assigned to that event parameter, the function J1939Dcm\_DemTriggerOnDTCStatus shall be called. ]()

#### 7.10.4 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 event status information to the FiM.

The FiM uses the information of dependencies provided by the SWCs.

**[SWS\_Dem\_00029]** [ If [DemTriggerFimReports](#) (refer to [DemGeneral](#)) is enabled, the Dem module shall notify the FiM module (refer to [8]) on each event status change of all events (refer also to [\[SWS\\_Dem\\_00016\]](#)), by calling the function Fim\_DemTriggerOnEventStatus. ]([SRS\\_Diag\\_04031](#))

Note: The function Fim\_DemTriggerOnEventStatus is equal to [<Module>\\_DemTriggerOnEventStatus](#) (refer to chapter [8.4.3.2](#)). This configuration parameter is for simplicity to avoid a reference of the function to each individual EventId.

The Dem module provides the function [Dem\\_GetEventStatus](#) for possible plausibility checks of the FiM, rebuilding, startup, etc. of inhibition relations by the FiM.

**[SWS\_Dem\_01189]** [ In case the [Fim](#) is used, 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. ]([SRS\\_Diag\\_04031](#))

### 7.10.5 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 event 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]** [ 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). ]([SRS\\_Diag\\_04107](#))

Usually this verification is done in the API [Dem\\_Init](#) (refer to chapter [8.3.2.2](#)) 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]** [ 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\\_ReportErrorStatus](#)). These errors denote the defective NVRAM.

**[SWS\_Dem\_00340]** [ After the API [Dem\\_Init](#) has finished, the Dem shall be fully operational. ]([SRS\\_BSW\\_00101](#))

**[SWS\_Dem\_00550]** [ The Dem module shall provide the configuration parameter [DemImmediateNvStorage](#) per DTC (refer to [DemDTCAttributes](#)) to trigger the storage of the respective [event memory entry](#) including its event related data in nonvolatile memory (refer to [NvM\\_WriteBlock](#)) immediately. ]([SRS\\_BSW\\_00101](#))

**[SWS\_Dem\_00551]** [ If immediate nonvolatile storage is enabled for a specific DTC, the Dem module shall trigger the storage for new [event memory entries](#) and after every change of the event related data ([event memory entry](#) was updated). ]()

Note: For [event memory entries](#), which are stored immediately, it is necessary to ensure data consistency (e.g. with the event status byte) during [Dem\\_Init](#).

Note: If the immediate nonvolatile storage is disabled, the [event memory entry](#) and its event related data are stored persistently during the shutdown phase (refer to [\[SWS\\_Dem\\_00102\]](#), [\[SWS\\_Dem\\_00341\]](#) and the note below).

**[SWS\_Dem\_00552]** [ If immediate nonvolatile storage is enabled for a specific DTC, the Dem module shall not trigger further immediate write operations to NVRAM for this DTC, if its occurrence counter has reached the threshold defined by the configuration parameter [DemImmediateNvStorageLimit](#) (refer to [DemGeneral](#)). ]()

Note: Write operations to NVRAM will perform in any case at ECU shutdown.

**[SWS\_Dem\_00102]** [ 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. ]([SRS\\_BSW\\_00336](#))

**[SWS\_Dem\_00341]** [ For changed nonvolatile data, the Dem module shall trigger the storage to NVRAM before or during [Dem\\_Shutdown](#). ]([SRS\\_BSW\\_00336](#))

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]** [ 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](#). ]([SRS\\_Diag\\_04077](#))

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]** [ 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. ]([SRS\\_BSW\\_00171](#))

Note: If writing of nonvolatile Dem data fails, the Dem module is not able to perform any adequate reaction.

#### 7.10.6 Interaction with Default Error Tracer ([Det](#))

The interaction with the [Det](#) is described in chapter [7.12](#).

#### 7.10.7 Interaction with Diagnostic Log & Trace (Dlt)

**[SWS\_Dem\_00517]** [ If [DemTriggerDltReports](#) (refer to [DemGeneral](#)) is enabled, the Dem module shall notify the Dlt module (refer to [23]) on each event sta-

tus change of all events (refer also to SWS\_Dem\_00016), by calling the function Dlt\_DemTriggerOnEventStatus. ](SRS\_Diag\_04099)

Note: The function Dlt\_DemTriggerOnEventStatus is equal to <Module>\_DemTriggerOnEventStatus. This configuration parameter is for simplicity to avoid a reference of the function to each individual EventId.

After the Dlt module has been notified by Dlt\_DemTriggerOnEventStatus, it will make use of the functions Dem\_DltGetMostRecentFreezeFrameRecordData (refer to chapter 8.3.7.1) and Dem\_DltGetAllExtendedDataRecords to collect the current event related data.

**[SWS\_Dem\_00633]** [ The function Dem\_DltGetMostRecentFreezeFrameRecordData shall return the data of the most recent freeze frame record of the requested diagnostic event (refer to DemFreezeFrameClassRef). Note: This does not include the OBD-II or J1939 freeze frame. ](SRS\_Diag\_04099)

**[SWS\_Dem\_00992]** [ The format of the data in the destination buffer (DestBuffer) of the function Dem\_DltGetMostRecentFreezeFrameRecordData 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. ](SRS\_Diag\_04099)

**[SWS\_Dem\_00634]** [ If DemTriggerDltReports is enabled, the Dem module shall provide access on all extended data records (refer to Dem\_DltGetAllExtendedDataRecords) to the Dlt module. ](SRS\_Diag\_04099)

**[SWS\_Dem\_00635]** [ The function Dem\_DltGetAllExtendedDataRecords shall report the data of all extended data records of the requested diagnostic event. ](SRS\_Diag\_04099)

**[SWS\_Dem\_00993]** [ The format of the data in the destination buffer (DestBuffer) of the function Dem\_DltGetAllExtendedDataRecords is raw hexadecimal values and contains no header information like RecordNumber. The size of the buffer equals to the configuration settings of all respective data elements. ](SRS\_Diag\_04099)

## 7.10.8 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 7.6.7).

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 implementation specific 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 Version check

For details refer to the chapter 5.1.8 “Version Check” in SWS\_BSWGeneral.

## 7.12 Error classification

### 7.12.1 Development Errors

**[SWS\_Dem\_00173]** [ The errors shown in table 7.7 shall be detectable by the Dem module depending on its configuration (development / production mode). ] (SWS\_BSW\_00050, SWS\_BSW\_00212)

Type or error	Relevance	Related error code	Value [hex]
API function called with a parameter value, which is not allowed by active configuration	Development	DEM_E_WRONG_CONFIGURATION	0x10
API function called with a NULL pointer	Development	DEM_E_PARAM_POINTER	0x11
API function called with wrong parameter value	Development	DEM_E_PARAM_DATA	0x12
API function called with wrong length parameter value	Development	DEM_E_PARAM_LENGTH	0x13
Dem initialisation failed (refer to SWS_BSW_00151)	Development	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])	Development	DEM_E_UNINIT	0x20
No valid data available by the SW-C	Development	DEM_E_NODATAAVAILABLE	0x30
The requested record number is not supported by the event.	Development	DEM_E_WRONG_RECORDNUMBER	0x31
The requested DID is not supported by the freeze frame	Development	DEM_E_WRONG_DIDNUMBER	0x32
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]).	Development	DEM_E_WRONG_CONDITION	0x40

**Table 7.7: Types of errors which can be detected by the Dem module**

**[SWS\_Dem\_00124]** If development error detection is enabled and any instance calls any Dem API, excluding `Dem_ReportErrorStatus`, `Dem_ResetEventDebounceStatus` and `Dem_GetVersionInfo`, before the Dem was fully initialized, the Dem module shall set the error code `DEM_E_UNINIT`. *(SRS\_BSW\_00406)*

Note: If development error detection is disabled and the Dem is not fully initialized, the behavior of the APIs is undefined.

**[SWS\_Dem\_00364]** If development error detection is enabled and any instance calls `Dem_ReportErrorStatus` or `Dem_ResetEventDebounceStatus` before the Dem was preinitialized, the Dem module shall set the error code `DEM_E_UNINIT`. *(SRS\_BSW\_00406)*

Note: If development error detection is disabled and the Dem is not preinitialized, the behavior of these APIs is undefined.

**[SWS\_Dem\_00368]** If development error detection is enabled and any instance calls any Dem API, excluding `Dem_ReportErrorStatus`, `Dem_ResetEventDebounceStatus` and `Dem_GetVersionInfo`, after `Dem_Shutdown` (refer to chapter 8.3.2.3) 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]** 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`. *(SRS\_Diag\_04057)*

Note: For example, [Dem\\_DcmGetNextFilteredDTCAndFDC](#) is called, after [Dem\\_DcmSetDTCFilter](#) with [FilterForFaultDetectionCounter = FALSE](#) was called.

**[SWS\_Dem\_00370]** [ If development error detection is enabled and a [Dem](#) function detects a development error, excluding [DEM\\_E\\_NODATAAVAILABLE](#), then the [Dem](#) function shall return immediately with [E\\_NOT\\_OK](#) (in case of [Std\\_ReturnType](#)) or an appropriate negative return value, after the development error was reported. ]()

### 7.12.2 Runtime Errors

There are no runtime errors.

### 7.12.3 Transient Faults

There are no transient faults.

### 7.12.4 Production Errors

There are no production errors.

### 7.12.5 Extended Production Errors

There are no extended production errors.

## 7.13 Error detection

For details refer to the chapter 7.2.3.2 “Configuration of Development Errors” in [SWS\\_BSWGeneral](#).

## 7.14 Error notification

For details refer to the chapter 7.2.3.3 “Reporting Development Errors” in [SWS\\_BSWGeneral](#).

## 7.15 Support for Debugging

For details refer to the chapter 7.1.17 “Debugging support” in [SWS\\_BSWGeneral](#).

## 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.

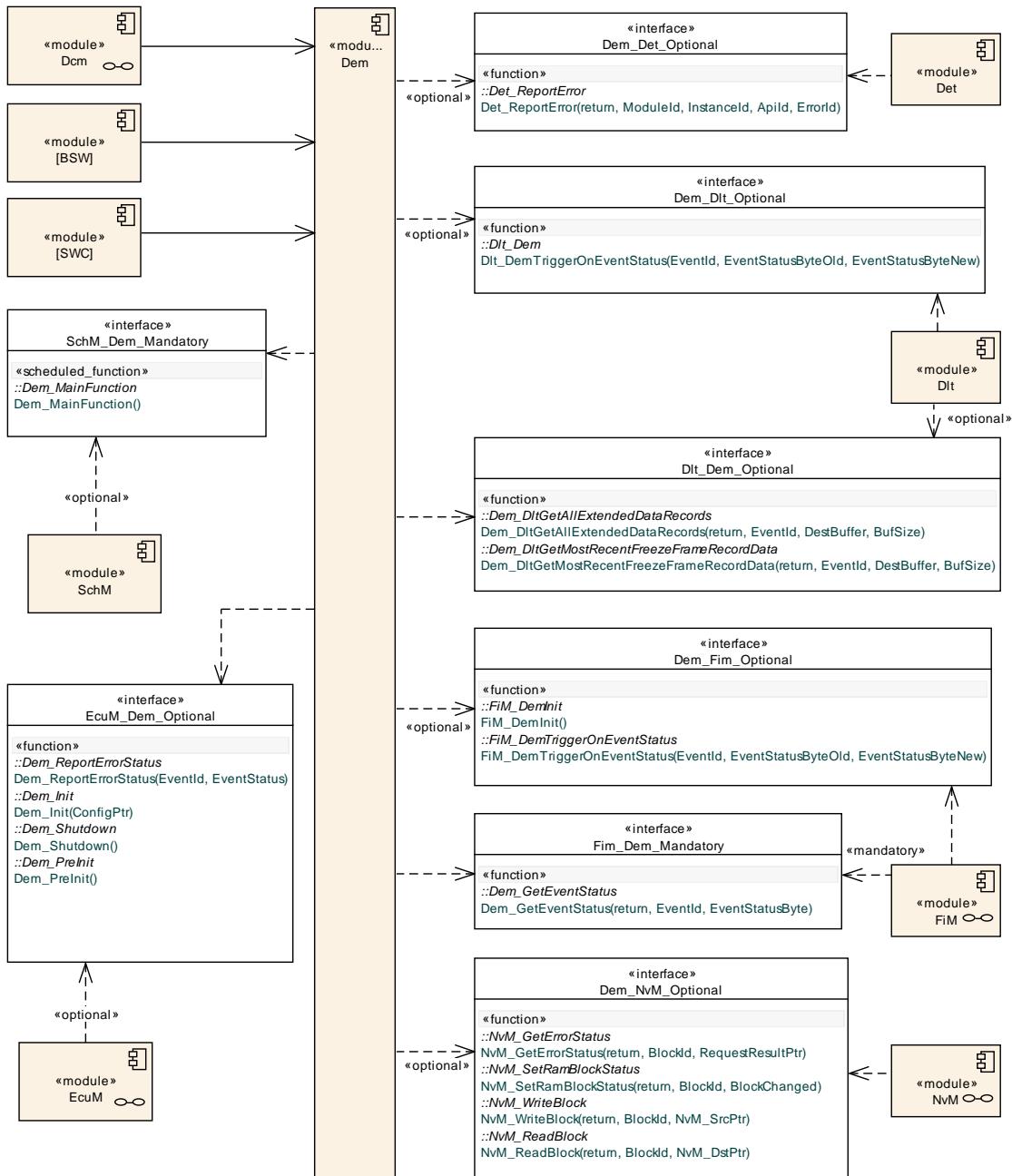
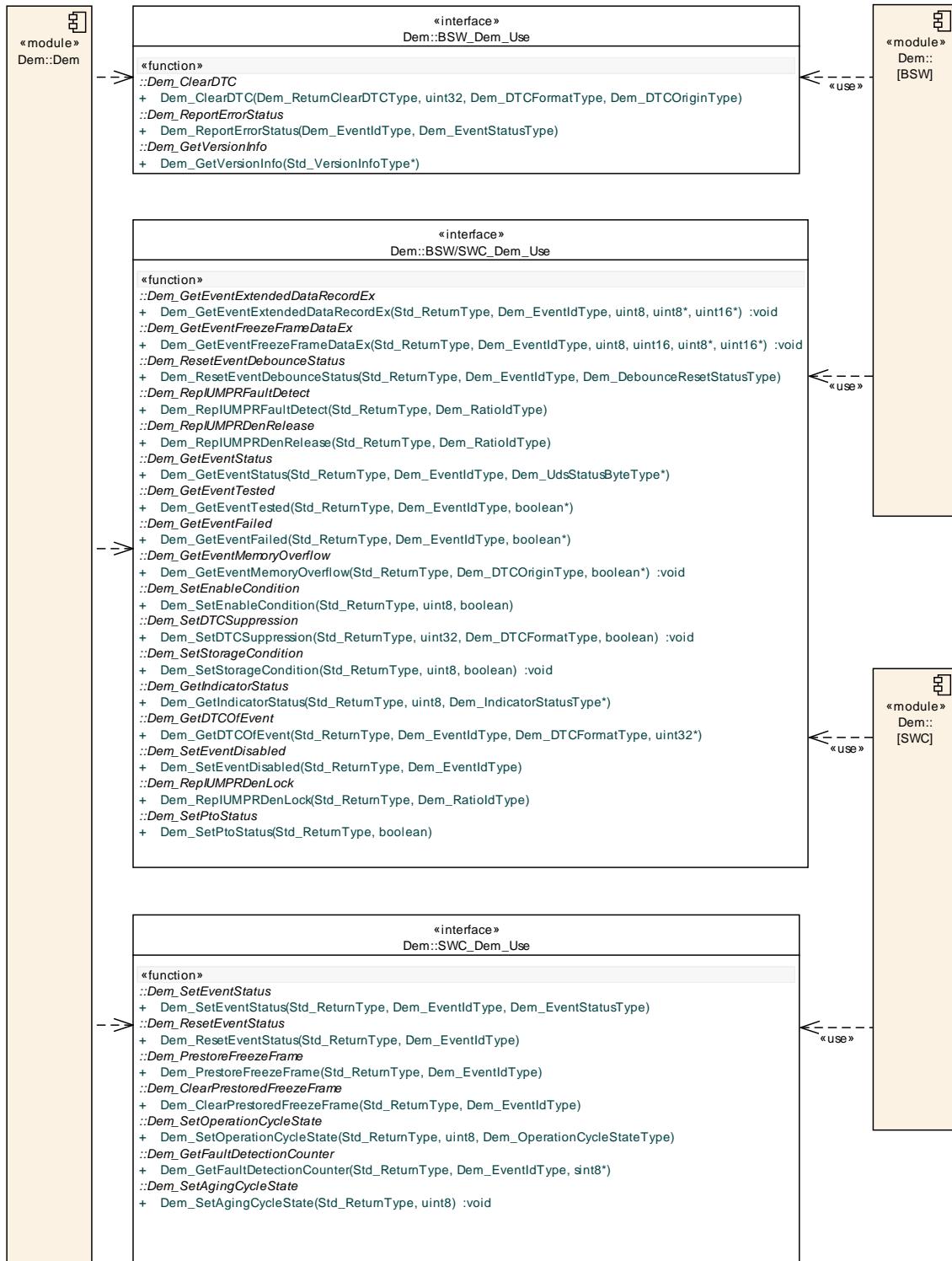
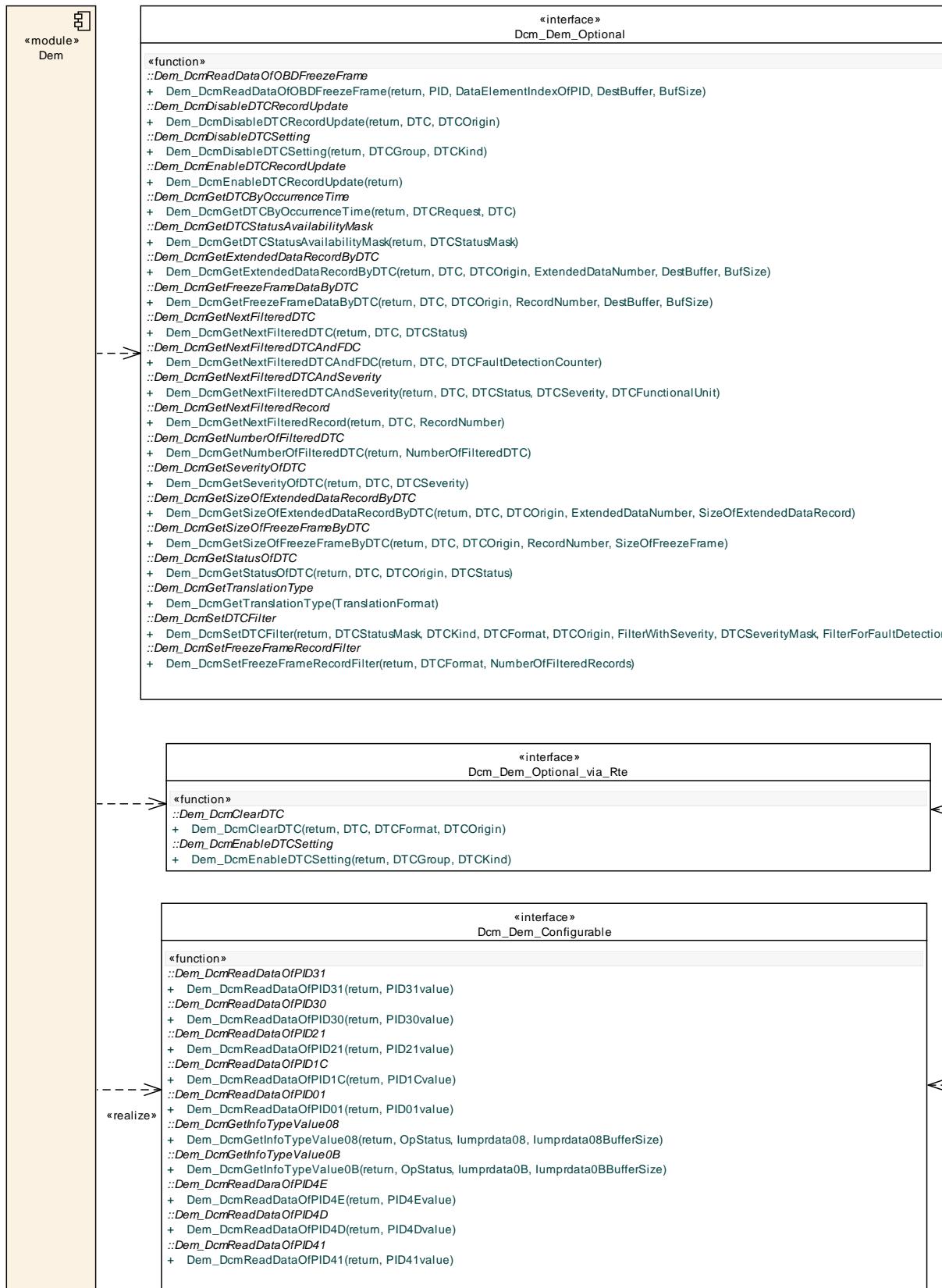


Figure 8.1: Overview of interfaces between the Dem and other BSW modules


**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] [

<b>Module</b>	<b>Imported Type</b>
Dcm	Dcm_OpStatusType
NvM	NvM_BlockIdType NvM_RequestResultType
Std_Types	Std_ReturnType Std_VersionInfoType

**Table 8.1: Dem\_ImportedTypes**

] (SRS\_BSW\_00301)

## 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] [

<b>Name:</b>	Dem_ComponentIdType		
<b>Type:</b>	uint16		
<b>Range:</b>	1 .. 65535	-	Internal identifier of a monitored component. Remark: 0 is not a valid value
<b>Description:</b>	Identification of a DemComponent by assigned ComponentId. The ComponentId is automatically assigned by the Dem.		

**Table 8.2: Dem\_ComponentIdType**

] (SRS\_Diag\_04142)

#### 8.2.1.2 Dem\_ConfigType

[SWS\_Dem\_00924] [

<b>Name:</b>	Dem_ConfigType
<b>Type:</b>	Structure

<b>Element:</b>		implementation specific	-
<b>Description:</b>	This type of the external data structure shall contain the post build initialization data for the Dem.		

**Table 8.3: Dem\_ConfigType**

]()

### 8.2.1.3 Dem\_EventIdType

[SWS\_Dem\_00925] [

<b>Name:</b>	Dem_EventIdType		
<b>Type:</b>	uint16		
<b>Range:</b>	1 .. 65535	-	Internal identifier of a diagnostic event Remark: 0 is not a valid value
<b>Description:</b>	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.		

**Table 8.4: Dem\_EventIdType**

]()

### 8.2.1.4 Dem\_EventStatusType

[SWS\_Dem\_00926] [

<b>Name:</b>	Dem_EventStatusType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_EVENT_STATUS_P_ASSED DEM_EVENT_STATUS_F_AILED DEM_EVENT_STATUS_P_REPASSED	0x00 0x01 0x02	Monitor reports qualified test result passed. Monitor reports qualified test result failed. Monitor reports non-qualified test result pre-passed (debounced Dem-internally).
	DEM_EVENT_STATUS_P_REFAILED	0x03	Monitor reports non-qualified test result pre-failed (debounced Dem-internally).

	DEM_EVENT_STATUS_F DC_THRESHOLD_REACHED	0x04  0x05 – 0xFF	Monitor triggers the storage of Extended-DataRecords and Freeze-Frames ON_FDC_THRESHOLD. reserved
<b>Description:</b>	This type contains all monitor test result values, which can be reported via Dem_ReportErrorStatus() and Dem_SetEventStatus().		

**Table 8.5: Dem\_EventStatusType**

↴()

### 8.2.1.5 Dem\_DebouncingStateType

[SWS\_Dem\_01000] ↴

<b>Name:</b>	Dem_DebouncingStateType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_TEMPORARILY_DEFECTIVE	0x01	Bit 0: Temporarily Defective (corresponds to 0 < FDC < 127)
	DEM_FINALY_DEFECTIVE	0x02	Bit 1: finally Defective (corresponds to FDC = 127)
	DEM_TEMPORARILY_HEALED	0x04	Bit 2: temporarily healed (corresponds to -128 < FDC < 0)
	DEM_TEST_COMPLETE	0x08	Bit 3: Test complete (corresponds to FDC = -128 or FDC = 127)
	DEM_DTR_UPDATE	0x10	Bit 4: DTR Update (= Test complete && Debouncing complete && enable conditions / storage conditions fulfilled)
<b>Description:</b>	–		

**Table 8.6: Dem\_DebouncingStateType**

↴()

### 8.2.1.6 Dem\_DebounceResetStatusType

[SWS\_Dem\_00927] ↴

<b>Name:</b>	Dem_DebounceResetStatusType
<b>Type:</b>	uint8

<b>Range:</b>	DEM_DEBOUNCE_STATUS_FREEZE DEM_DEBOUNCE_STATUS_RESET	0x00 0x01 0x02 – 0xFF	Freeze the internal debounce counter/timer. Reset the internal debounce counter/timer. reserved
<b>Description:</b>	This type contains all definitions to control an internal debounce counter/timer via the function Dem_ResetEventDebounceStatus().		

**Table 8.7: Dem\_DebounceResetStatusType**

↴()

### 8.2.1.7 Dem\_UdsStatusByteType

[SWS\_Dem\_00928] ↵

<b>Name:</b>	Dem_UdsStatusByteType		
<b>Type:</b>	uint8		
<b>Range:</b>	range  DEM_UDS_STATUS_TF DEM_UDS_STATUS_TFT OC DEM_UDS_STATUS_PDT C DEM_UDS_STATUS_CDT C DEM_UDS_STATUS_TNC SLC DEM_UDS_STATUS_TFS LC DEM_UDS_STATUS_TNC TOC DEM_UDS_STATUS_WIR	0x00..0xFF  0x01 0x02 0x04 0x08 0x10 0x20 0x40 0x80	the possible range from 0x00 to 0xFF allows for all combinations of the individual status bits  bit 0: TestFailed bit 1: TestFailedThisOperationCycle bit 2: PendingDTC  bit 3: ConfirmedDTC  bit 4: TestNotCompletedSinceLastClear bit 5: TestFailedSinceLastClear bit 6: TestNotCompletedThisOperationCycle bit 7: WarningIndicatorRequested
<b>Description:</b>	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.		

**Table 8.8: Dem\_UdsStatusByteType**

↴()

### 8.2.1.8 Dem\_OperationCycleStateType

[SWS\_Dem\_00929] ↗

<b>Name:</b>	Dem_OperationCycleStateType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_CYCLE_STATE_START DEM_CYCLE_STATE_ENDD	0x00 0x01	Start/restart the operation cycle End the operation cycle
<b>Description:</b>	This type contains operation cycle state values, which can be reported via Dem_SetOperationCycleState() / Dem_GetOperationCycleState().		

**Table 8.9: Dem\_OperationCycleStateType**

↳

### 8.2.1.9 Dem\_IndicatorStatusType

[SWS\_Dem\_00930] ↗

<b>Name:</b>	Dem_IndicatorStatusType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_INDICATOR_OFF DEM_INDICATOR_CONTINUOUS DEM_INDICATOR_BLINKING DEM_INDICATOR_BLINKING_CONT DEM_INDICATOR_SLOW_FLASH DEM_INDICATOR_FAST_FLASH DEM_INDICATOR_ON_DEMAND DEM_INDICATOR_SHORT	0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07	Indicator off mode Indicator continuously on mode Indicator blinking mode Indicator blinking or continuously on mode Indicator slow flashing mode Indicator fast flashing mode Indicator on-demand mode Indicator short mode
<b>Description:</b>	Indicator mode used by Dem_GetIndicatorStatus().		

**Table 8.10: Dem\_IndicatorStatusType**

↳

### 8.2.1.10 Dem\_DTCKindType

[SWS\_Dem\_00932] ↗

<b>Name:</b>	Dem_DTCKindType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_DTC_KIND_ALL_DTCS DEM_DTC_KIND_EMISSION_REL_DTCS	0x01 0x02	Select all DTCs Select OBD-relevant DTCs
<b>Description:</b>	This type is used to filter DTCs for their kind.		

**Table 8.11: Dem\_DTCKindType**

↴()

### 8.2.1.11 Dem\_DTCFormatType

[SWS\_Dem\_00933] ↴

<b>Name:</b>	Dem_DTCFormatType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_DTC_FORMAT_OBD DEM_DTC_FORMAT_UDS DEM_DTC_FORMAT_J1939	0 1 2	selects the 2-byte OBD DTC format (refer to configuration parameter DemObdDTC) selects the 3-byte UDS DTC format (refer to configuration parameter DemUdsDTC) selects the merged SPN + FMI to 3-byte J1939 DTC format (refer to DemJ1939DTC)
<b>Description:</b>	This type is used to select the format of the DTC value.		

**Table 8.12: Dem\_DTCFormatType**

↴()

### 8.2.1.12 Dem\_DTCOriginType

[SWS\_Dem\_00934] ↴

<b>Name:</b>	Dem_DTCOriginType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_DTC_ORIGIN_PRI_MARY_MEMORY DEM_DTC_ORIGIN_MIR_ROR_MEMORY	0x01 0x02	Event information located in the primary memory Event information located in the mirror memory

	DEM_DTC_ORIGIN_PERMANENT_MEMORY	0x03	The Event information is located in the permanent memory
	DEM_DTC_ORIGIN_USERDEFINED_MEMORY_XX	0xXX	Event information located in the user defined memory, where XX is the configured DemUserDefinedMemoryIdentifier in hexadecimal
<b>Description:</b>	This enum is used to define the location of the events. The definition and use of the different memory types is OEM-specific.		

**Table 8.13: Dem\_DTCOriginType**

↴()

### 8.2.1.13 Dem\_DTCRequestType

[SWS\_Dem\_00935] ↴

<b>Name:</b>	Dem_DTCRequestType		
<b>Type:</b>	uint8		
<b>Range:</b>	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
<b>Description:</b>	This type is used to request a DTC with specific attributes.		

**Table 8.14: Dem\_DTCRequestType**

↴()

### 8.2.1.14 Dem\_DTCTranslationFormatType

[SWS\_Dem\_00936] ↴

<b>Name:</b>	Dem_DTCTranslationFormatType		
<b>Type:</b>	uint8		
<b>Range:</b>	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_N_SAEJ1939_73	0x02	SAEJ1939-73 DTC format
	DEM_DTC_TRANSLATION_N_ISO11992_4	0x03	ISO11992-4 DTC format
	DEM_DTC_TRANSLATION_J2012DA_FORMAT_04	0x04	SAE_J2012-DA_DTCFormat_04 DTC format
<b>Description:</b>	DTC translation format as defined in ISO14229-1 Service 0x19 returned by Dem_DcmGetTranslationType().		

**Table 8.15: Dem\_DTCTranslationFormatType**

]()

### 8.2.1.15 Dem\_DTCSeverityType

[SWS\_Dem\_00937] [

Name:	Dem_DTCSeverityType		
Type:	uint8		
Range:	DEM_SEVERITY_NO_SEVERITY	0x00	No severity information available
	DEM_SEVERITY_WWHOBD_CLASS_NO_CLASS	0x01	No class information
	DEM_SEVERITY_WWHOBD_CLASS_A	0x02	WWH-OBD Class A
	DEM_SEVERITY_WWHOBD_CLASS_B1	0x04	WWH-OBD Class B1
	DEM_SEVERITY_WWHOBD_CLASS_B2	0x08	WWH-OBD Class B2
	DEM_SEVERITY_WWHOBD_CLASS_C	0x10	WWH-OBD Class C
	DTC_CLASS	0x1f	Selects one of the defined DTC classes (or "NO_CLASS")
	DEM_SEVERITY_MAINTENANCE_ONLY	0x20	maintenance required
	DEM_SEVERITY_CHECK_AT_NEXT_HALT	0x40	check at next halt
	DEM_SEVERITY_CHECK_IMMEDIATELY	0x80	Check immediately
<b>Description:</b>	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.		

**Table 8.16: Dem\_DTCSeverityType**

]()

### 8.2.1.16 Dem\_RatioldType

[SWS\_Dem\_00940] ↴

<b>Name:</b>	Dem_RatioldType		
<b>Type:</b>	uint8, uint16		
<b>Range:</b>	0 .. 255, 0 .. 65535	-	Configurable, size depends on system complexity (refer to range of configuration parameter DemRatiold)
<b>Description:</b>	OBD specific ratio Id (related to a specific event, a FID, and an IUMPR group). This type depends on the Dem configuration.		

Table 8.17: Dem\_RatioldType

]()

### 8.2.1.17 Dem\_DTRControlType

[SWS\_Dem\_00941] ↴

<b>Name:</b>	Dem_DTRControlType		
<b>Type:</b>	uint8		
<b>Range:</b>	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.
<b>Description:</b>	Control parameter for the interpretation of the reported test results.		

Table 8.18: Dem\_DTRControlType

]()

### 8.2.1.18 Dem\_InitMonitorReasonType

[SWS\_Dem\_00942] ↴

<b>Name:</b>	Dem_InitMonitorReasonType		
<b>Type:</b>	uint8		
<b>Range:</b>	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 (re-)started.
	DEM_INIT_MONITOR_REENABLED	0x03	Enable conditions or DTC settings re-enabled.
	DEM_INIT_MONITOR_STORAGEREENABLED	0x04	Storage condition reenabled.
<b>Description:</b>	(Re-)Initialization reason returned by the callback <Module>_DemInitMonitorFor<EventName>().		

Table 8.19: Dem\_InitMonitorReasonType

]()

### 8.2.1.19 Dem\_IumprDenomCondIdType

[SWS\_Dem\_00943] ↴

<b>Name:</b>	Dem_IumprDenomCondIdType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_IUMPR_GENERAL_DENOMINATOR	0x01	IUMPR denominator condition "General Denominator"
	DEM_IUMPR_DEN_COND_COLDSTART	0x02	Additional IUMPR denominator condition "Cold Start"
	DEM_IUMPR_DEN_COND_EVAP	0x03	Additional IUMPR denominator condition "EVAP"
	DEM_IUMPR_DEN_COND_500MI	0x04	Additional IUMPR denominator condition "500 miles"
<b>Description:</b>	This type contains all possible additional IUMPR denominator conditions to be broadcasted among OBD-relevant ECUs.		

Table 8.20: Dem\_IumprDenomCondIdType

]()

### 8.2.1.20 Dem\_IumprDenomCondStatusType

[SWS\_Dem\_00944] [

<b>Name:</b>	Dem_IumprDenomCondStatusType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_IUMPR_DEN_STAT_US_NOT_REACHED	0x00	Condition of IUMPR-Denominator given by IUMPRDenCondId is not met (yet).
	DEM_IUMPR_DEN_STAT_US_REACHED	0x01	Condition of IUMPR-Denominator given by IUMPRDenCondId is met
	DEM_IUMPR_DEN_STAT_US_INHIBITED	0x02	Condition of IUMPR-Denominator given by IUMPRDenCondId is inhibited and cannot be reached.
	0x03 – 0xFF		reserved
<b>Description:</b>	This type contains all possible states of an additional IUMPR denominator condition to be broadcasted among OBD-relevant ECUs.		

**Table 8.21: Dem\_IumprDenomCondStatusType**

]()

### 8.2.1.21 Dem\_J1939DcmDTCStatusFilterType

[SWS\_Dem\_00945] [

<b>Name:</b>	Dem_J1939DcmDTCStatusFilterType		
<b>Type:</b>	uint8		
<b>Range:</b>	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
<b>Description:</b>	The type to distinguish which DTCs should be filtered.		

**Table 8.22: Dem\_J1939DcmDTCStatusFilterType**

]()

### 8.2.1.22 Dem\_J1939DcmSetClearFilterType

[SWS\_Dem\_00946] ↴

<b>Name:</b>	Dem_J1939DcmSetClearFilterType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_J1939DTC_CLEAR_ALL DEM_J1939DTC_CLEAR_PREVIOUSLY_ACTIVE	0 1	active DTCs previously active DTCs
<b>Description:</b>	The type to distinguish which DTCs gets cleared		

Table 8.23: Dem\_J1939DcmSetClearFilterType

↳()

### 8.2.1.23 Dem\_J1939DcmSetFreezeFrameFilterType

[SWS\_Dem\_00947] ↴

<b>Name:</b>	Dem_J1939DcmSetFreezeFrameFilterType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_J1939DCM_FREEZEFRAME DEM_J1939DCM_EXPANDED_FREEZEFRAME DEM_J1939DCM_SPONS_IN_EXPANDED_FREEZEFRAME	0 1 2	FreezeFrame (DM04) ExpandedFreezeFrame (DM25) SPNs in Expanded-FreezeFrame (DM24)
<b>Description:</b>	The type to distinguish which DTCs gets cleared		

Table 8.24: Dem\_J1939DcmSetFreezeFrameFilterType

↳()

### 8.2.1.24 Dem\_J1939DcmLampStatusType

[SWS\_Dem\_00948] ↴

<b>Name:</b>	Dem_J1939DcmLampStatusType		
<b>Type:</b>	uint16		
<b>Range:</b>	bits 8-7 bits 6-5 bits 4-3 bits 2-1	- - - -	Malfunction Indicator Lamp Status Red Stop Lamp Status Amber Warning Lamp Status Protect Lamp Status

	bits 8-7	-	Flash Malfunction Indicator Lamp
	bits 6-5	-	Flash Red Stop Lamp
	bits 4-3	-	Flash Amber Warning Lamp
	bits 2-1	-	Flash Protect Lamp
<b>Description:</b>	For details refer SAE J1939-73		

**Table 8.25: Dem\_J1939DcmLampStatusType**

]()

### 8.2.1.25 Dem\_J1939DcmDiagnosticReadiness1Type

[SWS\_Dem\_00949] [

<b>Name:</b>	Dem_J1939DcmDiagnosticReadiness1Type		
<b>Type:</b>	Structure		
<b>Element:</b>	uint8 uint8  uint8 uint8  uint16  uint16	ActiveTroubleCodes PreviouslyActive DiagnosticTrouble Codes  OBDCOMpliance Continuously MonitoredSystems Support_Status NonContinuously MonitoredSystems Support  NonContinuously MonitoredSystems Status	Number of active DTCs Number of previously active DTCs  OBD Compliance Identifies the continuously monitored system support and status Identifies the non-continuously monitored systems support Identifies the non-continuously monitored systems status
<b>Description:</b>	This structure represents all data elements of the DM05 message. The encoding shall be done according SAE J1939-73		

**Table 8.26: Dem\_J1939DcmDiagnosticReadiness1Type**

]()

### 8.2.1.26 Dem\_J1939DcmDiagnosticReadiness2Type

[SWS\_Dem\_00950] [

<b>Name:</b>	Dem_J1939DcmDiagnosticReadiness2Type		
<b>Type:</b>	Structure		
<b>Element:</b>	uint16	DistanceTraveled WhileMILis Activated	The kilometers accumulated while the MIL is activated

	uint16	DistanceSinceDTCsCleared	Distance accumulated since emission related DTCs were cleared
	uint16	MinutesRunbyEngineWhileMILisActivated	Accumulated count (in minutes) while the MIL is activated (on)
	uint16	TimeSinceDiagnosticTroubleCodesCleared	Engine running time accumulated since emission related DTCs were cleared
<b>Description:</b>	This structure represents all data elements of the DM21 message. The encoding shall be done according SAE J1939-73		

**Table 8.27: Dem\_J1939DcmDiagnosticReadiness2Type**

↴()

### 8.2.1.27 Dem\_J1939DcmDiagnosticReadiness3Type

[SWS\_Dem\_00951] ↵

Name:	Dem_J1939DcmDiagnosticReadiness3Type		
Type:	Structure		
Element:	uint16	TimeSinceEngineStart	Time since key-on that the engine has been running.
	uint8	NumberofWarmupsSinceDTCsCleared	Number of OBD warm-up cycles since all DTCs were cleared
	uint8	ContinuouslyMonitoredSystemsEnableCompletedStatus	Identifies the continuously monitored system enable/completed support and status.
	uint16	NonContinuouslyMonitoredSystemsEnableStatus	Enable status of non-continuous monitors this monitoring cycle
	uint16	NonContinuouslyMonitoredSystems	Completion status of non-continuous monitors this monitoring cycle
<b>Description:</b>	This structure represents all data elements of the DM26 message. The encoding shall be done according SAE J1939-73		

**Table 8.28: Dem\_J1939DcmDiagnosticReadiness3Type**

↴()

## 8.2.2 Dem return types

### 8.2.2.1 Dem\_ReturnGetStatusOfDTCType

[SWS\_Dem\_00952] ↗

<b>Name:</b>	Dem_ReturnGetStatusOfDTCType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_STATUS_OK DEM_STATUS_WRONG_DTC DEM_STATUS_WRONG_DTCORIGIN DEM_STATUS_FAILED DEM_STATUS_PENDING	0x00 0x01 0x02 0x03 0x04	Status of DTC is OK DTC value not existing (in this format) Wrong DTC origin  DTC failed The requested value is calculated asynchronously and currently not available. The caller can retry later.
<b>Description:</b>	Used to return the status of Dem_DcmGetStatusOfDTC.		

Table 8.29: Dem\_ReturnGetStatusOfDTCType

]()

### 8.2.2.2 Dem\_ReturnGetSeverityOfDTCType

[SWS\_Dem\_00953] ↗

<b>Name:</b>	Dem_ReturnGetSeverityOfDTCType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_GET_SEVERITYOF_DTC_OK DEM_GET_SEVERITYOF_DTC_WRONG_DTC DEM_GET_SEVERITYOF_DTC_NOSEVERITY DEM_GET_SEVERITYOF_DTC_PENDING	0x00 0x01 0x02 0x03	Severity successfully returned. DTC value not existing (in UDS format) Severity information is not available The requested value is calculated asynchronously and currently not available. The caller can retry later.
<b>Description:</b>	Used to return the status of Dem_DcmGetSeverityOfDTC.		

Table 8.30: Dem\_ReturnGetSeverityOfDTCType

]()

### 8.2.2.3 Dem\_ReturnGetFunctionalUnitOfDTCType

[SWS\_Dem\_00954] ↗

<b>Name:</b>	Dem_ReturnGetFunctionalUnitOfDTCType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_GET_FUNCTIONAL UNITOFTC_OK DEM_GET_FUNCTIONAL UNITOFTC_WRONG_DT C	0x00 0x01	Functional unit successfully returned. DTC value not existing (in UDS format)
<b>Description:</b>	Used to return the status of Dem_DcmGetFunctionalUnitOfDTC.		

**Table 8.31: Dem\_ReturnGetFunctionalUnitOfDTCType**

↳

### 8.2.2.4 Dem\_ReturnSetFilterType

[SWS\_Dem\_00955] ↗

<b>Name:</b>	Dem_ReturnSetFilterType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_FILTER_ACCEPTED DEM_WRONG_FILTER	0x00 0x01	Filter was accepted Wrong filter selected
<b>Description:</b>	Used to return the status of (re-)setting a specific filter.		

**Table 8.32: Dem\_ReturnSetFilterType**

↳

### 8.2.2.5 Dem\_ReturnGetNumberOfFilteredDTCType

[SWS\_Dem\_00956] ↗

<b>Name:</b>	Dem_ReturnGetNumberOfFilteredDTCType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_NUMBER_OK DEM_NUMBER_FAILED DEM_NUMBER_PENDING	0x00 0x01 0x02	Getting number of filtered DTCs was successful. Getting number of filtered DTCs failed. The requested value is calculated asynchronously and currently not available. The caller can retry later.
<b>Description:</b>	Used to return the status of Dem_DcmGetNumberOfFilteredDTC.		

**Table 8.33: Dem\_ReturnGetNumberOfFilteredDTCType**

]()

### 8.2.2.6 Dem\_ReturnGetNextFilteredElementType

[SWS\_Dem\_00957] [

<b>Name:</b>	Dem_ReturnGetNextFilteredElementType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_FILTERED_OK	0x00	Returned next filtered element
	DEM_FILTERED_NO_MATCHING_ELEMENT	0x01	No further element (matching the filter criteria) found
	DEM_FILTERED_PENDING	0x02	The requested value is calculated asynchronously and currently not available. The caller can retry later. Only used by asynchronous interfaces.
	DEM_FILTERED_BUFFER_TOO_SMALL	0x03	Buffer in the BufSize parameter is not huge enough
<b>Description:</b>	Used to return the status of Dem_DcmGetNextFilteredXxx and Dem_J1939DcmGetNextFilteredXxx.		

**Table 8.34: Dem\_ReturnGetNextFilteredElementType**

]()

### 8.2.2.7 Dem\_ReturnGetDTCByOccurrenceTimeType

[SWS\_Dem\_00958] [

<b>Name:</b>	Dem_ReturnGetDTCByOccurrenceTimeType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_OCCURR_OK	0x00	matching DTC available no DTC is matching the requested occurrence time
	DEM_OCCURR_NOT_AVAILABLE	0x01	
<b>Description:</b>	Used to return the status of Dem_DtcGetDTCByOccurrenceTime.		

**Table 8.35: Dem\_ReturnGetDTCByOccurrenceTimeType**

]()

### 8.2.2.8 Dem\_ReturnDisableDTCRecordUpdateType

[SWS\_Dem\_00959] [

<b>Name:</b>	Dem_ReturnDisableDTCRecordUpdateType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_DISABLE_DTCREC UP_OK	0x00	Event memory update of DTC successfully disabled
	DEM_DISABLE_DTCREC UP_WRONG_DTC	0x01	DTC value not existing (in UDS format)
	DEM_DISABLE_DTCREC UP_WRONG_DTCORIGIN	0x02	Wrong DTC origin
	DEM_DISABLE_DTCREC UP_PENDING	0x03	Disabling is currently not possible. The caller can retry later.
	<b>Description:</b> Used to return the status of Dem_DcmDisableDTCRecordUpdate		

Table 8.36: Dem\_ReturnDisableDTCRecordUpdateType

]()

### 8.2.2.9 Dem\_ReturnGetFreezeFrameDataByDTCType

[SWS\_Dem\_00960] [

<b>Name:</b>	Dem_ReturnGetFreezeFrameDataByDTCType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_GET_FFDATABYDT C_OK	0x00	Size successfully returned.
	DEM_GET_FFDATABYDT C_WRONG_DTC	0x01	DTC value not existing (in UDS format)
	DEM_GET_FFDATABYDT C_WRONG_DTCORIGIN	0x02	Wrong DTC origin
	DEM_GET_FFDATABYDT C_WRONG_RECORDNUMBR	0x03	Record number is not supported by configuration and therefore invalid provided buffer size to small
	DEM_GET_FFDATABYDT C_WRONG_BUFFERSIZE	0x04	The requested value is calculated asynchronously and currently not available. The caller can retry later.
	DEM_GET_FFDATABYDT C_PENDING	0x05	
	<b>Description:</b> Used to return the status of Dem_DcmGetFreezeFrameDataByDTC.		

Table 8.37: Dem\_ReturnGetFreezeFrameDataByDTCType

]()

### 8.2.2.10 Dem\_ReturnGetExtendedDataRecordByDTCType

[SWS\_Dem\_00961] [

<b>Name:</b>	Dem_ReturnGetExtendedDataRecordByDTCType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_RECORD_OK	0x00	Extended data record successfully returned
	DEM_RECORD_WRONG_DTC	0x01	DTC value not existing (in UDS format)
	DEM_RECORD_WRONG_DTCORIGIN	0x02	Origin wrong
	DEM_RECORD_WRONG_NUMBER	0x03	Record number is not supported by configuration and therefore invalid
	DEM_RECORD_WRONG_BUFFERSIZE	0x04	Provided buffer too small
	DEM_RECORD_PENDING	0x05	The requested value is calculated asynchronously and currently not available. The caller can retry later.
<b>Description:</b>	Used to return the status of Dem_DcmGetExtendedDataRecordByDTC.		

Table 8.38: Dem\_ReturnGetExtendedDataRecordByDTCType

]()

### 8.2.2.11 Dem\_ReturnGetSizeOfDataByDTCType

[SWS\_Dem\_00962] [

<b>Name:</b>	Dem_ReturnGetSizeOfDataByDTCType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_GETSIZEBYDTC_OK	0x00	Size successfully returned
	DEM_GETSIZEBYDTC_WRONG_DTC	0x01	DTC value not existing (in UDS format)
	DEM_GETSIZEBYDTC_WRONG_DTCORIGIN	0x02	Wrong DTC origin
	DEM_GETSIZEBYDTC_WRONG_RECNUM	0x03	Record number is not supported by configuration and therefore invalid

	DEM_GETSIZEBYDTC_P ENDING	0x04	The requested value is calculated asynchronously and currently not available. The caller can retry later.
<b>Description:</b>	Used to return the status of Dem_DcmGetSizeOfFreezeFrameByDTC and Dem_DcmGetSizeOfExtendedDataRecordByDTC.		

**Table 8.39: Dem\_ReturnGetSizeOfDataByDTCType**

↴()

### 8.2.2.12 Dem\_ReturnClearDTCType

[SWS\_Dem\_00963] ↴

<b>Name:</b>	Dem_ReturnClearDTCType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_CLEAR_OK DEM_CLEAR_WRONG_DTC DEM_CLEAR_WRONG_DTCORIGIN DEM_CLEAR_FAILED DEM_CLEAR_PENDING  DEM_CLEAR_BUSY  DEM_CLEAR_MEMORY_ERROR	0x00 0x01 0x02 0x03 0x04  0x05  0x06	DTC successfully cleared DTC value not existing (in this format) Wrong DTC origin  DTC clearing failed The DTC clearing is performed asynchronously and still pending. The caller can retry later. DTC not cleared, as another clearing process is in progress. The caller can retry later. An error occurred during erasing a memory location (e.g. if Dem_ClearDTCBehavior in Dem is set to DEM_CLR-RESP_NONVOLATILE_FINISH and erasing of non-volatile-block failed).
<b>Description:</b>	Used to return the status of Dem_<...>ClearDTC.		

**Table 8.40: Dem\_ReturnClearDTCType**

↴()

### 8.2.2.13 Dem\_ReturnControlDTCSettingType

[SWS\_Dem\_00964] ↴

<b>Name:</b>	Dem_ReturnControlDTCSettingType		
<b>Type:</b>	uint8		
<b>Range:</b>	DEM_CONTROL_DTC_SETTING_OK DEM_CONTROL_DTC_SETTING_N_OK DEM_CONTROL_DTC_SETTING_WRONG_DTCGROUP	0x00 0x01 0x02	DTC setting control successful DTC setting control not successful DTC setting control not successful because group of DTC was wrong
<b>Description:</b>	Used to return the status of Dem_DcmDisableDTCSetting and Dem_DcmEnableDTCSetting.		

**Table 8.41: Dem\_ReturnControlDTCSettingType**

]()

## 8.3 Function definitions

This is a list of functions provided for upper layer modules.

### 8.3.1 Dem\_GetVersionInfo

[SWS\_Dem\_00177] [

<b>Service name:</b>	Dem_GetVersionInfo	
<b>Syntax:</b>	<pre>void Dem_GetVersionInfo(     Std_VersionInfoType* versioninfo )</pre>	
<b>Service ID[hex]:</b>	0x00	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	versioninfo	Pointer to where to store the version information of this module.
<b>Return value:</b>	None	
<b>Description:</b>	<p>Returns the version information of this module.</p> <p>API Availability: This API will be available only if  <code>{ecuc(Dem/DemGeneral.DemVersionInfoApi)} == true</code></p>	

**Table 8.42: Dem\_GetVersionInfo**

] (SRS\_BSW\_00402, SRS\_BSW\_00407)

### 8.3.2 Interface ECU State Manager <=> Dem

#### 8.3.2.1 Dem\_PreInit

[SWS\_Dem\_00179] [

<b>Service name:</b>	Dem_PreInit
<b>Syntax:</b>	void Dem_PreInit( void )
<b>Service ID[hex]:</b>	0x01
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	Non Reentrant
<b>Parameters (in):</b>	None
<b>Parameters (inout):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	Initializes the internal states necessary to process events reported by BSW-modules.

Table 8.43: Dem\_PreInit

]()

#### 8.3.2.2 Dem\_Init

[SWS\_Dem\_00181] [

<b>Service name:</b>	Dem_Init	
<b>Syntax:</b>	void Dem_Init( const Dem_ConfigType* ConfigPtr )	
<b>Service ID[hex]:</b>	0x02	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	ConfigPtr	Pointer to the configuration set in VARIANT-POST-BUILD.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	Initializes or reinitializes this module.	

Table 8.44: Dem\_Init

] ([SRS\\_BSW\\_00101](#))

### 8.3.2.3 Dem\_Shutdown

[SWS\_Dem\_00182] [

<b>Service name:</b>	Dem_Shutdown
<b>Syntax:</b>	void Dem_Shutdown( void )
<b>Service ID[hex]:</b>	0x03
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	Non Reentrant
<b>Parameters (in):</b>	None
<b>Parameters (inout):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	Shuts down this module.

Table 8.45: Dem\_Shutdown

] (SRS\_BSW\_00336)

### 8.3.3 Interface BSW modules / SW-Components <=> Dem

#### 8.3.3.1 Dem\_ReportErrorStatus

[SWS\_Dem\_00206] [

<b>Service name:</b>	Dem_ReportErrorStatus	
<b>Syntax:</b>	void Dem_ReportErrorStatus( Dem_EventIdType EventId, Dem_EventStatusType EventStatus )	
<b>Service ID[hex]:</b>	0x0f	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Reentrant for different EventIds. Non reentrant for the same EventId.	
<b>Parameters (in):</b>	EventId EventStatus	Identification of an event by assigned Event ID. Monitor test result
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	Queues the reported events from the BSW modules (API is only used by BSW modules). The interface has an asynchronous behavior, because the processing of the event is done within the Dem main function. OBD Events Suppression shall be ignored for this computation.	

Table 8.46: Dem\_ReportErrorStatus

]()

### 8.3.3.2 Dem\_SetEventAvailable

[SWS\_Dem\_01080] [

<b>Service name:</b>	Dem_SetEventAvailable	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetEventAvailable(     Dem_EventIdType EventId,     boolean AvailableStatus )</pre>	
<b>Service ID[hex]:</b>	0x37	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	EventId AvailableStatus	Identification of an event by assigned EventId. This parameter specifies whether the respective Event shall be available (TRUE) or not (FALSE).
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful. E_NOT_OK: change of available status not accepted
<b>Description:</b>	Set the available status of a specific Event.	

Table 8.47: Dem\_SetEventAvailable

] (SRS\_Diag\_04126)

### 8.3.3.3 Dem\_SetEventStatus

[SWS\_Dem\_00183] [

<b>Service name:</b>	Dem_SetEventStatus	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetEventStatus(     Dem_EventIdType EventId,     Dem_EventStatusType EventStatus )</pre>	
<b>Service ID[hex]:</b>	0x04	
<b>Sync/Async:</b>	Synchronous/Asynchronous	
<b>Reentrancy:</b>	Reentrant for different EventIds. Non reentrant for the same EventId.	
<b>Parameters (in):</b>	EventId EventStatus	Identification of an event by assigned EventId. Monitor test result
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: set of event status was successful E_NOT_OK: set of event status failed or could not be accepted (e.g.: the operation cycle configured for this event has not been started, an according enable condition has been disabled)

<b>Description:</b>	Processes the events reported by SW-Cs via RTE. This API can only be used through the RTE, and therefore no declaration is exported via Dem.h. Some bits of the UDS DTC status byte changes synchronously or asynchronously (refer to SWS_Dem_00036 and SWS_Dem_00379). OBD Events Suppression shall be ignored for this computation.
---------------------	---

**Table 8.48: Dem\_SetEventStatus**

]()

### 8.3.3.4 Dem\_ResetEventDebounceStatus

[SWS\_Dem\_00683] [

<b>Service name:</b>	Dem_ResetEventDebounceStatus	
<b>Syntax:</b>	Std_ReturnType Dem_ResetEventDebounceStatus( Dem_EventIdType EventId, Dem_DebounceResetStatusType DebounceResetStatus )	
<b>Service ID[hex]:</b>	0x09	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different EventIds. Non reentrant for the same EventId.	
<b>Parameters (in):</b>	EventId DebounceReset Status	Identification of an event by assigned EventId. Freeze or reset the internal debounce counter/timer of the specified event.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Only on development error
<b>Description:</b>	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_PrelInit has been completed (refer to SWS_Dem_00438 and SWS_Dem_00167).	

**Table 8.49: Dem\_ResetEventDebounceStatus**

]()

### 8.3.3.5 Dem\_ResetEventStatus

[SWS\_Dem\_00185] [

<b>Service name:</b>	Dem_ResetEventStatus	
<b>Syntax:</b>	Std_ReturnType Dem_ResetEventStatus( Dem_EventIdType EventId )	

<b>Service ID[hex]:</b>	0x05	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different EventIds. Non reentrant for the same EventId.	
<b>Parameters (in):</b>	EventId	Identification of an event by assigned EventId.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: reset of event status was successful E_NOT_OK: reset of event status failed or is not allowed, because the event is already tested in this operation cycle
<b>Description:</b>	Resets the event failed status. This API can only be used through the RTE and therefore no declaration is exported via Dem.h.	

**Table 8.50: Dem\_ResetEventStatus**

]()

### 8.3.3.6 Dem\_PrestoreFreezeFrame

[SWS\_Dem\_00188] [

<b>Service name:</b>	Dem_PrestoreFreezeFrame	
<b>Syntax:</b>	Std_ReturnType Dem_PrestoreFreezeFrame( Dem_EventIdType EventId )	
<b>Service ID[hex]:</b>	0x06	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different EventIds. Non reentrant for the same EventId.	
<b>Parameters (in):</b>	EventId	Identification of an event by assigned EventId.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK Freeze frame prestorage was successful E_NOT_OK Freeze frame prestorage failed
<b>Description:</b>	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/DemEventParameter.DemFFPrestorage-Supported)} == true)	

**Table 8.51: Dem\_PrestoreFreezeFrame**

] (SRS\_Diag\_04074)

### 8.3.3.7 Dem\_ClearPrestoredFreezeFrame

[SWS\_Dem\_00193] [

<b>Service name:</b>	Dem_ClearPrestoredFreezeFrame	
<b>Syntax:</b>	Std_ReturnType Dem_ClearPrestoredFreezeFrame ( Dem_EventIdType EventId )	
<b>Service ID[hex]:</b>	0x07	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different EventIds. Non reentrant for the same EventId.	
<b>Parameters (in):</b>	EventId	Identification of an event by assigned EventId.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: Clear prestored freeze frame was successful E_NOT_OK: Clear prestored freeze frame failed
<b>Description:</b>	<p>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.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/DemConfigSet/DemEventParameter.DemFFPrestorage-Supported)} == true)</p>	

**Table 8.52: Dem\_ClearPrestoredFreezeFrame**

] (SRS\_Diag\_04074)

### 8.3.3.8 Dem\_SetOperationCycleState

[SWS\_Dem\_00194] [

<b>Service name:</b>	Dem_SetOperationCycleState	
<b>Syntax:</b>	Std_ReturnType Dem_SetOperationCycleState ( uint8 OperationCycleId, Dem_OperationCycleStateType CycleState )	
<b>Service ID[hex]:</b>	0x08	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	OperationCycleId	Identification of operation cycle, like power cycle, driving cycle.
	CycleState	New operation cycle state: (re-)start or end
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: set of operation cycle was accepted and will be handled asynchronously E_NOT_OK: set of operation cycle was rejected

<b>Description:</b>	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.
---------------------	--

**Table 8.53: Dem\_SetOperationCycleState**

] ([SRS\\_Diag\\_04076](#))

### 8.3.3.9 Dem\_GetOperationCycleState

[[SWS\\_Dem\\_00729](#)] [

<b>Service name:</b>	Dem_GetOperationCycleState	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_GetOperationCycleState(     uint8 OperationCycleId,     Dem_OperationCycleStateType* CycleState )</pre>	
<b>Service ID[hex]:</b>	0x9e	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	OperationCycleId	Identification of operation cycle, like power cycle, driving cycle.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	CycleState	Cycle status information
<b>Return value:</b>	Std_ReturnType	E_OK: read out of operation cycle was successful E_NOT_OK: read out of operation cycle failed
<b>Description:</b>	Gets information about the status of a specific operation cycle. This API can only be used through the RTE and therefore no declaration is exported via Dem.h.	

**Table 8.54: Dem\_GetOperationCycleState**

]()

### 8.3.3.10 Dem\_SetAgingCycleState

[[SWS\\_Dem\\_00554](#)] [

<b>Service name:</b>	Dem_SetAgingCycleState
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetAgingCycleState(     uint8 OperationCycleId )</pre>
<b>Service ID[hex]:</b>	0x11

<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	OperationCycleId	Identification of aging cycle.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: set of aging cycle was successful E_NOT_OK: set of aging cycle failed
<b>Description:</b>	Triggers the next aging cycle state. This API can only be used through the RTE, and therefore no declaration is exported via Dem.h.	

**Table 8.55: Dem\_SetAgingCycleState**

]()

### 8.3.3.11 Dem\_SetWIRStatus

[SWS\_Dem\_00839] [

<b>Service name:</b>	Dem_SetWIRStatus	
<b>Syntax:</b>	Std_ReturnType Dem_SetWIRStatus( Dem_EventIdType EventId, boolean WIRStatus )	
<b>Service ID[hex]:</b>	0x7a	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different EventIds. Non reentrant for the same EventId.	
<b>Parameters (in):</b>	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"
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: The request is accepted E_NOT_OK: not be accepted (e.g. disabled controlDTCSetting) and should be repeated.
<b>Description:</b>	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.	

**Table 8.56: Dem\_SetWIRStatus**

]()

### 8.3.3.12 Dem\_GetComponentFailed

[SWS\_Dem\_01115] [

<b>Service name:</b>	Dem_GetComponentFailed	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_GetComponentFailed(     Dem_ComponentIdType ComponentId,     boolean* ComponentFailed )</pre>	
<b>Service ID[hex]:</b>	0x2a	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	ComponentId	Identification of a DemComponent
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	ComponentFailed	TRUE: failed FALSE: not failed
<b>Return value:</b>	Std_ReturnType	E_OK: getting "ComponentFailed" was successful E_NOT_OK: getting "ComponentFailed" was not successful
<b>Description:</b>	Gets the failed status of a DemComponent.	

Table 8.57: Dem\_GetComponentFailed

] (SRS\_Diag\_04142)

### 8.3.3.13 Dem\_GetEventStatus

[SWS\_Dem\_00195] [

<b>Service name:</b>	Dem_GetEventStatus	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_GetEventStatus (     Dem_EventIdType EventId,     Dem_UdsStatusByteType* EventStatusByte )</pre>	
<b>Service ID[hex]:</b>	0x0a	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	EventId	Identification of an event by assigned EventId.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	EventStatusByte	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.
<b>Return value:</b>	Std_ReturnType	E_OK: get of event status was successful E_NOT_OK: get of event status failed
<b>Description:</b>	Gets the current extended event status of an event.	

Table 8.58: Dem\_GetEventStatus

」(SRS\_Diag\_04096)

### 8.3.3.14 Dem\_GetEventFailed

[SWS\_Dem\_00196] 「

<b>Service name:</b>	Dem_GetEventFailed	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_GetEventFailed(     Dem_EventIdType EventId,     boolean* EventFailed )</pre>	
<b>Service ID[hex]:</b>	0x0b	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	EventId	Identification of an event by assigned EventId.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	EventFailed	TRUE - Last Failed FALSE - not Last Failed
<b>Return value:</b>	Std_ReturnType	E_OK: get of "EventFailed" was successful E_NOT_OK: get of "EventFailed" was not successful
<b>Description:</b>	Gets the event failed status of an event.	

Table 8.59: Dem\_GetEventFailed

」()

### 8.3.3.15 Dem\_GetEventTested

[SWS\_Dem\_00197] 「

<b>Service name:</b>	Dem_GetEventTested	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_GetEventTested(     Dem_EventIdType EventId,     boolean* EventTested )</pre>	
<b>Service ID[hex]:</b>	0x0c	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	EventId	Identification of an event by assigned EventId.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	EventTested	TRUE - event tested this cycle FALSE - event not tested this cycle
<b>Return value:</b>	Std_ReturnType	E_OK: get of event state "tested" successful E_NOT_OK: get of event state "tested" failed
<b>Description:</b>	Gets the event tested status of an event.	

Table 8.60: Dem\_GetEventTested

]()

### 8.3.3.16 Dem\_GetDebouncingOfEvent

[SWS\_Dem\_00730] [

<b>Service name:</b>	Dem_GetDebouncingOfEvent	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_GetDebouncingOfEvent (     Dem_EventIdType EventId,     Dem_DeBouncingStateType* DebouncingState )</pre>	
<b>Service ID[hex]:</b>	0x9f	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	EventId	Identification of an event by assigned EventId.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	DebouncingState	<p>Bit 0 Temporarily Defective (corresponds to <math>0 &lt; FDC &lt; 127</math>)</p> <p>Bit 1 finally Defective (corresponds to <math>FDC = 127</math>)</p> <p>Bit 2 temporarily healed (corresponds to <math>-128 &lt; FDC &lt; 0</math>)</p> <p>Bit 3 Test complete (corresponds to <math>FDC = -128</math> or <math>FDC = 127</math>)</p> <p>Bit 4 DTR Update (= Test complete &amp;&amp; Debouncing complete &amp;&amp; enable conditions / storage conditions fulfilled)</p>
<b>Return value:</b>	Std_ReturnType	<p>E_OK: get of debouncing status per event state successful</p> <p>E_NOT_OK: get of debouncing per event state failed</p>
<b>Description:</b>	<p>Gets the debouncing status of an event.</p> <p>This function shall not be used for EventId with native debouncing within their functions. It is rather for EventIds using debouncing within the Dem.</p>	

Table 8.61: Dem\_GetDebouncingOfEvent

]()

### 8.3.3.17 Dem\_GetDTCOfEvent

[SWS\_Dem\_00198] [

<b>Service name:</b>	Dem_GetDTCOfEvent
<b>Syntax:</b>	<pre>Std_ReturnType Dem_GetDTCOfEvent (     Dem_EventIdType EventId,     Dem_DTCFormatType DTCFormat,     uint32* DTCOfEvent )</pre>
<b>Service ID[hex]:</b>	0x0d

<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	EventId DTCFormat	Identification of an event by assigned EventId. Defines the output-format of the requested DTC value.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	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.
<b>Return value:</b>	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
<b>Description:</b>	Gets the DTC of an event.	

**Table 8.62: Dem\_GetDTCOfEvent**

]()

### 8.3.3.18 Dem\_SetEnableCondition

[SWS\_Dem\_00201] [

<b>Service name:</b>	Dem_SetEnableCondition	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetEnableCondition(     uint8 EnableConditionID,     boolean ConditionFulfilled )</pre>	
<b>Service ID[hex]:</b>	0x39	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	EnableConditionID ConditionFulfilled	This parameter identifies the enable condition. This parameter specifies whether the enable condition assigned to the EnableConditionID is fulfilled (TRUE) or not fulfilled (FALSE).
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	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.
<b>Description:</b>	Sets an enable condition.  API Availability: This API will be available only if <code>{ecuc(Dem/DemGeneral/DemEnableCondition)} != NULL</code>	

**Table 8.63: Dem\_SetEnableCondition**

]()

### 8.3.3.19 Dem\_SetStorageCondition

[SWS\_Dem\_00556] [

<b>Service name:</b>	Dem_SetStorageCondition	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetStorageCondition(     uint8 StorageConditionID,     boolean ConditionFulfilled )</pre>	
<b>Service ID[hex]:</b>	0x38	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	StorageConditionID ConditionFulfilled	This parameter identifies the storage condition. This parameter specifies whether the storage condition assigned to the StorageConditionID is fulfilled (TRUE) or not fulfilled (FALSE).
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	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.
<b>Description:</b>	<p>Sets a storage condition.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral/DemStorageCondition)} != NULL)</p>	

**Table 8.64: Dem\_SetStorageCondition**

]()

### 8.3.3.20 Dem\_GetFaultDetectionCounter

[SWS\_Dem\_00203] [

<b>Service name:</b>	Dem_GetFaultDetectionCounter	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_GetFaultDetectionCounter(     Dem_EventIdType EventId,     sint8* FaultDetectionCounter )</pre>	
<b>Service ID[hex]:</b>	0x3e	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	EventId	Identification of an event by assigned EventId.
<b>Parameters (inout):</b>	None	

<b>Parameters (out):</b>	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
<b>Return value:</b>	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
<b>Description:</b>	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.	

**Table 8.65: Dem\_GetFaultDetectionCounter**

↴()

### 8.3.3.21 Dem\_GetIndicatorStatus

[SWS\_Dem\_00205] ↴

<b>Service name:</b>	Dem_GetIndicatorStatus	
<b>Syntax:</b>	Std_ReturnType Dem_GetIndicatorStatus( uint8 IndicatorId, Dem_IndicatorStatusType* IndicatorStatus )	
<b>Service ID[hex]:</b>	0x29	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	IndicatorId	Number of indicator
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	IndicatorStatus	Status of the indicator, like off, on, or blinking.
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation failed
<b>Description:</b>	Gets the indicator status derived from the event status.  API Availability: This API will be available only if ({ecuc(Dem/DemGeneral/DemIndicator)} != NULL)	

**Table 8.66: Dem\_GetIndicatorStatus**

↴()

### 8.3.3.22 Dem\_GetEventFreezeFrameData

[SWS\_Dem\_00558] ↴

<b>Service name:</b>	Dem_GetEventFreezeFrameData (obsolete)	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_GetEventFreezeFrameData(     Dem_EventIdType EventId,     uint8 RecordNumber,     boolean ReportTotalRecord,     uint16 DataId,     uint8* DestBuffer )</pre>	
<b>Service ID[hex]:</b>	0x31	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	EventId RecordNumber  ReportTotalRecord  DataId	Identification of an event by assigned EventId. This parameter is a unique identifier for a freeze frame record as defined in ISO14229-1. <b>0xFF means most recent freeze frame record is returned.</b> 0x00 is only supported if the Dem module supports WWH-OBD (refer to DemOBDSupport) This parameter is obsolete and shall be set to FALSE. This function requests a single PID/DID. This parameter specifies the DID (ISO14229-1) that shall be copied to the destination buffer.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation could not be performed DEM_E_NODATAAVAILABLE: The requested event data is not currently stored (but the request was valid) DEM_E_WRONG_RECORDNUMBER: The requested record number is not supported by the event DEM_E_WRONG_DIDNUMBER: The requested DID is not supported by the freeze frame
<b>Description:</b>	Gets the data of a freeze frame by event.  <b>Tags:</b> atp.Status=obsolete atp.StatusRevisionBegin=4.2.2	

Table 8.67: Dem\_GetEventFreezeFrameData

]()

### 8.3.3.23 Dem\_GetEventFreezeFrameDataEx

[SWS\_Dem\_01191] [

<b>Service name:</b>	Dem_GetEventFreezeFrameDataEx
----------------------	-------------------------------

<b>Syntax:</b>	Std_ReturnType Dem_GetEventFreezeFrameDataEx( Dem_EventIdType EventId, uint8 RecordNumber, uint16 DataId, uint8* DestBuffer, uint16* BufSize )	
<b>Service ID[hex]:</b>	0x6e	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	EventId RecordNumber  DataId	Identification of an event by assigned EventId. This parameter is a unique identifier for a freeze frame record as defined in ISO14229-1. <b>0xFF means most recent freeze frame record is returned.</b> 0x00 is only supported if the Dem module supports WWH-OBD (refer to DemOBDSupport) This parameter specifies the DID (ISO14229-1) that shall be copied to the destination buffer.
<b>Parameters (inout):</b>	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.
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation could not be performed DEM_E_NODATAAVAILABLE: The requested event data is not currently stored (but the request was valid) DEM_E_WRONG_RECORDNUMBER: The requested record number is not supported by the event DEM_E_WRONG_DIDNUMBER: The requested DID is not supported by the freeze frame DEM_E_WRONG_BUFFERSIZE: The provided buffer size is too small
<b>Description:</b>	Gets the data of a freeze frame by event.	

**Table 8.68: Dem\_GetEventFreezeFrameDataEx**

↴()

### 8.3.3.24 Dem\_GetEventExtendedDataRecord

**[SWS\_Dem\_00557]** ↴

<b>Service name:</b>	Dem_GetEventExtendedDataRecord (obsolete)
----------------------	---

<b>Syntax:</b>	Std_ReturnType Dem_GetEventExtendedDataRecord( Dem_EventIdType EventId, uint8 RecordNumber, uint8* DestBuffer )	
<b>Service ID[hex]:</b>	0x30	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	EventId RecordNumber	Identification of an event by assigned EventId. Identification of requested Extended data record. Valid values are between 0x01 and 0xEF as defined in ISO14229-1.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation could not be performed DEM_E_NODATAAVAILABLE: The requested event data is not currently stored (but the request was valid) DEM_E_WRONG_RECORDNUMBER: The requested record number is not supported by the event
<b>Description:</b>	Gets the data of an extended data record by event.  <b>Tags:</b> atp.Status=obsolete atp.StatusRevisionBegin=4.2.2	

**Table 8.69: Dem\_GetEventExtendedDataRecord**

↴()

### 8.3.3.25 Dem\_GetEventExtendedDataRecordEx

[SWS\_Dem\_01190] ↵

<b>Service name:</b>	Dem_GetEventExtendedDataRecordEx	
<b>Syntax:</b>	Std_ReturnType Dem_GetEventExtendedDataRecordEx( Dem_EventIdType EventId, uint8 RecordNumber, uint8* DestBuffer, uint16* BufSize )	
<b>Service ID[hex]:</b>	0x6d	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	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.
<b>Parameters (inout):</b>	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.
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation could not be performed DEM_E_NODATAAVAILABLE: The requested event data is not currently stored (but the request was valid) DEM_E_WRONG_RECORDNUMBER: The requested record number is not supported by the event DEM_E_WRONG_BUFFERSIZE: The provided buffer size is too small
<b>Description:</b>	Gets the data of an extended data record by event.	

**Table 8.70: Dem\_GetEventExtendedDataRecordEx**

↴()

### 8.3.3.26 Dem\_GetEventMemoryOverflow

**[SWS\_Dem\_00559]**

<b>Service name:</b>	Dem_GetEventMemoryOverflow	
<b>Syntax:</b>	Std_ReturnType Dem_GetEventMemoryOverflow( Dem_DTCOriginType DTCOrigin, boolean* OverflowIndication )	
<b>Service ID[hex]:</b>	0x32	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	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.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	OverflowIndication	This parameter returns TRUE if the according event memory was overflowed, otherwise it returns FALSE.
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation failed or is not supported
<b>Description:</b>	Gets the event memory overflow indication status.	

**Table 8.71: Dem\_GetEventMemoryOverflow**

」(SRS\_Diag\_04093)

### 8.3.3.27 Dem\_GetNumberOfEventMemoryEntries

[SWS\_Dem\_00652] 「

<b>Service name:</b>	Dem_GetNumberOfEventMemoryEntries	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_GetNumberOfEventMemoryEntries (     Dem_DTCOriginType DTCOrigin,     uint8* NumberOfEventMemoryEntries )</pre>	
<b>Service ID[hex]:</b>	0x35	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	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.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	NumberOfEventMemoryEntries	Number of entries currently stored in the requested event memory.
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation failed
<b>Description:</b>	Returns the number of entries currently stored in the requested event memory.	

Table 8.72: Dem\_GetNumberOfEventMemoryEntries

」(SRS\_Diag\_04109)

### 8.3.3.28 Dem\_SetComponentAvailable

[SWS\_Dem\_01117] 「

<b>Service name:</b>	Dem_SetComponentAvailable	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetComponentAvailable (     Dem_ComponentIdType ComponentId,     boolean AvailableStatus )</pre>	
<b>Service ID[hex]:</b>	0x2b	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	ComponentId AvailableStatus	Identification of a DemComponent. This parameter specifies whether the respective Component shall be available (TRUE) or not (FALSE).
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful

<b>Description:</b>	Set the availability of a specific DemComponent.
---------------------	--

**Table 8.73: Dem\_SetComponentAvailable**

](*SRS\_Diag\_04142*)

### 8.3.3.29 Dem\_SetDTCSuppression

[SWS\_Dem\_01047] [

<b>Service name:</b>	Dem_SetDTCSuppression	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetDTCSuppression(     uint32 DTC,     Dem_DTCFormatType DTCFormat,     boolean SuppressionStatus )</pre>	
<b>Service ID[hex]:</b>	0x33	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTC DTCFormat SuppressionStatus	Diagnostic Trouble Code Defines the input-format of the provided DTC value. This parameter specifies whether the respective DTC shall be disabled (TRUE) or enabled (FALSE).
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK (Operation was successful), E_NOT_OK (operation failed or event entry for this DTC still exists)
<b>Description:</b>	<p>Set the suppression status of a specific DTC.</p> <p>API Availability: This API will be available only if          ({ecuc(Dem/DemGeneral.DemSuppressionSupport)} == DEM_DTC_SUPPRESSION)</p>	

**Table 8.74: Dem\_SetDTCSuppression**

]()

### 8.3.3.30 Dem\_ClearDTC

[SWS\_Dem\_00665] [

<b>Service name:</b>	Dem_ClearDTC	
<b>Syntax:</b>	<pre>Dem_ReturnClearDTCType Dem_ClearDTC(     uint32 DTC,     Dem_DTCFormatType DTCFormat,     Dem_DTCOriginType DTCOrigin )</pre>	

<b>Service ID[hex]:</b>	0x23	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	DTC  DTCFormat DTCOrigin	Defines the DTC in respective format, that shall be cleared from the event memory. If the DTC fits to a DTC group number, all DTCs of the group shall be cleared.  Defines the input-format of the provided DTC value. If the Dem supports more than one event memory this parameter is used to select the source memory the DTCs shall be read from.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Dem_ReturnClearDTCType	Status of the operation of type Dem_ReturnClearDTCType.
<b>Description:</b>	Clears single DTCs, as well as groups of DTCs. This API is intended for complex device driver. It can only be used through the RTE (due to work-around described below SWS_Dem_00659), and therefore no declaration is exported via Dem.h.	

**Table 8.75: Dem\_ClearDTC**

」(SRS\_Diag\_04122)

### 8.3.4 Interface Dcm <=> Dem

#### 8.3.4.1 Access DTCs and Status Information

##### 8.3.4.1.1 Dem\_DcmGetTranslationType

[SWS\_Dem\_00230] 「

<b>Service name:</b>	Dem_DcmGetTranslationType	
<b>Syntax:</b>	Dem_DTCTranslationFormatType Dem_DcmGetTranslationType( void )	
<b>Service ID[hex]:</b>	0x3c	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Dem_DTCTranslationFormatType	Returns the configured DTC translation format. A combination of different DTC formats is not possible.
<b>Description:</b>	Gets the supported DTC formats of the ECU. The supported formats are configured via DemTypeOfDTCSupported.	

**Table 8.76: Dem\_DcmGetTranslationType**

]()

### 8.3.4.1.2 Dem\_DcmGetDTCStatusAvailabilityMask

[SWS\_Dem\_00213] [

<b>Service name:</b>	Dem_DcmGetDTCStatusAvailabilityMask	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_DcmGetDTCStatusAvailabilityMask (     Dem_UdsStatusByteType* DTCStatusMask )</pre>	
<b>Service ID[hex]:</b>	0x16	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	E_OK: get of DTC status mask was successful E_NOT_OK: get of DTC status mask failed
<b>Description:</b>	Gets the DTC Status availability mask.	

Table 8.77: Dem\_DcmGetDTCStatusAvailabilityMask

]()

### 8.3.4.1.3 Dem\_DcmGetStatusOfDTC

[SWS\_Dem\_00212] [

<b>Service name:</b>	Dem_DcmGetStatusOfDTC	
<b>Syntax:</b>	<pre>Dem_ReturnGetStatusOfDTCType Dem_DcmGetStatusOfDTC (     uint32 DTC,     Dem_DTCOriginType DTCOrigin,     uint8* DTCStatus )</pre>	
<b>Service ID[hex]:</b>	0x15	
<b>Sync/Async:</b>	Synchronous or Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTC DTCOrigin	Diagnostic Trouble Code in UDS format. If the Dem supports more than one event memory this parameter is used to select the source memory the DTCs shall be read from.
<b>Parameters (inout):</b>	None	

<b>Parameters (out):</b>	DTCStatus	This parameter receives the status information of the requested DTC. If the return value of the function call is other than DEM_STATUS_OK this parameter does not contain valid data. 0x00...0xFF match DTCStatusMask as defined in ISO14229-1
<b>Return value:</b>	Dem_ReturnGetStatusOfDTCType	Status of the operation of type Dem_ReturnGetStatusOfDTCType.
<b>Description:</b>		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_STATUS_WRONG_DTC.

**Table 8.78: Dem\_DcmGetStatusOfDTC**

](*SRS\_Diag\_04066*)

#### 8.3.4.1.4 Dem\_DcmGetSeverityOfDTC

[SWS\_Dem\_00232] [

<b>Service name:</b>	Dem_DcmGetSeverityOfDTC	
<b>Syntax:</b>	Dem_ReturnGetSeverityOfDTCType Dem_DcmGetSeverityOfDTCType( uint32 DTC, Dem_DTCSeverityType* DTCSeverity )	
<b>Service ID[hex]:</b>	0x0e	
<b>Sync/Async:</b>	Synchronous or Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTC	Diagnostic Trouble Code in UDS format.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	DTCSeverity	This parameter contains the DTCSeverity according to ISO 14229-1.
<b>Return value:</b>	Dem_ReturnGetSeverityOfDTCType	Status of the operation of type Dem_ReturnGetSeverityOfDTCType.
<b>Description:</b>	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).	

**Table 8.79: Dem\_DcmGetSeverityOfDTC**

]()

#### 8.3.4.1.5 Dem\_DcmGetFunctionalUnitOfDTC

[SWS\_Dem\_00594] [

<b>Service name:</b>	Dem_DcmGetFunctionalUnitOfDTC	
<b>Syntax:</b>	<pre>Dem_ReturnGetFunctionalUnitOfDTCType Dem_DcmGetFunctionalUnitOfDTC(     uint32 DTC,     uint8* DTFunctionalUnit )</pre>	
<b>Service ID[hex]:</b>	0x34	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTC	Diagnostic Trouble Code in UDS format.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	DTFunctionalUnit	Functional unit value of this DTC
<b>Return value:</b>	Dem_ReturnGetFunctionalUnitOfDTCType	Status of the operation of type Dem_ReturnGetFunctionalUnitOfDTCType.
<b>Description:</b>	Gets the functional unit of the requested DTC.	

**Table 8.80: Dem\_DcmGetFunctionalUnitOfDTC**

]()

### 8.3.4.1.6 Dem\_DcmSetDTCFilter

[SWS\_Dem\_00208] [

<b>Service name:</b>	Dem_DcmSetDTCFilter	
<b>Syntax:</b>	<pre>Dem_ReturnSetFilterType Dem_DcmSetDTCFilter(     Dem_UdsStatusByteType DTCSecurityMask,     Dem_DTCKindType DTCKind,     Dem_DTCFormatType DTCFormat,     Dem_DTCOriginType DTCOrigin,     boolean FilterWithSeverity,     Dem_DTCSeverityType DTCSeverityMask,     boolean FilterForFaultDetectionCounter )</pre>	
<b>Service ID[hex]:</b>	0x13	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTCSecurityMask	<p>Status-byte mask for DTC status-byte filtering</p> <p>Values:</p> <p>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)</p> <p>0x01..0xFF: Status-byte mask according to ISO 14229-1 DTCSecurityMask (handed over by Dcm from service request directly) to filter for DTCs with at least one status bit set matching this status-byte mask</p>

	DTCKind	Defines the functional group of DTCs to be reported (e.g. all DTC, OBD-relevant DTC). If passed value does not fit to Configuration, the DET error DEM_E_WRONG_CONFIGURATION shall be reported, e.g if DTCKind "DEM_DTC_KIND_EMISIION_REL_DTCs" is passed, but no emission related DTCs are configured.
	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, e.g. if DTCOrigin "DEM_DTC_ORIGIN_MIRROR_MEMORY" is passed, but no mirror memory is configured.
	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.
	DTCSeverityMask	This parameter contains the DTCSeverityMask according to ISO14229-1 (see for example Service 0x19, subfunction 0x08). If the value is invalid the function returns DEM_WRONG_FILTER.
	FilterForFaultDetectionCounter	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 GetFaultDetectionCounter.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Dem_ReturnSet FilterType	Status of the operation to (re-)set a DTC filter.

<b>Description:</b> <p>Sets the DTC Filter.</p> <p>The server shall perform a bit-wise logical AND-ing operation between the parameter DTCStatusMask and the current DTC status in the server. In addition to the DTCStatusAvailabilityMask, the server shall return all DTCs for which the result of the AND-ing operation is non-zero [i.e. <math>(statusOfDTC \&amp; DTCStatusMask) != 0</math>]. 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</p> <p><math>((statusOfDTC \&amp; DTCStatusMask) != 0) \&amp;&amp; ((severity \&amp; DTCSSeverity- Mask) != 0) == TRUE</math></p>
--

**Table 8.81: Dem\_DcmSetDTCFilter**

]()

#### 8.3.4.1.7 Dem\_DcmGetNumberOfFilteredDTC

[SWS\_Dem\_00214] [

<b>Service name:</b> <p>Dem_DcmGetNumberOfFilteredDTC</p>	
<b>Syntax:</b> <p>Dem_ReturnGetNumberOfFilteredDTCType Dem_DcmGetNumber OfFilteredDTC(   uint16* NumberOfFilteredDTC )</p>	
<b>Service ID[hex]:</b> <p>0x17</p>	
<b>Sync/Async:</b> <p>Asynchronous</p>	
<b>Reentrancy:</b> <p>Non Reentrant</p>	
<b>Parameters (in):</b> <p>None</p>	
<b>Parameters (inout):</b> <p>None</p>	
<b>Parameters (out):</b> <p>NumberOfFilteredD TC</p>	<p>The number of DTCs matching the defined status mask.</p>
<b>Return value:</b> <p>Dem_ReturnGet NumberOfFilteredD TCType</p>	<p>Status of the operation to retrieve a number of DTC from the Dem</p>
<b>Description:</b> <p>Gets the number of a filtered DTC.</p>	

**Table 8.82: Dem\_DcmGetNumberOfFilteredDTC**

]()

#### 8.3.4.1.8 Dem\_DcmGetNextFilteredDTC

[SWS\_Dem\_00215] [

<b>Service name:</b>	Dem_DcmGetNextFilteredDTC	
<b>Syntax:</b>	<pre>Dem_ReturnGetNextFilteredElementType Dem_DcmGetNext FilteredDTC(     uint32* DTC,     Dem_UdsStatusByteType* DTCStatus )</pre>	
<b>Service ID[hex]:</b>	0x18	
<b>Sync/Async:</b>	Synchronous or Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	DTC  DTCStatus	Receives the DTC value in respective format of the filter returned by this function. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.  This parameter receives the status information of the requested DTC.  It follows the format as defined in ISO14229-1 If the return value of the function call is other than DEM_FILTERED_OK this parameter does not contain valid data.
<b>Return value:</b>	Dem_ReturnGetNextFilteredElementType	Status of the operation to retrieve a DTC from the Dem. The value DEM_FILTERED_PENDING is not always allowed (refer to SWS_DEM_00653).
<b>Description:</b>	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.	

**Table 8.83: Dem\_DcmGetNextFilteredDTC**

]()

### 8.3.4.1.9 Dem\_DcmGetNextFilteredDTCAndFDC

[SWS\_Dem\_00227] [

<b>Service name:</b>	Dem_DcmGetNextFilteredDTCAndFDC	
<b>Syntax:</b>	<pre>Dem_ReturnGetNextFilteredElementType Dem_DcmGetNext FilteredDTCAndFDC(     uint32* DTC,     sint8* DTCFaultDetectionCounter )</pre>	
<b>Service ID[hex]:</b>	0x3b	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	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 DEM_FILTERED_OK this parameter does not contain valid data.

	DTCFaultDetectionCounter	This parameter receives the Fault Detection Counter information of the requested DTC. If the return value of the function call is other than DEM_FILTERED_OK this parameter does not contain valid data.  -128dec...127dec PASSED...FAILED according to ISO 14229-1
<b>Return value:</b>	Dem_ReturnGetNextFilteredElementType	Status of the operation to retrieve a DTC from the Dem.
<b>Description:</b>	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.	

**Table 8.84: Dem\_DcmGetNextFilteredDTCAndFDC**

]()

### 8.3.4.1.10 Dem\_DcmGetNextFilteredDTCAndSeverity

[SWS\_Dem\_00281] [

<b>Service name:</b>	Dem_DcmGetNextFilteredDTCAndSeverity		
<b>Syntax:</b>	<pre>Dem_ReturnGetNextFilteredElementType Dem_DcmGetNext FilteredDTCAndSeverity(     uint32* DTC,     Dem_UdsStatusByteType* DTCStatus,     Dem_DTCSeverityType* DTCSeverity,     uint8* DTCFunctionalUnit )</pre>		
<b>Service ID[hex]:</b>	0x3d		
<b>Sync/Async:</b>	Asynchronous		
<b>Reentrancy:</b>	Non Reentrant		
<b>Parameters (in):</b>	None		
<b>Parameters (inout):</b>	None		
<b>Parameters (out):</b>	DTC DTCStatus DTCSeverity	Receives the DTC value in respective format of the filter returned by this function. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data. This parameter receives the status information of the requested DTC. It follows the format as defined in ISO14229-1 If the return value of the function call is other than DEM_FILTERED_OK this parameter does not contain valid data. Receives the severity value returned by the function. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.	

	DTCFunctionalUnit	Receives the functional unit value returned by the function. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.
<b>Return value:</b>	Dem_ReturnGetNextFilteredElementType	Status of the operation to retrieve a DTC from the Dem.
<b>Description:</b>	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.	

**Table 8.85: Dem\_DcmGetNextFilteredDTCAndSeverity**

]()

### 8.3.4.1.11 Dem\_DcmSetFreezeFrameRecordFilter

[SWS\_Dem\_00209] [

<b>Service name:</b>	Dem_DcmSetFreezeFrameRecordFilter	
<b>Syntax:</b>	<pre>Dem_ReturnSetFilterType Dem_DcmSetFreezeFrameRecord Filter(     Dem_DTCFormatType DTCFormat,     uint16* NumberOfFilteredRecords )</pre>	
<b>Service ID[hex]:</b>	0x3f	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTCFormat	Defines the output-format of the requested DTC values for the sub-sequent API calls.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	NumberOfFilteredRecords	Number of freeze frame records currently stored in the event memory.
<b>Return value:</b>	Dem_ReturnSetFilterType	Status of the operation to (re-)set a freeze frame record filter.
<b>Description:</b>	Sets a freeze frame record filter.	

**Table 8.86: Dem\_DcmSetFreezeFrameRecordFilter**

]()

### 8.3.4.1.12 Dem\_DcmGetNextFilteredRecord

[SWS\_Dem\_00224] [

<b>Service name:</b>	Dem_DcmGetNextFilteredRecord
----------------------	------------------------------

<b>Syntax:</b>	Dem_ReturnGetNextFilteredElementType Dem_DcmGetNextFilteredRecord( uint32* DTC, uint8* RecordNumber )	
<b>Service ID[hex]:</b>	0x3a	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	DTC  RecordNumber	Receives the DTC value in respective format of the filter returned by this function. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.  Freeze frame record number of the reported DTC (relative addressing). If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.
<b>Return value:</b>	Dem_ReturnGetNextFilteredElementType	Status of the operation to retrieve a DTC and its associated snapshot record number from the Dem.
<b>Description:</b>	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.	

**Table 8.87: Dem\_DcmGetNextFilteredRecord**

]()

### 8.3.4.1.13 Dem\_DcmGetDTCByOccurrenceTime

[SWS\_Dem\_00218] [

<b>Service name:</b>	Dem_DcmGetDTCByOccurrenceTime	
<b>Syntax:</b>	Dem_ReturnGetDTCByOccurrenceTimeType Dem_DcmGetDTCByOccurrenceTime( Dem_DTCRequestType DTCRequest, uint32* DTC )	
<b>Service ID[hex]:</b>	0x19	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTCRequest	This parameter defines the request type of the DTC.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	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.

<b>Return value:</b>	Dem_ReturnGetDTCByOccurrenceTimeType	Status of the operation of type Dem_ReturnGetDTCByOccurrenceTimeType.
<b>Description:</b>	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.	

**Table 8.88: Dem\_DcmGetDTCByOccurrenceTime**

↴()

### 8.3.4.1.14 Dem\_DcmControlDTCStatusChangedNotification

[SWS\_Dem\_00846] ↴

<b>Service name:</b>	Dem_DcmControlDTCStatusChangedNotification	
<b>Syntax:</b>	<pre>void Dem_DcmControlDTCStatusChangedNotification(     boolean TriggerNotification )</pre>	
<b>Service ID[hex]:</b>	0xb0	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	TriggerNotification	This parameter specifies whether the triggering of the notification shall be enabled (TRUE) or disabled (FALSE).
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	Controls the triggering of Dcm_DemTriggerOnDTCStatus.	

**Table 8.89: Dem\_DcmControlDTCStatusCN**

↴()

### 8.3.4.2 Access extended data records and FreezeFrame data

#### 8.3.4.2.1 Dem\_DcmDisableDTCTRecordUpdate

[SWS\_Dem\_00233] ↴

<b>Service name:</b>	Dem_DcmDisableDTCTRecordUpdate	
<b>Syntax:</b>	<pre>Dem_ReturnDisableDTCTRecordUpdateType Dem_DcmDisableDTCTRecordUpdate(     uint32 DTC,     Dem_DTCOriginType DTCOrigin )</pre>	
<b>Service ID[hex]:</b>	0x1a	

<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTC  DTCOrigin	Selects the DTC in UDS format, for which DTC record update shall be disabled.  If the Dem supports more than one event memory, this parameter is used to select the source memory for which DTC record update shall be disabled.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Dem_ReturnDisableDTCRecordUpdateType	Status of the operation to disable the event memory update of a specific DTC.
<b>Description:</b>	Disables the event memory update of a specific DTC (only one at one time).	

**Table 8.90: Dem\_DcmDisableDTCRecordUpdate**

↴()

### 8.3.4.2.2 Dem\_DcmEnableDTCRecordUpdate

**[SWS\_Dem\_00234]** ↴

<b>Service name:</b>	Dem_DcmEnableDTCRecordUpdate	
<b>Syntax:</b>	Std_ReturnType Dem_DcmEnableDTCRecordUpdate( void )	
<b>Service ID[hex]:</b>	0x1b	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned.
<b>Description:</b>	Enables the event memory update of the DTC disabled by Dem_DcmDisableDTCRecordUpdate() before.	

**Table 8.91: Dem\_DcmEnableDTCRecordUpdate**

↴()

### 8.3.4.2.3 Dem\_DcmGetFreezeFrameDataByDTC

**[SWS\_Dem\_00236]** ↴

<b>Service name:</b>	Dem_DcmGetFreezeFrameDataByDTC
----------------------	--------------------------------

<b>Syntax:</b>	Dem_ReturnGetFreezeFrameDataByDTCType Dem_DcmGetFreezeFrameDataByDTC( uint32 DTC, Dem_DTCOriginType DTCOrigin, uint8 RecordNumber, uint8* DestBuffer, uint16* BufSize )	
<b>Service ID[hex]:</b>	0x1d	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTC DTCOrigin  RecordNumber	Diagnostic Trouble Code in UDS format. If the Dem supports more than one event memory, this parameter is used to select the source memory the DTCs shall be read from. This parameter is a unique identifier for a freeze frame record as defined in ISO 15031-5 and ISO 14229-1. This record number is unique per DTC (relative addressing). The value 0xFF is not allowed. The value 0x00 indicates the DTC-specific OBD freeze frame.
<b>Parameters (inout):</b>	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.
<b>Parameters (out):</b>	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]}
<b>Return value:</b>	Dem_ReturnGetFreezeFrameDataByDTCType	Status of the operation to retrieve freeze frame data by DTC.
<b>Description:</b>	Gets freeze frame data by DTC. The function stores the data in the provided DestBuffer.	

**Table 8.92: Dem\_DcmGetFreezeFrameDataByDTC**

](SRS\_Diag\_04066)

### 8.3.4.2.4 Dem\_DcmGetSizeOfFreezeFrameByDTC

[SWS\_Dem\_00238] [

<b>Service name:</b>	Dem_DcmGetSizeOfFreezeFrameByDTC
----------------------	----------------------------------

<b>Syntax:</b>	Dem_ReturnGetSizeOfDataByDTCType Dem_DcmGetSizeOfFreezeFrameByDTC ( uint32 DTC, Dem_DTCOriginType DTCOrigin, uint8 RecordNumber, uint16* SizeOfFreezeFrame )	
<b>Service ID[hex]:</b>	0x1f	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTC DTCOrigin  RecordNumber	Diagnostic Trouble Code in UDS format. If the Dem supports more than one event memory, this parameter is used to select the source memory the DTCs shall be read from.  This parameter is a unique identifier for a freeze frame record as defined in ISO 15031-5 and ISO 14229-1. This record number is unique per DTC (relative addressing). The value 0xFF is explicitly allowed to request the overall size.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	SizeOfFreezeFrame	Number of bytes in the requested freeze frame record.
<b>Return value:</b>	Dem_ReturnGet SizeOfDataByDTC Type	Status of the operation to retrieve the size of freeze frame data.
<b>Description:</b>	Gets the size of freeze frame data by DTC.	

**Table 8.93: Dem\_DcmGetSizeOfFreezeFrameByDTC**

] (SRS\_Diag\_04066)

### 8.3.4.2.5 Dem\_DcmGetExtendedDataRecordByDTC

[SWS\_Dem\_00239] [

<b>Service name:</b>	Dem_DcmGetExtendedDataRecordByDTC	
<b>Syntax:</b>	Dem_ReturnGetExtendedDataRecordByDTCType Dem_DcmGetExtendedDataRecordByDTC ( uint32 DTC, Dem_DTCOriginType DTCOrigin, uint8 ExtendedDataNumber, uint8* DestBuffer, uint16* BufSize )	
<b>Service ID[hex]:</b>	0x20	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTC	Diagnostic Trouble Code in UDS format.

	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.
	ExtendedDataNumber	Identification/Number of requested extended data record. The values 0xFE and 0xFF are not allowed.
<b>Parameters (inout):</b>	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.
<b>Parameters (out):</b>	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 raw hexadecimal values and contains no header-information.
<b>Return value:</b>	Dem_ReturnGetExtendedDataRecordByDTCType	Status of the operation to retrieve extended data by DTC.
<b>Description:</b>	Gets extended data by DTC. The function stores the data in the provided DestBuffer.	

**Table 8.94: Dem\_DcmGetExtendedDataRecordByDTC**

」(SRS\_Diag\_04066)

### 8.3.4.2.6 Dem\_DcmGetSizeOfExtendedDataRecordByDTC

[SWS\_Dem\_00240] 「

<b>Service name:</b>	Dem_DcmGetSizeOfExtendedDataRecordByDTC	
<b>Syntax:</b>	Dem_ReturnGetSizeOfDataByDTCType Dem_DcmGetSizeOfExtendedDataRecordByDTC( uint32 DTC, Dem_DTCOriginType DTCOrigin, uint8 ExtendedDataNumber, uint16* SizeOfExtendedDataRecord )	
<b>Service ID[hex]:</b>	0x21	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTC DTCOrigin  ExtendedDataNumber	Diagnostic Trouble Code in UDS format. If the Dem supports more than one event memory, this parameter is used to select the source memory the DTCs shall be read from. Identification/Number of requested extended data record. Additionally the values 0xFE and 0xFF are explicitly allowed to request the overall size of all OBD records / all records.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	SizeOfExtendedDataRecord	Size of the requested extended data record(s) including record number size

<b>Return value:</b>	Dem_ReturnGet SizeOfDataByDTC Type	Status of the operation to retrieve the size of extended data.
<b>Description:</b>	Gets the size of extended data by DTC.	

**Table 8.95: Dem\_DcmGetSizeOfExtendedDataRecordByDTC**

」(SRS\_Diag\_04066)

### 8.3.4.3 DTC storage

#### 8.3.4.3.1 Dem\_DcmCheckClearParameter

[SWS\_Dem\_01196] 「

<b>Service name:</b>	Dem_DcmCheckClearParameter	
<b>Syntax:</b>	<pre>Dem_ReturnClearDTCType Dem_DcmCheckClearParameter(     uint32 DTC,     Dem_DTCFormatType DTCFormat,     Dem_DTCOriginType DTCOrigin )</pre>	
<b>Service ID[hex]:</b>	0x7b	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	DTC  DTCFormat DTCOrigin	Defines the DTC in respective format, that shall be cleared from the event memory. Either a single DTC or a DTC group may be passed.  Defines the input-format of the provided DTC value. This parameter is used to select the source memory the DTCs shall be cleared from.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Dem_ReturnClearDTCType	Returns the result of the parameter check when performing a clear command with same parameters (refer to Dem_DcmClearDTC).  Only the following return values will be used: DEM_CLEAR_OK: DTC successfully cleared DEM_CLEAR_WRONG_DTC: DTC value not existing (in this format) DEM_CLEAR_WRONG_DTCORIGIN: Wrong DTC origin DEM_CLEAR_FAILED: In case of general errors with clear parameters
<b>Description:</b>	Performs a parameter check and gives the result which would also be returned by calling clear with same parameters.	

**Table 8.96: Dem\_DcmCheckClearParameter**

」()

### 8.3.4.3.2 Dem\_DcmClearDTC

[SWS\_Dem\_00241] [

<b>Service name:</b>	Dem_DcmClearDTC	
<b>Syntax:</b>	<pre>Dem_ReturnClearDTCType Dem_DcmClearDTC(     uint32 DTC,     Dem_DTCFormatType DTCFormat,     Dem_DTCOriginType DTCOrigin )</pre>	
<b>Service ID[hex]:</b>	0x22	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTC  DTCFormat DTCOrigin	Defines the DTC in respective format, that shall be cleared from the event memory. If the DTC fits to a DTC group number, all DTCs of the group shall be cleared.  Defines the input-format of the provided DTC value. If the Dem supports more than one event memory this parameter is used to select the source memory the DTCs shall be read from.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Dem_ReturnClearDTCType	Status of the operation of type Dem_ReturnClearDTCType.
<b>Description:</b>	Clears single DTCs, as well as groups of DTCs. This API is intended for the Dcm. It can only be used through the RTE (due to work-around described below SWS_Dem_00009), and therefore no declaration is exported via Dem_Dcm.h.	

Table 8.97: Dem\_DcmClearDTC

] (SRS\_Diag\_04066)

### 8.3.4.3.3 Dem\_DcmDisableDTCSetting

[SWS\_Dem\_00242] [

<b>Service name:</b>	Dem_DcmDisableDTCSetting	
<b>Syntax:</b>	<pre>Dem_ReturnControlDTCSettingType Dem_DcmDisableDTCSetting(     uint32 DTGroup,     Dem_DTCKindType DTCKind )</pre>	
<b>Service ID[hex]:</b>	0x24	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTGroup	Defines the group of DTC that shall be disabled to store in event memory.

	DTCKind	This parameter defines the requested DTC kind, either only OBD-relevant DTCs or all DTCs
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Dem_ReturnControl DTCSettingType	Returns status of the operation
<b>Description:</b>	Disables the DTC setting for a DTC group.	

**Table 8.98: Dem\_DcmDisableDTCSetting**

]()

### 8.3.4.3.4 Dem\_DcmEnableDTCSetting

[SWS\_Dem\_00243] [

<b>Service name:</b>	Dem_DcmEnableDTCSetting	
<b>Syntax:</b>	Dem_ReturnControlDTCSettingType Dem_DcmEnableDTCSetting( uint32 DTCGroup, Dem_DTCKindType DTCKind )	
<b>Service ID[hex]:</b>	0x25	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTCGroup  DTCKind	Defines the group of DTC that shall be enabled to store in event memory.  This parameter defines the requested DTC kind, either only OBD-relevant DTCs or all DTCs
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Dem_ReturnControl DTCSettingType	Returns the status of the operation
<b>Description:</b>	Enables the DTC setting for a DTC group. This API is intended for the Dcm. It can only be used through the RTE (due to work-around described below SWS_Dem_00035), and therefore no declaration is exported via Dem_Dcm.h.	

**Table 8.99: Dem\_DcmEnableDTCSetting**

]()

## 8.3.5 OBD-specific Dcm <=> Dem Interfaces

### 8.3.5.1 Dem\_DcmGetInfoTypeValue08

[SWS\_Dem\_00316] [

<b>Service name:</b>	Dem_DcmGetInfoTypeValue08	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_DcmGetInfoTypeValue08 (   Dcm_OpStatusType OpStatus,   uint8* Iumprdata08,   uint8* Iumprdata08BufferSize )</pre>	
<b>Service ID[hex]:</b>	0x6b	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	OpStatus	Only DCM_INITIAL will appear, because this API behaves synchronous.
<b>Parameters (inout):</b>	Iumprdata08BufferSize	The maximum number of data bytes that can be written to the Iumprdata08 Buffer.
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned.
<b>Description:</b>	<p>Service is used for requesting IUMPR data according to Info-Type \$08. This interface is derived from the prototype &lt;Module&gt;_GetInfotypeValueData() defined by the Dcm. Therefore Dcm_OpStatusType and Std_ReturnType are contained. API is needed in OBD-relevant ECUs only.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/Dem-General.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p>	

**Table 8.100: Dem\_DcmGetInfoTypeValue08**

]()

### 8.3.5.2 Dem\_DcmGetInfoTypeValue0B

**[SWS\_Dem\_00317]**

<b>Service name:</b>	Dem_DcmGetInfoTypeValue0B	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_DcmGetInfoTypeValue0B (   Dcm_OpStatusType OpStatus,   uint8* Iumprdata0B,   uint8* Iumprdata0BBufferSize )</pre>	
<b>Service ID[hex]:</b>	0x6c	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	OpStatus	Only DCM_INITIAL will appear, because this API behaves synchronous.
<b>Parameters (inout):</b>	Iumprdata0BBufferSize	The maximum number of data bytes that can be written to the Iumprdata0B Buffer.
<b>Parameters (out):</b>	Iumprdata0B	Buffer containing the number of data elements (as defined in ISO-15031-5) and contents of InfoType \$0B. The buffer is provided by the Dcm.

<b>Return value:</b>	Std_ReturnType	Always E_OK is returned.
<b>Description:</b>	Service is used for requesting IUMPR data according to Info-Type \$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}	

**Table 8.101: Dem\_DcmGetInfotypeValue0B**

↴()

### 8.3.5.3 Dem\_DcmReadDataOfPID01

[SWS\_Dem\_00318] ↴

<b>Service name:</b>	Dem_DcmReadDataOfPID01	
<b>Syntax:</b>	Std_ReturnType Dem_DcmReadDataOfPID01 ( uint8* PID01value )	
<b>Service ID[hex]:</b>	0x61	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	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}	

**Table 8.102: Dem\_DcmReadDataOfPID01**

↴()

### 8.3.5.4 Dem\_DcmReadDataOfPID1C

[SWS\_Dem\_00325] [

<b>Service name:</b>	Dem_DcmReadDataOfPID1C	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_DcmReadDataOfPID1C (     uint8* PID1Cvalue )</pre>	
<b>Service ID[hex]:</b>	0x63	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	PID1Cvalue	Buffer containing the contents of PID \$1C computed by the Dem. The value of PID\$1C is configuration within DemOBDCompliance. 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.
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	<p>Service to report the value of PID \$1C computed by the Dem. API is needed in OBD-relevant ECUs only.</p> <p>API Availability: This API will be available only if  <math>\{\{ecuc(Dem/DemGeneral.DemOBDSupport)\} \neq DEM_OBD_NO_OBD_SUPPORT\}</math></p>	

Table 8.103: Dem\_DcmReadDataOfPID1C

]()

### 8.3.5.5 Dem\_DcmReadDataOfPID21

[SWS\_Dem\_00319] [

<b>Service name:</b>	Dem_DcmReadDataOfPID21	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_DcmReadDataOfPID21 (     uint8* PID21value )</pre>	
<b>Service ID[hex]:</b>	0x64	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	

<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	<p>Service to report the value of PID \$21 computed by the Dem. API is needed in OBD-relevant ECUs only.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p>	

**Table 8.104: Dem\_DcmReadDataOfPID21**

]()

### 8.3.5.6 Dem\_DcmReadDataOfPID30

[SWS\_Dem\_00320] [

<b>Service name:</b>	Dem_DcmReadDataOfPID30	
<b>Syntax:</b>	Std_ReturnType Dem_DcmReadDataOfPID30 ( uint8* PID30value )	
<b>Service ID[hex]:</b>	0x65	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	<p>Service to report the value of PID \$30 computed by the Dem. API is needed in OBD-relevant ECUs only.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p>	

**Table 8.105: Dem\_DcmReadDataOfPID30**

]()

### 8.3.5.7 Dem\_DcmReadDataOfPID31

[SWS\_Dem\_00321] [

<b>Service name:</b>	Dem_DcmReadDataOfPID31	
<b>Syntax:</b>	Std_ReturnType Dem_DcmReadDataOfPID31 ( uint8* PID31value )	
<b>Service ID[hex]:</b>	0x66	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	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_NO_OBD_SUPPORT)	

Table 8.106: Dem\_DcmReadDataOfPID31

]()

### 8.3.5.8 Dem\_DcmReadDataOfPID41

[SWS\_Dem\_00322] [

<b>Service name:</b>	Dem_DcmReadDataOfPID41	
<b>Syntax:</b>	Std_ReturnType Dem_DcmReadDataOfPID41 ( uint8* PID41value )	
<b>Service ID[hex]:</b>	0x67	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	

<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	<p>Service to report the value of PID \$41 computed by the Dem. API is needed in OBD-relevant ECUs only.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p>	

**Table 8.107: Dem\_DcmReadDataOfPID41**

]()

### 8.3.5.9 Dem\_DcmReadDataOfPID4D

[SWS\_Dem\_00323] [

<b>Service name:</b>	Dem_DcmReadDataOfPID4D	
<b>Syntax:</b>	Std_ReturnType Dem_DcmReadDataOfPID4D ( uint8* PID4Dvalue )	
<b>Service ID[hex]:</b>	0x68	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	<p>Service to report the value of PID \$4D computed by the Dem. API is needed in OBD-relevant ECUs only.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p>	

**Table 8.108: Dem\_DcmReadDataOfPID4D**

]()

### 8.3.5.10 Dem\_DcmReadDataOfPID4E

[SWS\_Dem\_00324] [

<b>Service name:</b>	Dem_DcmReadDataOfPID4E	
<b>Syntax:</b>	Std_ReturnType Dem_DcmReadDataOfPID4E ( uint8* PID4Value )	
<b>Service ID[hex]:</b>	0x69	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	PID4Value	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.
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	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_NO_OBD_SUPPORT)	

Table 8.109: Dem\_DcmReadDataOfPID4E

]()

### 8.3.5.11 Dem\_DcmReadDataOfPID91

[SWS\_Dem\_01187] [

<b>Service name:</b>	Dem_DcmReadDataOfPID91	
<b>Syntax:</b>	Std_ReturnType Dem_DcmReadDataOfPID91 ( uint8* PID91value )	
<b>Service ID[hex]:</b>	0x6a	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	

<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	<p>Service to report the value of PID \$91 computed by the Dem. API is needed in OBD-relevant ECUs only.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p>	

**Table 8.110: Dem\_DcmReadDataOfPID91**

]()

### 8.3.5.12 Dem\_DcmReadDataOfOBDFreezeFrame

[SWS\_Dem\_00327] [

<b>Service name:</b>	Dem_DcmReadDataOfOBDFreezeFrame	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_DcmReadDataOfOBDFreezeFrame(     uint8 PID,     uint8 DataElementIndexOfPID,     uint8* DestBuffer,     uint16* BufSize )</pre>	
<b>Service ID[hex]:</b>	0x52	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	PID  DataElementIndex OfPID	This parameter is an identifier for a PID as defined in ISO15031-5.  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).
<b>Parameters (inout):</b>	DestBuffer  BufSize	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.  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.
<b>Parameters (out):</b>	None	

<b>Return value:</b>	Std_ReturnType	E_OK Freeze frame data was successfully reported E_NOT_OK Freeze frame data was not successfully reported
<b>Description:</b>	<p>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.</p> <p>API is needed in OBD-relevant ECUs only.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p>	

**Table 8.111: Dem\_DcmReadDataOfOBDFreezeFrame**

↴()

### 8.3.5.13 Dem\_DcmGetDTCOfOBDFreezeFrame

**[SWS\_Dem\_00624]** ↴

<b>Service name:</b>	Dem_DcmGetDTCOfOBDFreezeFrame	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_DcmGetDTCOfOBDFreezeFrame (     uint8 FrameNumber,     uint32* DTC,     Dem_DTCFormatType DTCFormat )</pre>	
<b>Service ID[hex]:</b>	0x53	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	FrameNumber  DTCFormat	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.  Output format of the DTC value.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	E_OK: operation was successful E_NOT_OK: no DTC available
<b>Description:</b>	<p>Gets DTC by freeze frame record number.</p> <p>API is needed in OBD-relevant ECUs only.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p>	

**Table 8.112: Dem\_DcmGetDTCOfOBDFreezeFrame**

↴()

### 8.3.5.14 Dem\_DcmGetAvailableOBDMIDs

[SWS\_Dem\_00766] [

<b>Service name:</b>	Dem_DcmGetAvailableOBDMIDs	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_DcmGetAvailableOBDMIDs (     uint8 Obdmid,     uint32* Obdmidvalue )</pre>	
<b>Service ID[hex]:</b>	0xa3	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	Obdmid	Availability OBDMID (\$00,\$20, \$40...)
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	Obdmidvalue	Bit coded information on the support of OBDMIDs.
<b>Return value:</b>	Std_ReturnType	E_OK: Report of DTR result successful E_NOT_OK: Report of DTR result failed
<b>Description:</b>	<p>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.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/Dem-General.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p>	

Table 8.113: Dem\_DcmGetAvailableOBDMIDs

]()

### 8.3.5.15 Dem\_DcmGetNumTIDsOfOBDMID

[SWS\_Dem\_00767] [

<b>Service name:</b>	Dem_DcmGetNumTIDsOfOBDMID	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_DcmGetNumTIDsOfOBDMID (     uint8 Obdmid,     uint8* numberoftIDs )</pre>	
<b>Service ID[hex]:</b>	0xa4	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	Obdmid	OBDMID subject of the request to identify the number of assigned TIDs
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	numberoftIDs	Number of assigned TIDs for the requested OBDMID. Used as loop value for the DCM to retrieve all OBD/TID result data.
<b>Return value:</b>	Std_ReturnType	E_OK: get number of TIDs successful E_NOT_OK: get number of TIDs failed

<b>Description:</b> 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/Dem-General.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)
--

**Table 8.114: Dem\_DcmGetNumTIDsOfOBDMID**

]()

### 8.3.5.16 Dem\_DcmGetDTRData

[SWS\_Dem\_00768] [

<b>Service name:</b> <b>Syntax:</b> <pre>Std_ReturnType Dem_DcmGetDTRData(     uint8 Obdmid,     uint8 TIDindex,     uint8* TIDvalue,     uint8* UaSID,     uint16* Testvalue,     uint16* Lowlimvalue,     uint16* Upplimvalue )</pre>										
<b>Service ID[hex]:</b> 0xa5 <b>Sync/Async:</b> Synchronous <b>Reentrancy:</b> Reentrant										
<b>Parameters (in):</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">Obdmid</td> <td style="padding: 2px;">Identification of a DTR element by assigned DTRId.</td> </tr> <tr> <td style="padding: 2px;">TIDindex</td> <td style="padding: 2px;">Index of the TID within the DEM. Runs from 0 to "numberOfTIDs" obtained in the call to Dem_DcmGetNumTIDsOfOBDMID()</td> </tr> </table>	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()						
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()									
<b>Parameters (inout):</b> None										
<b>Parameters (out):</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">TIDvalue</td> <td style="padding: 2px;">TID to be put on the tester reponse</td> </tr> <tr> <td style="padding: 2px;">UaSID</td> <td style="padding: 2px;">UaSID to be put on the tester reponse</td> </tr> <tr> <td style="padding: 2px;">Testvalue</td> <td style="padding: 2px;">Latest test result</td> </tr> <tr> <td style="padding: 2px;">Lowlimvalue</td> <td style="padding: 2px;">Lower limit value associated to the latest test result</td> </tr> <tr> <td style="padding: 2px;">Upplimvalue</td> <td style="padding: 2px;">Upper limit value associated to the latest test result</td> </tr> </table>	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
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									
<b>Return value:</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">Std_ReturnType</td> <td style="padding: 2px;">E_OK: Report of DTR result successful E_NOT_OK: Report of DTR result failed</td> </tr> </table>	Std_ReturnType	E_OK: Report of DTR result successful E_NOT_OK: Report of DTR result failed								
Std_ReturnType	E_OK: Report of DTR result successful E_NOT_OK: Report of DTR result failed									
<b>Description:</b> 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)										

**Table 8.115: Dem\_DcmGetDTRData**

]()

### 8.3.6 Interface J1939Dcm <=> Dem

#### 8.3.6.1 Access DTCs and Status Information

##### 8.3.6.1.1 Dem\_J1939DcmSetDTCFilter

[SWS\_Dem\_00970] [

<b>Service name:</b>	Dem_J1939DcmSetDTCFilter	
<b>Syntax:</b>	<pre>Dem_ReturnSetFilterType Dem_J1939DcmSetDTCFilter(     Dem_J1939DcmDTCStatusFilterType DTCStatusFilter,     Dem_DTCKindType DTCKind,     uint8 node,     Dem_J1939DcmLampStatusType* LampStatus )</pre>	
<b>Service ID[hex]:</b>	0x90	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTCStatusFilter  DTCKind  node	The following types are available:  DEM_J1939DTC_ACTIVE DEM_J1939DTC_PREVIOUSLY_ACTIVE DEM_J1939DTC_PENDING DEM_J1939DTC_PERMANENT DEM_J1939DTC_CURRENTLY_ACTIVE  Defines the functional group of DTCs to be reported (e.g. all DTC, OBD-relevant DTC)  Nm node Id
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	LampStatus	Receives the commulated lamp status.
<b>Return value:</b>	Dem_ReturnSetFilterType	Status of the operation to (re-)set a DTC filter.
<b>Description:</b>	The function sets the DTC filter for a specific node and returns the composite lamp status of the filtered DTCs.	

Table 8.116: Dem\_J1939DcmSetDTCFilter

]()

##### 8.3.6.1.2 Dem\_J1939DcmGetNumberOfFilteredDTC

[SWS\_Dem\_00972] [

<b>Service name:</b>	Dem_J1939DcmGetNumberOfFilteredDTC
----------------------	------------------------------------

<b>Syntax:</b>	Dem_ReturnGetNumberOfFilteredDTCType Dem_J1939DcmGetNumberOfFilteredDTC( uint16* NumberOfFilteredDTC )	
<b>Service ID[hex]:</b>	0x91	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	NumberOfFilteredDTC	The number of DTCs matching the defined status mask.
<b>Return value:</b>	Dem_ReturnGetNumberOfFilteredDTCType	Status of the operation to retrieve a number of DTC from the Dem
<b>Description:</b>	Gets the number of currently filtered DTCs set by the function Dem_J1939DcmSetDTCFilter.	

**Table 8.117: Dem\_J1939DcmGetNumberOfFilteredDTC**

↴()

### 8.3.6.1.3 Dem\_J1939DcmGetNextFilteredDTC

[SWS\_Dem\_00973] ↵

<b>Service name:</b>	Dem_J1939DcmGetNextFilteredDTC	
<b>Syntax:</b>	Dem_ReturnGetNextFilteredElementType Dem_J1939DcmGetNextFilteredDTC( uint32* J1939DTC, uint8* OccurrenceCounter )	
<b>Service ID[hex]:</b>	0x92	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	J1939DTC  OccurrenceCounter	Receives the J1939DTC value. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.  This parameter receives the corresponding occurrence counter. If the return value of the function call is other than DEM_FILTERED_OK this parameter does not contain valid data.
<b>Return value:</b>	Dem_ReturnGetNextFilteredElementType	Status of the operation to retrieve a DTC from the Dem.
<b>Description:</b>	Gets the next filtered J1939 DTC.	

**Table 8.118: Dem\_J1939DcmGetNextFilteredDTC**

↴()

### 8.3.6.1.4 Dem\_J1939DcmFirstDTCwithLampStatus

[SWS\_Dem\_00974] ↴

<b>Service name:</b>	Dem_J1939DcmFirstDTCwithLampStatus	
<b>Syntax:</b>	<pre>void Dem_J1939DcmFirstDTCwithLampStatus(     uint8 node )</pre>	
<b>Service ID[hex]:</b>	0x93	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	node	Nm node Id of requesting client
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	The function sets the filter to the first applicable DTC for the DM31 response for a specific node.	

Table 8.119: Dem\_J1939DcmFirstDTCwithLampStatus

↴()

### 8.3.6.1.5 Dem\_J1939DcmGetNextDTCwithLampStatus

[SWS\_Dem\_00975] ↴

<b>Service name:</b>	Dem_J1939DcmGetNextDTCwithLampStatus	
<b>Syntax:</b>	<pre>Dem_ReturnGetNextFilteredElementType Dem_J1939DcmGet NextDTCwithLampStatus(     Dem_J1939DcmLampStatusType* LampStatus,     uint32* J1939DTC,     uint8* OccurrenceCounter )</pre>	
<b>Service ID[hex]:</b>	0x94	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	LampStatus  J1939DTC	<p>Receives the lamp status returned by this function. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.</p> <p>Receives the J1939DTC value. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.</p>

	OccurrenceCounter	This parameter receives the corresponding occurrence counter. If the return value of the function call is other than DEM_FILTERED_OK this parameter does not contain valid data.
<b>Return value:</b>	Dem_ReturnGetNextFilteredElementType	Status of the operation to retrieve a DTC from the Dem.
<b>Description:</b>	Gets the next filtered J1939 DTC for DM31 including current LampStatus.	

**Table 8.120: Dem\_J1939DcmGetNextDTCwithLampStatus**

↴()

### 8.3.6.2 DTC storage

#### 8.3.6.2.1 Dem\_J1939DcmClearDTC

[SWS\_Dem\_00976] ↴

<b>Service name:</b>	Dem_J1939DcmClearDTC	
<b>Syntax:</b>	Dem_ReturnClearDTCType Dem_J1939DcmClearDTC( Dem_J1939DcmSetClearFilterType DTCTypeFilter, uint8 node )	
<b>Service ID[hex]:</b>	0x95	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTCTypeFilter node	The following types are available: DEM_J1939DTC_CLEAR_ALL DEM_J1939DTC_CLEAR_PREVIOUSLY_ACTIVE Nm node Id of requesting client
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Dem_ReturnClearDTCType	Status of the operation of type Dem_ReturnClearDTCType.
<b>Description:</b>	Clears active DTCs as well as previously active DTCs.	

**Table 8.121: Dem\_J1939DcmClearDTC**

↴()

#### 8.3.6.2.2 Dem\_J1939DcmSetFreezeFrameFilter

[SWS\_Dem\_00977] ↴

<b>Service name:</b>	Dem_J1939DcmSetFreezeFrameFilter
----------------------	----------------------------------

<b>Syntax:</b>	Dem_ReturnSetFilterType Dem_J1939DcmSetFreezeFrame Filter( Dem_J1939DcmSetFreezeFrameFilterType FreezeFrameKind, uint8 node )	
<b>Service ID[hex]:</b>	0x96	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	FreezeFrameKind  node	The following types are available: DEM_J1939DCM_FREEZEFRAME DEM_J1939DCM_EXPANDED_FREEZEFRAME DEM_J1939DCM_SPNS_IN_EXPANDED_FREEZEFRAME Nm node Id of requesting client
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Dem_ReturnSetFilterType	Status of the operation to (re-)set a FreezeFrame filter.
<b>Description:</b>	The function sets the FreezeFrame filter for a specific node.	

**Table 8.122: Dem\_J1939DcmSetFreezeFrameFilter**

↴()

### 8.3.6.2.3 Dem\_J1939DcmGetNextFreezeFrame

[SWS\_Dem\_00978] ↵

<b>Service name:</b>	Dem_J1939DcmGetNextFreezeFrame	
<b>Syntax:</b>	Dem_ReturnGetNextFilteredElementType Dem_J1939DcmGetNextFreezeFrame( uint32* J1939DTC, uint8* OccurrenceCounter, uint8* DestBuffer, uint16* BufSize )	
<b>Service ID[hex]:</b>	0x97	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	DestBuffer  BufSize	This parameter contains a byte pointer that points to the buffer, to which the freeze frame data record shall be written to.  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 DestBuffer
<b>Parameters (out):</b>	J1939DTC	Receives the J1939DTC value. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.

	OccurrenceCounter	This parameter receives the corresponding occurrence counter. If the return value of the function call is other than DEM_FILTERED_OK this parameter does not contain valid data.
<b>Return value:</b>	Dem_ReturnGetNextFilteredElementType	Status of the operation to retrieve freeze frame data by DTC.
<b>Description:</b>	Gets next freeze frame data. The function stores the data in the provided DestBuffer.	

**Table 8.123: Dem\_J1939DcmGetNextFreezeFrame**

]()

### 8.3.6.2.4 Dem\_J1939DcmGetNextSPNInFreezeFrame

[SWS\_Dem\_00979] [

<b>Service name:</b>	Dem_J1939DcmGetNextSPNInFreezeFrame	
<b>Syntax:</b>	Dem_ReturnGetNextFilteredElementType Dem_J1939DcmGetNextSPNInFreezeFrame( uint32* SPNSupported, uint8* SPNDataLength )	
<b>Service ID[hex]:</b>	0x98	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	SPNSupported SPNDataLength	This parameter contains the next SPN in the ExpandedFreezeFrame This parameter contains the corresponding dataLength of the SPN
<b>Return value:</b>	Dem_ReturnGetNextFilteredElementType	Status of the operation to retrieve freeze frame data by DTC.
<b>Description:</b>	Gets next SPN.	

**Table 8.124: Dem\_J1939DcmGetNextSPNInFreezeFrame**

]()

### 8.3.6.3 Reporting

#### 8.3.6.3.1 Dem\_J1939DcmSetRatioFilter

[SWS\_Dem\_00980] [

<b>Service name:</b>	Dem_J1939DcmSetRatioFilter	
<b>Syntax:</b>	<pre>Dem_ReturnSetFilterType Dem_J1939DcmSetRatioFilter(     uint16* IgnitionCycleCounter,     uint16* OBDMonitoringConditionsEncountered,     uint8 node )</pre>	
<b>Service ID[hex]:</b>	0x99	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	node	Nm node Id of requesting client
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	IgnitionCycle Counter OBDMonitoringConditionsEncountered	Ignition Cycle Counter OBD Monitoring Conditions Encountered
<b>Return value:</b>	Dem_ReturnSetFilterType	Status of the operation to (re-)set a DTC filter.
<b>Description:</b>	The function sets the Ratio filter for a specific node and returns the corresponding Ignition Cycle Counter and General Denominator.	

**Table 8.125: Dem\_J1939DcmSetRatioFilter**

↴()

### 8.3.6.3.2 Dem\_J1939DcmGetNextFilteredRatio

[SWS\_Dem\_00981] ↵

<b>Service name:</b>	Dem_J1939DcmGetNextFilteredRatio	
<b>Syntax:</b>	<pre>Dem_ReturnGetNextFilteredElementType Dem_J1939DcmGet NextFilteredRatio(     uint16* SPN,     uint16* Numerator,     uint16* Denominator )</pre>	
<b>Service ID[hex]:</b>	0x9a	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	SPN  Numerator	Receives the SPN of the applicable system monitor. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.  Receives the Numerator of the applicable system monitor. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.

	Denominator	Receives the Denominator of the applicable system monitor. If the return value of the function is other than DEM_FILTERED_OK this parameter does not contain valid data.
<b>Return value:</b>	Dem_ReturnGetNextFilteredElementType	0x00 DEM_FILTERED_OK Ratio available in out parameter 0x01 DEM_FILTERED_NO_FURTHER_ELEMENT No further element available
<b>Description:</b>	Gets the next filtered Ratio.	

**Table 8.126: Dem\_J1939DcmGetNextFilteredRatio**

↴()

### 8.3.6.3.3 Dem\_J1939DcmReadDiagnosticReadiness1

[SWS\_Dem\_00982] ↴

<b>Service name:</b>	Dem_J1939DcmReadDiagnosticReadiness1	
<b>Syntax:</b>	Std_ReturnType Dem_J1939DcmReadDiagnosticReadiness1( Dem_J1939DcmDiagnosticReadiness1Type* DataValue, uint8 node )	
<b>Service ID[hex]:</b>	0x9b	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	node	Nm node Id of requesting client
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	DataValue	Buffer of 8 bytes containing the contents of Diagnostic Readiness 1 (DM05) computed by the Dem.
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation failed
<b>Description:</b>	Service to report the value of Diagnostic Readiness 1 (DM05) computed by the Dem.	

**Table 8.127: Dem\_J1939DcmReadDiagnosticReadiness1**

↴()

### 8.3.6.3.4 Dem\_J1939DcmReadDiagnosticReadiness2

[SWS\_Dem\_00983] ↴

<b>Service name:</b>	Dem_J1939DcmReadDiagnosticReadiness2	
<b>Syntax:</b>	Std_ReturnType Dem_J1939DcmReadDiagnosticReadiness2( Dem_J1939DcmDiagnosticReadiness2Type* DataValue, uint8 node )	

<b>Service ID[hex]:</b>	0x9c	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	node	Nm node Id of requesting client
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	DataValue	Buffer of 8 bytes containing the contents of Diagnostic Readiness 2 (DM21) computed by the Dem.
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation failed
<b>Description:</b>	Service to report the value of Diagnostic Readiness 2 (DM21) computed by the Dem.	

**Table 8.128: Dem\_J1939DcmReadDiagnosticReadiness2**

]()

### 8.3.6.3.5 Dem\_J1939DcmReadDiagnosticReadiness3

[SWS\_Dem\_00770] [

<b>Service name:</b>	Dem_J1939DcmReadDiagnosticReadiness3	
<b>Syntax:</b>	Std_ReturnType Dem_J1939DcmReadDiagnosticReadiness3( Dem_J1939DcmDiagnosticReadiness3Type* DataValue, uint8 node )	
<b>Service ID[hex]:</b>	0x9d	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	node	Nm node Id of requesting client
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	DataValue	Buffer of 8 bytes containing the contents of Diagnostic Readiness 3 (DM26) computed by the Dem.
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation failed
<b>Description:</b>	Service to report the value of Diagnostic Readiness 3 (DM26) computed by the Dem.	

**Table 8.129: Dem\_J1939DcmReadDiagnosticReadiness3**

]()

## 8.3.7 Interface Dlt <=> Dem

### 8.3.7.1 Dem\_DltGetMostRecentFreezeFrameRecordData

[SWS\_Dem\_00636] [

<b>Service name:</b>	Dem_DltGetMostRecentFreezeFrameRecordData	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_DltGetMostRecentFreezeFrameRecord Data(     Dem_EventIdType EventId,     uint8* DestBuffer,     uint16* BufSize )</pre>	
<b>Service ID[hex]:</b>	0x41	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	EventId	Identification of an event by assigned EventId.
<b>Parameters (inout):</b>	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.
<b>Parameters (out):</b>	DestBuffer	This parameter contains a byte pointer that points to the buffer, to which the freeze frame record shall be written to. The format is raw hexadecimal values and contains no header-information.
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful. DEM_E_NODATAAVAILABLE: The requested event data is not currently stored (but the request was valid). DEM_E_PENDING: The requested data is currently transported from NvM and needs to be requested again.
<b>Description:</b>	Gets the data of an most recent freeze frame record by event. The OBD-II freeze frame is not returned by this function.	

**Table 8.130: Dem\_DltGetMostRecentFreezeFrameRecord**

] ([SRS\\_Diag\\_04099](#))

### 8.3.7.2 Dem\_DltGetAllExtendedDataRecords

[[SWS\\_Dem\\_00637](#)] [

<b>Service name:</b>	Dem_DltGetAllExtendedDataRecords	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_DltGetAllExtendedDataRecords (     Dem_EventIdType EventId,     uint8* DestBuffer,     uint16* BufSize )</pre>	
<b>Service ID[hex]:</b>	0x40	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	EventId	Identification of an event by assigned EventId.

<b>Parameters (inout):</b>	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.
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful. DEM_E_NODATAAVAILABLE: The requested event data is not currently stored (but the request was valid). DEM_E_PENDING: The requested data is currently transported from NvM and needs to be requested again.
<b>Description:</b>	Gets the data of all extended data records of an event.	

**Table 8.131: Dem\_DltGetAllExtendedDataRecords**

] ([SRS\\_Diag\\_04099](#))

### 8.3.8 OBD-specific Interfaces

#### 8.3.8.1 Dem\_SetEventDisabled

[SWS\_Dem\_00312] [

<b>Service name:</b>	Dem_SetEventDisabled	
<b>Syntax:</b>	Std_ReturnType Dem_SetEventDisabled( Dem_EventIdType EventId )	
<b>Service ID[hex]:</b>	0x51	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different EventIds. Non reentrant for the same EventId.	
<b>Parameters (in):</b>	EventId	Identification of an event by assigned EventId.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK set of event to disabled was successfull. E_NOT_OK set of event disabled failed
<b>Description:</b>	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)	

**Table 8.132: Dem\_SetEventDisabled**

]()

### 8.3.8.2 Dem\_ReplIUMPRFaultDetect

[SWS\_Dem\_00313] [

<b>Service name:</b>	Dem_ReplIUMPRFaultDetect	
<b>Syntax:</b>	Std_ReturnType Dem_ReplIUMPRFaultDetect ( Dem_RatioIDType RatioID )	
<b>Service ID[hex]:</b>	0x73	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different RatioIDs. Non reentrant for the same RatioID.	
<b>Parameters (in):</b>	RatioID	Ratio Identifier reporting that a respective monitor could have found a fault - only used when interface option "API" is selected
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK report of IUMPR result was successfully reported
<b>Description:</b>	Service for reporting that faults are possibly found because all conditions are fulfilled. 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)	

Table 8.133: Dem\_ReplIUMPRFaultDetect

]()

### 8.3.8.3 Dem\_SetIUMPRDenCondition

[SWS\_Dem\_00733] [

<b>Service name:</b>	Dem_SetIUMPRDenCondition	
<b>Syntax:</b>	Std_ReturnType Dem_SetIUMPRDenCondition ( Dem_IumprDenomCondIdType ConditionId, Dem_IumprDenomCondStatusType ConditionStatus )	
<b>Service ID[hex]:</b>	0xae	
<b>Sync/Async:</b>	Synchronous /Asynchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	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)
<b>Parameters (inout):</b>	None	

<b>Parameters (out):</b>	None	
<b>Return value:</b>	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.
<b>Description:</b>	<p>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.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p>	

**Table 8.134: Dem\_SetIUMPRDenCondition**

]()

### 8.3.8.4 Dem\_GetIUMPRDenCondition

**[SWS\_Dem\_00734]**

<b>Service name:</b>	Dem_GetIUMPRDenCondition	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_GetIUMPRDenCondition(     Dem_IumprDenomCondIdType ConditionId,     Dem_IumprDenomCondStatusType* ConditionStatus )</pre>	
<b>Service ID[hex]:</b>	0xaf	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	ConditionId	Identification of a IUMPR denominator condition ID (General Denominator, Cold start, EVAP, 500mi).
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	ConditionStatus	Status of the IUMPR denominator condition (Not-reached, reached, not reachable / inhibited)
<b>Return value:</b>	Std_ReturnType	E_OK: get of IUMPR denominator condition status was successful E_NOT_OK: get of condition status failed
<b>Description:</b>	<p>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.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p>	

**Table 8.135: Dem\_GetIUMPRDenCondition**

]()

### 8.3.8.5 Dem\_ReplIUMPRDenLock

[SWS\_Dem\_00314] [

<b>Service name:</b>	Dem_ReplIUMPRDenLock	
<b>Syntax:</b>	Std_ReturnType Dem_ReplIUMPRDenLock ( Dem_RatioIDType RatioID )	
<b>Service ID[hex]:</b>	0x71	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	Ratioid	Ratio Identifier reporting that specific denominator is locked (for physical reasons - e.g. temperature conditions or minimum activity)
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK report of IUMPR denominator status was successfully reported E_NOK report of IUMPR denominator status was not successfully reported
<b>Description:</b>	Service is used to lock 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)	

Table 8.136: Dem\_ReplIUMPRDenLock

]()

### 8.3.8.6 Dem\_ReplIUMPRDenRelease

[SWS\_Dem\_00315] [

<b>Service name:</b>	Dem_ReplIUMPRDenRelease	
<b>Syntax:</b>	Std_ReturnType Dem_ReplIUMPRDenRelease ( Dem_RatioIDType RatioID )	
<b>Service ID[hex]:</b>	0x72	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	Ratioid	Ratio Identifier reporting that specific denominator is released (for physical reasons - e.g. temperature conditions or minimum activity)
<b>Parameters (inout):</b>	None	

<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK report of IUMPR denominator status was successfully reported E_NOK report of IUMPR denominator status was not successfully reported
<b>Description:</b>	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)	

**Table 8.137: Dem\_ReplUMPRDenRelease**

]()

### 8.3.8.7 Dem\_SetPtoStatus

[SWS\_Dem\_00627] [

<b>Service name:</b>	Dem_SetPtoStatus	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetPtoStatus(     boolean PtoStatus )</pre>	
<b>Service ID[hex]:</b>	0x79	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	PtoStatus	sets the status of the PTO (TRUE==active; FALSE==inactive)
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	Returns E_OK when the new PTO-status has been adopted by the Dem; returns E_NOT_OK in all other cases.
<b>Description:</b>	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)	

**Table 8.138: Dem\_SetPtoStatus**

]()

### 8.3.8.8 Dem\_ReadDataOfPID01

[SWS\_Dem\_01167] [

<b>Service name:</b>	Dem_ReadDataOfPID01	
<b>Syntax:</b>	Std_ReturnType Dem_ReadDataOfPID01( uint8* PID01value )	
<b>Service ID[hex]:</b>	0xb3	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	Service to report the value of PID \$01 computed by the Dem. API is needed in OBD relevant ECUs only	

**Table 8.139: Dem\_ReadDataOfPID01**

↴()

### 8.3.8.9 Dem\_GetDataOfPID21

[SWS\_Dem\_01093] ↴

<b>Service name:</b>	Dem_GetDataOfPID21	
<b>Syntax:</b>	Std_ReturnType Dem_GetDataOfPID21( uint8* PID21value )	
<b>Service ID[hex]:</b>	0xb1	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	PID21value	Content of PID \$21 as raw hex value.
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	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/Dem-General/DemGeneralOBD.DemOBDCentralizedPID21Handling)} == true} && {{ecuc(Dem/DemGeneral.DemOBDSupport)} == DEM_OBD_MASTER_ECU}	

**Table 8.140: Dem\_GetDataOfPID21**

↴()

### 8.3.8.10 Dem\_SetDataOfPID21

[SWS\_Dem\_00735] [

<b>Service name:</b>	Dem_SetDataOfPID21	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetDataOfPID21 (     uint8* PID21value )</pre>	
<b>Service ID[hex]:</b>	0xa6	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	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.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	<p>Service to set the value of PID \$21 in the Dem by a software component. API is needed in OBD-relevant ECUs only.</p> <p>API Availability: This API will be available only if  <math>\{(ecuc(Dem/DemGeneral.DemOBDSupport)\} \neq DEM\_OBD\_NO\_OBD\_SUPPORT\}</math></p>	

Table 8.141: Dem\_SetDataOfPID21

]()

### 8.3.8.11 Dem\_SetDataOfPID31

[SWS\_Dem\_00736] [

<b>Service name:</b>	Dem_SetDataOfPID31	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetDataOfPID31 (     uint8* PID31value )</pre>	
<b>Service ID[hex]:</b>	0xa7	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	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.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	

<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	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)	

**Table 8.142: Dem\_SetDataOfPID31**

↴()

### 8.3.8.12 Dem\_SetDataOfPID4D

[SWS\_Dem\_00737] ↴

<b>Service name:</b>	Dem_SetDataOfPID4D	
<b>Syntax:</b>	Std_ReturnType Dem_SetDataOfPID4D ( uint8* PID4Dvalue )	
<b>Service ID[hex]:</b>	0xa8	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	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.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	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)	

**Table 8.143: Dem\_SetDataOfPID4D**

↴()

### 8.3.8.13 Dem\_SetDataOfPID4E

[SWS\_Dem\_00738] ↴

<b>Service name:</b>	Dem_SetDataOfPID4E	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetDataOfPID4E (     uint8* PID4Value )</pre>	
<b>Service ID[hex]:</b>	0xa9	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	PID4Value	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.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	<p>Service to set the value of PID \$4E in the Dem by a software component. API is needed in OBD-relevant ECUs only.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p>	

**Table 8.144: Dem\_SetDataOfPID4E**

↴()

### 8.3.8.14 Dem\_SetPfcCycleQualified

[SWS\_Dem\_00739] ↴

<b>Service name:</b>	Dem_SetPfcCycleQualified	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetPfcCycleQualified(     void )</pre>	
<b>Service ID[hex]:</b>	0xaa	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	<p>Marks the current OBD driving cycle as having met the criteria for the PFC cycle. API is needed in OBD-relevant ECUs only.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p>	

**Table 8.145: Dem\_SetPfcCycleQualified**

]()

### 8.3.8.15 Dem\_SetPfcCycleQualified

[SWS\_Dem\_00740] [

<b>Service name:</b>	Dem_SetPfcCycleQualified	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetPfcCycleQualified(     boolean* isqualified )</pre>	
<b>Service ID[hex]:</b>	0xab	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	isqualified	TRUE: During the current OBD driving cycle the criteria for the PFC cycle have been met. FALSE: During the current OBD driving cycle the criteria for the PFC cycle have not been met.
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	<p>Returns TRUE if the criteria for the PFC cycle have been met during the current OBD driving cycle. API is needed in OBD-relevant ECUs only.</p> <p>API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)</p>	

**Table 8.146: Dem\_GetPfcCycleQualified**

]()

### 8.3.8.16 Dem\_SetClearDTC

[SWS\_Dem\_00741] [

<b>Service name:</b>	Dem_SetClearDTC	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetClearDTC(     uint32 DTC,     Dem_DTCFormatType DTCFormat,     Dem_DTCOriginType DTCOrigin )</pre>	
<b>Service ID[hex]:</b>	0xac	

<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTC  DTCFormat DTCOrigin	Defines the DTC in respective format, that has been cleared from the event memory. Format of the provided DTC value. Event memory (e.g. MIRROR)
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	Return value unused - only for compatibility with according RTE operation.
<b>Description:</b>	API to inform the Dem (of a Dependend / Secondary ECU) about the reception of service \$04 execution by a software component. API is needed in OBD Dependend / Secondary ECUs only.  API Availability: This API will be available only if ({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT)	

**Table 8.147: Dem\_SetClearDTC**

]()

### 8.3.8.17 Dem\_DcmGetDTCSeverityAvailabilityMask

[SWS\_Dem\_01168] [

<b>Service name:</b>	Dem_DcmGetDTCSeverityAvailabilityMask	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_DcmGetDTCSeverityAvailabilityMask(     Dem_DTCSeverityType* DTCSeverityMask )</pre>	
<b>Service ID[hex]:</b>	0xb2	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	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.
<b>Return value:</b>	Std_ReturnType	E_OK: get of DTC severity mask was successful E_NOT_OK: get of DTC severity mask failed
<b>Description:</b>	Gets the DTC Severity availability mask.	

**Table 8.148: Dem\_DcmGetDTCSeverityAvailabilityMask**

]()

### 8.3.8.18 Dem\_GetB1Counter

[SWS\_Dem\_01169] [

<b>Service name:</b>	Dem_GetB1Counter	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_GetB1Counter (     uint16* B1Counter )</pre>	
<b>Service ID[hex]:</b>	0xb4	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	B1Counter	Buffer containing the B1 counter. The buffer is provided by the application with the size of 2 bytes.
<b>Return value:</b>	Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
<b>Description:</b>	Service to report the value of the B1 counter computed by the Dem. API is needed in WWH-OBD relevant ECUs only	

Table 8.149: Dem\_GetB1Counter

]()

### 8.3.8.19 Dem\_SetDTR

[SWS\_Dem\_00765] [

<b>Service name:</b>	Dem_SetDTR	
<b>Syntax:</b>	<pre>Std_ReturnType Dem_SetDTR (     uint16 DTRId,     sint32 TestResult,     sint32 LowerLimit,     sint32 UpperLimit,     Dem_DTRControlType Ctrlval )</pre>	
<b>Service ID[hex]:</b>	0xa2	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	DTRId TestResult LowerLimit UpperLimit Ctrlval	Identification of a DTR element by assigned DTRId. Test result of DTR Lower limit of DTR Upper limit of DTR Control value of the DTR to support its interpretation Dem-internally.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: Report of DTR result successful E_NOT_OK: Report of DTR result failed

<b>Description:</b> 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
--

**Table 8.150: Dem\_SetDTR**

]()

## 8.4 Expected Interfaces

In this chapter, all interfaces required from other modules are listed.

### 8.4.1 Mandatory Interfaces

This chapter defines all interfaces, which are required to fulfill the core functionality of the Dem module.

API function Description

<i>API function</i>	<i>Description</i>

**Table 8.151: Dem Mandatory Interfaces**

### 8.4.2 Optional Interfaces

This chapter defines all interfaces, which are required to fulfill an optional functionality of the Dem module.

[SWS\_Dem\_00255] [

<i>API function</i>	<i>Description</i>
Dcm_DemTriggerOnDTCstatus	Triggers on changes of the UDS DTC status byte. Allows to trigger on ROE Event for subservice OnDTCStatusChanged.
Det_ReportError	Service to report development errors.
Det_ReportError	Service to report development errors.

Dlt_DemTriggerOnEventStatus	This service is provided by the Dem in order to call Dlt upon status changes.
Dlt_DemTriggerOnEventStatus	This service is provided by the Dem in order to call Dlt upon status changes.
FiM_DemInit	This service re-initializes the FIM.
FiM_DemInit	This service re-initializes the FIM.
FiM_DemTriggerOnComponentStatus	Triggers on changes of the component failed status.
FiM_DemTriggerOnEventStatus	This service is provided by the Dem in order to call FiM upon status changes.
FiM_DemTriggerOnEventStatus	This service is provided by the Dem in order to call FiM upon status changes.
J1939Dcm_DemTriggerOnDTCStatus	Trigger for DM01 message that a DTC status change has happened.
NvM_GetErrorStatus	Service to read the block dependent error/status information.
NvM_GetErrorStatus	Service to read the block dependent error/status information.
NvM_ReadBlock	Service to copy the data of the NV block to its corresponding RAM block.
NvM_ReadBlock	Service to copy the data of the NV block to its corresponding RAM block.
NvM_SetRamBlockStatus	Service for setting the RAM block status of an NVRAM block.
NvM_SetRamBlockStatus	Service for setting the RAM block status of an NVRAM block.
NvM_WriteBlock	Service to copy the data of the RAM block to its corresponding NV block.
NvM_WriteBlock	Service to copy the data of the RAM block to its corresponding NV block.

**Table 8.152: Dem Optional Interfaces**

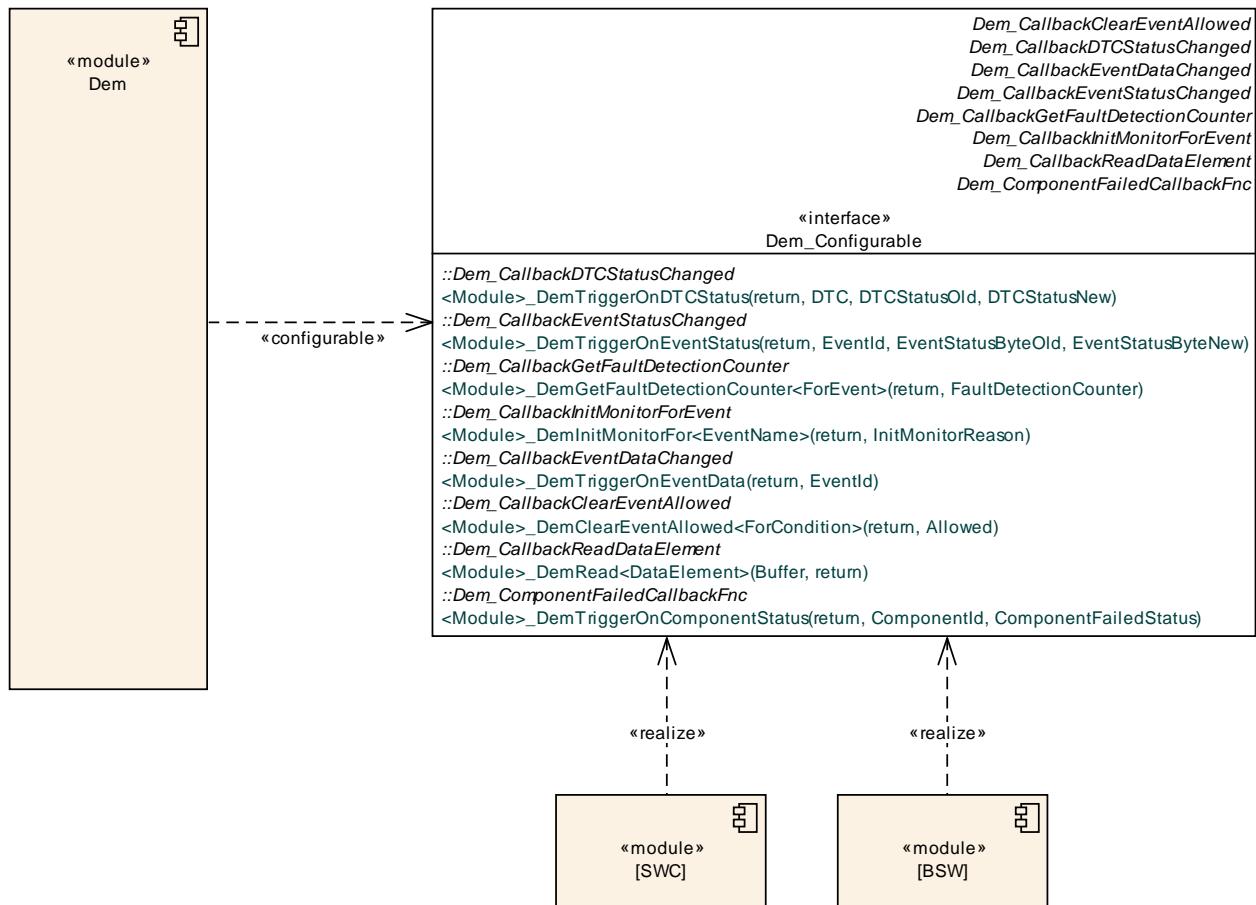
]([SRS\\_BSW\\_00171](#))

Note: Based on implementation strategy FiM\_DemInit can be used (refer also to chapter [7.10.3](#)).

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 [7.10.5](#)).

#### 8.4.3 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**

### 8.4.3.1 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.

#### 8.4.3.1.1 InitMonitorForEvent

##### [SWS\_Dem\_00256] ↗

<b>Service name:</b>	<Module>_DemInitMonitorFor<EventName>	
<b>Syntax:</b>	Std_ReturnType <Module>_DemInitMonitorFor<EventName> ( Dem_InitMonitorReasonType InitMonitorReason )	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	InitMonitorReason	Specific (re-)initialization reason evaluated from the monitor to identify the initialization kind to be performed.

<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	Return value unused - only for compatibility with according RTE operation.
<b>Description:</b>	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.	

**Table 8.153: InitMonitorForEvent**

]()

#### 8.4.3.1.2 DemTriggerOnComponentStatus

[SWS\_Dem\_01116] [

<b>Service name:</b>	<Module>_DemTriggerOnComponentStatus	
<b>Syntax:</b>	Std_ReturnType <Module>_DemTriggerOnComponentStatus ( Dem_ComponentIdType ComponentId, boolean ComponentFailedStatus )	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	ComponentId ComponentFailed Status	Identification of a DemComponent. New FAILED status of the component.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	Return value unused - only for compatibility with according RTE operation.
<b>Description:</b>	Triggers on changes of the DemComponent failed status.	

**Table 8.154: DemTriggerOnComponentStatus**

](SRS\_Diag\_04142)

#### 8.4.3.2 EventStatusChanged

[SWS\_Dem\_00259] [

<b>Service name:</b>	<Module>_DemTriggerOnEventStatus	
<b>Syntax:</b>	Std_ReturnType <Module>_DemTriggerOnEventStatus ( Dem_EventIdType EventId, Dem_UdsStatusByteType EventStatusByteOld, Dem_UdsStatusByteType EventStatusByteNew )	
<b>Sync/Async:</b>	Synchronous	

<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	EventId EventStatusByteOld  EventStatusByte New	Identification of an event by assigned EventId. UDS DTC status byte of event before change (refer to chapter "Status bit support"). UDS DTC status byte of event after change (refer to chapter "Status bit support").
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	Return value unused - only for compatibility with according RTE operation.
<b>Description:</b>	Triggers on changes of the UDS DTC status byte.	

**Table 8.155: DemTriggerOnEventStatus**

↴()

#### 8.4.3.3 DTCStatusChanged

**[SWS\_Dem\_00260]** ↴

<b>Service name:</b>	<Module>_DemTriggerOnDTCStatus	
<b>Syntax:</b>	Std_ReturnType <Module>_DemTriggerOnDTCStatus( uint32 DTC, Dem_UdsStatusByteType DTCStatusOld, Dem_UdsStatusByteType DTCStatusNew )	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTC DTCStatusOld DTCStatusNew	Diagnostic Trouble Code in UDS format. DTC status before change DTC status after change
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	Return value unused - only for compatibility with according RTE operation.
<b>Description:</b>	Triggers on changes of the UDS DTC status byte.	

**Table 8.156: DemTriggerOnDTCStatus**

↴()

#### 8.4.3.3.1 SetClearDTC

**[SWS\_Dem\_00747]** ↴

<b>Service name:</b>	<Module>_SetClearDTC
----------------------	----------------------

<b>Syntax:</b>	Std_ReturnType <Module>_SetClearDTC( uint32 DTC, Dem_DTCFormatType DTCFormat, Dem_DTCOriginType DTCOrigin )	
<b>Service ID[hex]:</b>	0xad	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	DTC  DTCFormat  DTCOrigin	Defines the DTC in respective format, that has been cleared from the event memory.  Format of the provided DTC value.  Event memory (e.g. MIRROR)
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	Return value unused - only for compatibility with according RTE operation.
<b>Description:</b>	API to inform a software component about the reception of service \$04 execution by a Dem of a Primary or Master ECU API is needed in OBD-relevant ECUs only.	

**Table 8.157: SetClearDTC**

↴()

#### 8.4.3.3.2 EventDataChanged

[SWS\_Dem\_00562] ↵

<b>Service name:</b>	<Module>_DemTriggerOnEventData	
<b>Syntax:</b>	Std_ReturnType <Module>_DemTriggerOnEventData( Dem_EventIdType EventId )	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	EventId	Identification of an event by assigned EventId.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	Return value unused - only for compatibility with according RTE operation.
<b>Description:</b>	Triggers on changes of the event related data in the event memory.	

**Table 8.158: DemTriggerOnEventData**

↴()

### 8.4.3.3.3 ClearEventAllowed

[SWS\_Dem\_00563] [

<b>Service name:</b>	<Module>_DemClearEventAllowed<ForCondition>	
<b>Syntax:</b>	Std_ReturnType <Module>_DemClearEventAllowed<ForCondition> ( boolean* Allowed )	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	Allowed	True - clearance of event is allowed False - clearance of event is not allowed
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation failed
<b>Description:</b>	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.	

Table 8.159: DemClearEventAllowed

]()

### 8.4.3.3.4 ReadDataElement

[SWS\_Dem\_00564] [

<b>Service name:</b>	<Module>_DemRead<DataElement>	
<b>Syntax:</b>	Std_ReturnType <Module>_DemRead<DataElement> ( uint8* Buffer )	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	Buffer	Buffer containing the value of the data element
<b>Return value:</b>	Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation failed
<b>Description:</b>	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.	

Table 8.160: DemRead

]()

#### 8.4.3.4 GetFaultDetectionCounter

[SWS\_Dem\_00263] [

<b>Service name:</b>	<Module>_DemGetFaultDetectionCounter<ForEvent>	
<b>Syntax:</b>	Std_ReturnType <Module>_DemGetFaultDetectionCounter< ForEvent>( sint8* FaultDetectionCounter )	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	FaultDetection Counter	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
<b>Return value:</b>	Std_ReturnType	E_OK: request was successful E_NOT_OK: request failed
<b>Description:</b>	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.	

Table 8.161: GetFaultDetection

]()

### 8.5 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.

#### 8.5.1 Dem\_MainFunction

[SWS\_Dem\_00266] [

<b>Service name:</b>	Dem_MainFunction	
<b>Syntax:</b>	void Dem_MainFunction( void )	
<b>Service ID[hex]:</b>	0x55	
<b>Description:</b>	Processes all not event based Dem internal functions.	

Table 8.162: Dem\_MainFunction

]()

**[SWS\_Dem\_00125]** [The function Dem\_MainFunction shall process all not event based Dem module internal functions.]()

For details refer to the chapter 8.5 “Scheduled Functions” in SWS\_BSWGeneral.

## 8.5.2 Runnable Entity MainFunction

```

1 RunnableEntity MainFunction
2     symbol "Dem_MainFunction"
3     canbeInvokedConcurrently = FALSE
4     SSCP = port CBStatusEvt_*, EventStatusChanged
5     SSCP = port GeneralCBStatusEvt, EventStatusChanged
6     SSCP = port CBStatusDTC_*, DTCStatusChanged
7     SSCP = port CBDDataEvt_*, EventDataChanged
8     SSCP = port GeneralCBDDataEvt, EventDataChanged
9     SSCP = port CBReadData_*, ReadData

```

## 8.6 Service Interfaces

### 8.6.1 Sender-Receiver-Interfaces

#### 8.6.1.1 DataServices\_Data

**[SWS\_Dem\_00850]** [

<b>Name</b>	DataServices_{Data}	
<b>Comment</b>	–	
<b>IsService</b>	false	
<b>Variation</b>	({ecuc(Dem/DemGeneral/DemDataElementClass)} instanceof {ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass)}) Data = {ecuc(Dem/DemGeneral/DemDataElementClass.SHORT-NAME)}	
<b>Data Elements</b>	data	
	Type	boolean
	Variation	{ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType)} == BOOLEAN
	data	
	Type	sint8
	Variation	{ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType)} == SINT8
	data	
	Type	sint16
	Variation	{ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass/DemDataElementDataType)} == SINT16
	data	
	Type	sint32

Variation	{ecuc(Dem/DemGeneral/DemDataElementClass/ DemExternalSRDataElementClass/ DemDataElementDataType)} == SINT32
data	
Type	uint32
Variation	{ecuc(Dem/DemGeneral/DemDataElementClass/ DemExternalSRDataElementClass/ DemDataElementDataType)} == UINT32
data	
Type	uint8
Variation	{ecuc(Dem/DemGeneral/DemDataElementClass/ DemExternalSRDataElementClass/ DemDataElementDataType)} == UINT8
data	
Type	uint16
Variation	{ecuc(Dem/DemGeneral/DemDataElementClass/ DemExternalSRDataElementClass/ DemDataElementDataType)} == UINT16

**Table 8.163: Service Interface DataServices\_{Data}**

」(SRS\_Diag\_04030, SRS\_Rte\_00182)

## 8.6.2 Client-Server-Interfaces

### 8.6.2.1 AgingCycle

**[SWS\_Dem\_00602]** [The Dem Service Component shall provide the interface AgingCycle as defined below (to provide the capability to set an aging cycle state).]

One port of this interface type is provided per aging cycle by the Dem Service Component. It has AgingCycleId as a port-defined argument.

<b>Name</b>	AgingCycle	
<b>Comment</b>	—	
<b>IsService</b>	true	
<b>Variation</b>	—	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.164: Service Interface AgingCycle**

## Operations

SetAgingCycleState		
<b>Comments</b>	—	
<b>Variation</b>	—	
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	—

**Table 8.165: Operation SetAgingCycleState**

]([SRS\\_Diag\\_04076](#), [SRS\\_Diag\\_04061](#))

### 8.6.2.2 CallbackClearEventAllowed

**[SWS\_Dem\_00620]** [The Dem Service Component shall provide the interface CallbackClearEventAllowed as defined below (to get the permission before clearing a specific event from the SW-C), if configured.

For each event, there can be one port of this interface type.

<b>Name</b>	CallbackClearEventAllowed	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.166: Service Interface CallbackClearEventAllowed**

### Operations

ClearEventAllowed		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	Allowed	
	Comment	True - clearance of event is allowed False - clearance of event is not allowed
	Type	boolean
	Variation	–
	Direction	OUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.167: Operation ClearEventAllowed**

]([SRS\\_Diag\\_04117](#))

### 8.6.2.3 CallbackEventStatusChange

**[SWS\_Dem\_00615]** [The Dem Service Component shall provide the interface CallbackEventStatusChange as defined below (to trigger SW-Cs on event status byte changes), if configured.

For each event, there can be several ports of this interface type.

<b>Name</b>	CallbackEventStatusChange
<b>Comment</b>	–
<b>IsService</b>	true
<b>Variation</b>	–

**Table 8.168: Service Interface CallbackEventStatusChange**

## Operations

EventStatusChanged																					
<b>Comments</b>	–																				
<b>Variation</b>	–																				
<b>Parameters</b>	<table border="1"> <tr> <td>EventStatusByteOld</td> <td></td> </tr> <tr> <td>Comment</td><td>–</td></tr> <tr> <td>Type</td><td>Dem_UdsStatusByteType</td></tr> <tr> <td>Variation</td><td>–</td></tr> <tr> <td>Direction</td><td>IN</td></tr> <tr> <td>EventStatusByteNew</td><td></td></tr> <tr> <td>Comment</td><td>–</td></tr> <tr> <td>Type</td><td>Dem_UdsStatusByteType</td></tr> <tr> <td>Variation</td><td>–</td></tr> <tr> <td>Direction</td><td>IN</td></tr> </table>	EventStatusByteOld		Comment	–	Type	Dem_UdsStatusByteType	Variation	–	Direction	IN	EventStatusByteNew		Comment	–	Type	Dem_UdsStatusByteType	Variation	–	Direction	IN
EventStatusByteOld																					
Comment	–																				
Type	Dem_UdsStatusByteType																				
Variation	–																				
Direction	IN																				
EventStatusByteNew																					
Comment	–																				
Type	Dem_UdsStatusByteType																				
Variation	–																				
Direction	IN																				

**Table 8.169: Operation EventStatusChanged**

]()

### 8.6.2.4 CallbackEventDataChanged

**[SWS\_Dem\_00618]** [The Dem Service Component shall provide the interface CallbackEventDataChanged as defined below (to trigger SW-Cs on event related data changes), if configured.

For each event, there can be one port of this interface type.

<b>Name</b>	CallbackEventDataChanged
<b>Comment</b>	–
<b>IsService</b>	true
<b>Variation</b>	–

**Table 8.170: Service Interface CallbackEventDataChanged**

## Operations

EventDataChanged	
<b>Comments</b>	–
<b>Variation</b>	–

**Table 8.171: Operation EventDataChanged**

]()

### 8.6.2.5 CallbackGetFaultDetectCounter

**[SWS\_Dem\_00622]** [The Dem Service Component shall provide the interface CallbackGetFaultDetectionCounter as defined below (to get the monitor-internal fault detection counter value of a specific event from the SW-C), if configured.

<b>Name</b>	CallbackGetFaultDetectCounter	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.172: Service Interface CallbackGetFaultDetectCounter**

### Operations

GetFaultDetectionCounter		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	FaultDetectionCounter	
	Comment	Value of FaultDetectionCounter
	Type	sint8
	Variation	–
	Direction	OUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.173: Operation GetFaultDetectionCounter**

]()

### 8.6.2.6 CallbackInitMonitorForEvent

**[SWS\_Dem\_00613]** [The Dem Service Component shall provide the interface CallbackInitMonitorForEvent as defined below (to trigger an event-specific initialization of the monitor part of the SW-C), if configured.

For each event, there can be one port of this interface type.

<b>Name</b>	CallbackInitMonitorForEvent	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.174: Service Interface CallbackInitMonitorForEvent**

## Operations

InitMonitorForEvent	
<b>Comments</b>	–
<b>Variation</b>	–
<b>Parameters</b>	InitMonitorReason
Comment	–
Type	Dem_InitMonitorReasonType
Variation	–
Direction	IN
<b>Possible Errors</b>	E_OK Operation successful
	E_NOT_OK –

**Table 8.175: Operation InitMonitorForEvent**

]()

### 8.6.2.7 CallbackDTCStatusChange

**[SWS\_Dem\_00617]** [The Dem Service Component shall provide the interface CallbackDTCStatusChange as defined below (to trigger SW-Cs on DTC status byte changes), if configured via [DemCallbackDTCStatusChanged](#), [DemCallbackOBDDTCStatusChanged](#), and / or [DemCallbackJ1939DTCStatusChanged](#).

There can be several ports of this interface type, provided globally by the Dem Service Component.

<b>Name</b>	CallbackDTCStatusChange	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.176: Service Interface CallbackDTCStatusChange**

## Operations

DTCStatusChanged	
<b>Comments</b>	–
<b>Variation</b>	–
<b>Parameters</b>	DTC
Comment	–
Type	uint32
Variation	–

	Direction	IN
	DTCStatusOld	
	Comment	—
	Type	Dem_UdsStatusByteType
	Variation	—
	Direction	IN
	DTCStatusNew	
	Comment	—
	Type	Dem_UdsStatusByteType
	Variation	—
	Direction	IN
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	—

**Table 8.177: Operation DTCStatusChanged**

]()

### 8.6.2.8 GeneralCallbackEventDataChanged

[SWS\_Dem\_00619] [

<b>Name</b>	GeneralCallbackEventDataChanged
<b>Comment</b>	—
<b>IsService</b>	true
<b>Variation</b>	—

**Table 8.178: Service Interface GeneralCallbackEventDataChanged**

### Operations

	EventDataChanged
<b>Comments</b>	—
<b>Variation</b>	—
<b>Parameters</b>	EventId
	Comment
	Type
	Variation
	Direction

**Table 8.179: Operation EventDataChanged**

]()

### 8.6.2.9 GeneralCallbackEventStatusChange

[SWS\_Dem\_00616] [

<b>Name</b>	GeneralCallbackEventStatusChange
<b>Comment</b>	–
<b>IsService</b>	true
<b>Variation</b>	–

**Table 8.180: Service Interface GeneralCallbackEventStatusChange**

#### Operations

EventStatusChanged																															
<b>Comments</b>	–																														
<b>Variation</b>	–																														
<b>Parameters</b>	<table border="1"> <tr> <td>EventId</td> <td>–</td> </tr> <tr> <td>Comment</td><td>–</td></tr> <tr> <td>Type</td><td>Dem_EventIdType</td></tr> <tr> <td>Variation</td><td>–</td></tr> <tr> <td>Direction</td><td>IN</td></tr> <tr> <td>EventStatusByteOld</td><td>–</td></tr> <tr> <td>Comment</td><td>–</td></tr> <tr> <td>Type</td><td>Dem_UdsStatusByteType</td></tr> <tr> <td>Variation</td><td>–</td></tr> <tr> <td>Direction</td><td>IN</td></tr> <tr> <td>EventStatusByteNew</td><td>–</td></tr> <tr> <td>Comment</td><td>–</td></tr> <tr> <td>Type</td><td>Dem_UdsStatusByteType</td></tr> <tr> <td>Variation</td><td>–</td></tr> <tr> <td>Direction</td><td>IN</td></tr> </table>	EventId	–	Comment	–	Type	Dem_EventIdType	Variation	–	Direction	IN	EventStatusByteOld	–	Comment	–	Type	Dem_UdsStatusByteType	Variation	–	Direction	IN	EventStatusByteNew	–	Comment	–	Type	Dem_UdsStatusByteType	Variation	–	Direction	IN
EventId	–																														
Comment	–																														
Type	Dem_EventIdType																														
Variation	–																														
Direction	IN																														
EventStatusByteOld	–																														
Comment	–																														
Type	Dem_UdsStatusByteType																														
Variation	–																														
Direction	IN																														
EventStatusByteNew	–																														
Comment	–																														
Type	Dem_UdsStatusByteType																														
Variation	–																														
Direction	IN																														

**Table 8.181: Operation EventStatusChanged**

] (SRS\_Diag\_04061)

### 8.6.2.10 CallbackComponentStatusChanged

[SWS\_Dem\_01195] [

<b>Name</b>	CallbackComponentStatusChanged
<b>Comment</b>	–
<b>IsService</b>	true
<b>Variation</b>	–

**Table 8.182: Service Interface CallbackComponentStatusChanged**

#### Operations

ComponentStatusChanged	
<b>Comments</b>	–
<b>Variation</b>	–
<b>Parameters</b>	ComponentFailedStatus
Comment	–
Type	boolean
Variation	–
Direction	IN

**Table 8.183: Operation ComponentStatusChanged**

]([SRS\\_Diag\\_04142](#))

### 8.6.2.11 Service Interface CddIf

**[SWS\_Dem\_00666]** [The Dem Service Component shall provide the interface CddIf as defined below (to provide the operations only related to complex device drivers). One port of this interface type is provided globally by the Dem Service Component.

Note: This port (Cdd) can only be connected to the respective port of the CDD module (refer to note below [[SWS\\_Dem\\_00659](#)]).

<b>Name</b>	CddIf	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	DEM_CLEAR_WRONG_DTC
	2	DEM_CLEAR_WRONG_DTCORIGIN
	3	DEM_CLEAR_FAILED
	4	DEM_CLEAR_PENDING
	5	DEM_CLEAR_BUSY
	6	DEM_CLEAR_MEMORY_ERROR

**Table 8.184: Service Interface CddIf**

### Operations

ClearDTC	
<b>Comments</b>	–
<b>Variation</b>	–
<b>Parameters</b>	DTC
Comment	–
Type	uint32
Variation	–
Direction	IN
DTCTFormat	
Comment	–
Type	Dem_DTCFormatType

	Variation	–
	Direction	IN
	DTCOrigin	
	Comment	–
	Type	Dem_DTCOriginType
	Variation	–
	Direction	IN
<b>Possible Errors</b>	E_OK	Operation successful
	DEM_CLEAR_WRONG_DTC	–
	DEM_CLEAR_WRONG_DTCORIGIN	–
	DEM_CLEAR_FAILED	–
	DEM_CLEAR_PENDING	–
	DEM_CLEAR_BUSY	–
	DEM_CLEAR_MEMORY_ERROR	–

**Table 8.185: Operation ClearDTC**

]()

### 8.6.2.12 Service Interface DTCSuppression

**[SWS\_Dem\_00608]** [The Dem Service Component shall provide the interface DTCSuppression as defined below (to provide the capability to control the suppression of DTCs).]

One port of this interface type is provided globally by the Dem Service Component.

<b>Name</b>	DTCSuppression	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	({ecuc(Dem/DemGeneral/DemSuppressionSupport)} == DEM_DTC_SUPPRESSION)	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.186: Service Interface DTCSuppression**

### Operations

SetDTCSuppression	
<b>Comments</b>	–
<b>Variation</b>	–
<b>Parameters</b>	DTC
	Comment
	Type
	Variation

	Direction	IN
	DTCFormat	
	Comment	—
	Type	Dem_DTCFormatType
	Variation	—
	Direction	IN
	SuppressionStatus	
	Comment	—
	Type	boolean
	Variation	—
	Direction	IN
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	—

**Table 8.187: Operation SetDTCSuppression**

] ([SRS\\_Diag\\_04030](#))

### 8.6.2.13 DataServices\_{Data}

**[SWS\_Dem\_00621]** [The Dem Service Component shall provide the interface DataServices\_{<Data>} as defined below (to get 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 7.48), if configured.

For each data element, one port of this interface type is provided by the SW-Cs.

<b>Name</b>	DataServices_{Data}	
<b>Comment</b>	—	
<b>IsService</b>	true	
<b>Variation</b>	(({ecuc(Dem/DemGeneral/DemDataElementClass)}) instanceof {ecuc(Dem/DemGeneral/DemDataElementClass/ DemExternalCSDDataElementClass)}))&& (({ecuc(Dem/DemGeneral/ DemDataElementClass/DemExternalCSDDataElementClass/ DemDataElementUsePort)} == true)) Data = {ecuc(Dem/ DemGeneral/DemDataElementClass.SHORT-NAME)}	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.188: Service Interface DataServices\_{Data}**

### Operations

ReadData	
----------	--

<b>Comments</b>	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)	
<b>Variation</b>	–	
<b>Parameters</b>	Data	
	Comment	–
	Type	DataArrayType_{Data}
	Variation	Data = {ecuc(Dem/DemGeneral/DemDataElementClass.SHORT-NAME)}
	Direction	OUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.189: Operation ReadData**

]([SRS\\_Diag\\_04061](#))

Note: The Dem- and Dcm-interfaces of this type are compatible, to be connected to the same provide-port of the application.

Note: The Dem Service Component supports synchronous client/server interfaces only (due to the used mechanisms for the [event memory](#)) and is compatible with the Dcm interface DataServices\_<Data> with the setting for USE\_DATA\_SYNCH\_CLIENT\_SERVER. All further operations contained in the Dcm interface CSDataServices\_<Data> like WriteData, ReadDataLength (relates to data elements with a variable length), ConditionCheckRead, etc. are provided by the SW-C and used by the Dcm, but are not required/considered by the Dem module.

Note: The Dem Service Component is compatible with the Dcm interface DataServices\_<Data> with the setting for USE\_DATA\_SENDER\_RECEIVER.

### 8.6.2.14 DcmIf

**[SWS\_Dem\_00609]** [The Dem Service Component shall provide the interface DcmIf as defined below (to provide the operations only related to the Dcm).]

One port of this interface type is provided globally by the Dem Service Component.

<b>Name</b>	DcmIf	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	DEM_CONTROL_DTC_SETTING_N_OK
	1	DEM_CLEAR_WRONG_DTC
	2	DEM_CONTROL_DTC_WRONG_DTCGROUP

	2	DEM_CLEAR_WRONG_DTCORIGIN
	3	DEM_CLEAR_FAILED
	4	DEM_CLEAR_PENDING
	5	DEM_CLEAR_BUSY
	6	DEM_CLEAR_MEMORY_ERROR

**Table 8.190: Service Interface DcmIf**

## Operations

DcmClearDTC		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	DTC	
Comment	–	
Type	uint32	
Variation	–	
Direction	IN	
DTCFormat		
Comment	–	
Type	Dem_DTCFormatType	
Variation	–	
Direction	IN	
DTCOrigin		
Comment	–	
Type	Dem_DTCOriginType	
Variation	–	
Direction	IN	
<b>Possible Errors</b>	E_OK	Operation successful
	DEM_CLEAR_WRONG_DTC	–
	DEM_CLEAR_WRONG_DTCORIGIN	–
	DEM_CLEAR_FAILED	–
	DEM_CLEAR_PENDING	–
	DEM_CLEAR_BUSY	–
	DEM_CLEAR_MEMORY_ERROR	–

**Table 8.191: Operation DcmClearDTC**

DcmEnableDTCSSetting		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	DTCSGroup	
Comment	–	
Type	uint32	
Variation	–	
Direction	IN	
DTCKind		
Comment	–	

	Type	Dem_DTCKindType
	Variation	—
	Direction	IN
<b>Possible Errors</b>	E_OK	Operation successful
	DEM_CONTROL_DTC_SETTING_N_OK	—
	DEM_CONTROL_DTC_WRONG_DTCGROUP	—

**Table 8.192: Operation DcmEnableDTCSetting**

]()

Note: This port (Dcm) can only be connected to the respective port of the Dcm module (refer to note below [[SWS\\_Dem\\_00009](#)] and [[SWS\\_Dem\\_00035](#)]).

### 8.6.2.15 DTRCentralReport

**[SWS\_Dem\_00769]** [The Dem Service Component shall provide the interface DTR-CentralReport as defined), if OBD support is configured.

<b>Name</b>	DTRCentralReport	
<b>Comment</b>	—	
<b>IsService</b>	true	
<b>Variation</b>	—	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.193: Service Interface DTRCentralReport**

### Operations

SetDTR		
<b>Comments</b>	—	
<b>Variation</b>	—	
<b>Parameters</b>	TestResult	
	Comment	—
	Type	sint32
	Variation	—
	Direction	IN
	LowerLimit	
	Comment	—
	Type	sint32
	Variation	—
	Direction	IN
	UpperLimit	
	Comment	—
	Type	sint32
	Variation	—

	Direction	IN
	Ctrlval	
	Comment	–
	Type	Dem_DTRControlType
	Variation	–
	Direction	IN
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.194: Operation SetDTR**

]()

### 8.6.2.16 EnableCondition

**[SWS\_Dem\_00604]** [The Dem Service Component shall provide the interface EnableCondition as defined below (to provide the capability to set an enable condition), if at least one enable condition is configured.

One port of this interface type is provided per enable condition by the Dem Service Component. It has EnableConditionId as a port-defined argument.

<b>Name</b>	EnableCondition	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.195: Service Interface EnableCondition**

### Operations

SetEnableCondition		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	ConditionFulfilled	
	Comment	–
	Type	boolean
	Variation	–
	Direction	IN
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.196: Operation SetEnableCondition**

]()

### 8.6.2.17 DiagnosticMonitor

**[SWS\_Dem\_00598]** [The Dem Service Component shall provide the interface DiagnosticMonitor as defined below (to 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 EventId as a port-defined argument.

<b>Name</b>	DiagnosticMonitor	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.197: Service Interface DiagnosticMonitor**

#### Operations

ClearPrestoredFreezeFrame		
<b>Comments</b>	–	
<b>Variation</b>	{ecuc(Dem/DemGeneral/DemMaxNumberPrestoredFF)} > 0	
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.198: Operation ClearPrestoredFreezeFrame**

PrestoreFreezeFrame		
<b>Comments</b>	–	
<b>Variation</b>	{ecuc(Dem/DemGeneral/DemMaxNumberPrestoredFF)} > 0	
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.199: Operation PrestoreFreezeFrame**

ResetEventDebounceStatus		
<b>Comments</b>	–	
<b>Variation</b>	({ecuc(Dem/DemGeneral/DemDebounceCounterBasedSupport)} == true)    ({ecuc(Dem/DemGeneral/DemDebounceTimeBasedSupport)} == true)	
<b>Parameters</b>	DebounceResetStatus	
	Comment	–
	Type	Dem_DebounceResetStatusType
	Variation	–
	Direction	IN
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.200: Operation ResetEventDebounceStatus**

ResetEventStatus		
Comments	–	
Variation	–	
Possible Errors	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.201: Operation ResetEventStatus**

SetEventDisabled		
Comments	–	
Variation	{ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT	
Possible Errors	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.202: Operation SetEventDisabled**

SetEventStatus		
Comments	–	
Variation	–	
Parameters	EventStatus	
	Comment	–
	Type	Dem_EventStatusType
	Variation	–
	Direction	IN
Possible Errors	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.203: Operation SetEventStatus**

]()

Note: Each port of the DiagnosticMonitor interface is only connected to one monitor port.

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.

### 8.6.2.18 EventStatus

**[SWS\_Dem\_00838]** [The Dem Service Component shall provide the interface EventStatus as defined below (to provide 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.

Name	EventStatus
------	-------------

<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.204: Service Interface EventStatus**

## Operations

SetWIRStatus		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	WIRStatus	
	Comment	–
	Type	boolean
	Variation	–
	Direction	IN
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.205: Operation SetWIRStatus**

]()

Note: Each port of the EventStatus interface is only connected to one port of the failsafe SW-C.

### 8.6.2.19 EventAvailable

[SWS\_Dem\_01193] ↴

<b>Name</b>	EventAvailable	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	{ecuc(Dem/DemGeneral/DemAvailabilitySupport)} == DEM_EVENT_AVAILABILITY	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.206: Service Interface EventAvailable**

## Operations

SetEventAvailable		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	AvailableStatus	

	Comment	–
	Type	boolean
	Variation	–
	Direction	IN
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.207: Operation SetEventAvailable**

]()

### 8.6.2.20 DiagnosticInfo

**[SWS\_Dem\_00599]** [The Dem Service Component shall provide the interface DiagnosticInfo as defined below (to provide 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.

<b>Name</b>	DiagnosticInfo	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK
	2	DEM_E_NO_DTC_AVAILABLE
	3	DEM_E_NO_FDC_AVAILABLE
	21	DEM_E_WRONG_BUFFERSIZE
	48	DEM_E_NODATAAVAILABLE
	49	DEM_E_WRONG_RECORDNUMBER
	50	DEM_E_WRONG_DIDNUMBER

**Table 8.208: Service Interface DiagnosticInfo**

### Operations

GetDTCOfEvent		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	DTCTFormat	
	Comment	–
	Type	Dem_DTCFormatType
	Variation	–
	Direction	IN
	DTCTOfEvent	
	Comment	–
	Type	uint32
	Variation	–
	Direction	OUT

<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	—
	DEM_E_NO_DTC_AVAILABLE	there is no DTC configured in the requested format

**Table 8.209: Operation GetDTCOfEvent**

GetDebouncingOfEvent		
<b>Comments</b>	—	
<b>Variation</b>	$((\{ecuc(Dem/DemConfigSet/DemEventParameter/DemEventClass/\\nDemDebounceAlgorithmClass)\} instanceof \{ecuc(Dem/\\nDemConfigSet/DemEventParameter/DemEventClass/\\nDemDebounceAlgorithmClass/DemDebounceCounterBased)\})$ $\ (\{ecuc(Dem/DemConfigSet/DemEventParameter/DemEventClass/\\nDemDebounceAlgorithmClass)\} instanceof \{ecuc(Dem/\\nDemConfigSet/DemEventParameter/DemEventClass/\\nDemDebounceAlgorithmClass/DemDebounceTimeBase)\}))$	
<b>Parameters</b>	DebouncingState	
	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)
	Type	Dem_DebouncingStateType
	Variation	—
	Direction	OUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	—

**Table 8.210: Operation GetDebouncingOfEvent**

GetEventExtendedDataRecord		
<b>Comments</b>	—	
<b>Variation</b>	—	
<b>Parameters</b>	RecordNumber	
	Comment	—
	Type	uint8
	Variation	—
	Direction	IN
	DestBuffer	
	Comment	—
	Type	Dem_MaxDataValueType
	Variation	—
	Direction	OUT
<b>Possible Errors</b>	E_OK	Operation successful

E_NOT_OK	Operation could not be performed
DEM_E_NODATAAVAILABLE	The requested event data is not currently stored (but the request was valid)
DEM_E_WRONG_RECORDNUMBER	The requested record number is not supported by the event

**Table 8.211: Operation GetEventExtendedDataRecord**

GetEventExtendedDataRecordEx		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	RecordNumber	
Comment	–	
Type	uint8	
Variation	–	
Direction	IN	
DestBuffer		
Comment	–	
Type	Dem_MaxDataValueType	
Variation	–	
Direction	OUT	
Buflen		
Comment	–	
Type	uint16	
Variation	–	
Direction	INOUT	
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–
	DEM_E_WRONG_BUFFERSIZE	The provided buffer size is too small
	DEM_E_NODATAAVAILABLE	The requested event data is not currently stored (but the request was valid)
	DEM_E_WRONG_RECORDNUMBER	The requested record number is not supported by the event

**Table 8.212: Operation GetEventExtendedDataRecordEx**

GetEventFailed		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	EventFailed	
Comment	TRUE - Last Failed FALSE - not Last Failed	
Type	boolean	
Variation	–	
Direction	OUT	
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.213: Operation GetEventFailed**

GetEventFreezeFrameData		
Comments	–	
Variation	–	
Parameters	RecordNumber	–
	Comment	–
	Type	uint8
	Variation	–
	Direction	IN
	ReportTotalRecord	
	Comment	–
	Type	boolean
	Variation	–
	Direction	IN
	DataId	–
	Comment	–
	Type	uint16
	Variation	–
	Direction	IN
	DestBuffer	–
	Comment	–
	Type	Dem_MaxDataValueType
	Variation	–
	Direction	OUT
Possible Errors	E_OK	Operation successful
	E_NOT_OK	Operation could not be performed
	DEM_E_NODATAAVAILABLE	The requested event data is not currently stored (but the request was valid)
	DEM_E_WRONG_RECORDNUMBER	The requested record number is not supported by the event
	DEM_E_WRONG_DIDNUMBER	The requested DID is not supported by the freeze frame

**Table 8.214: Operation GetEventFreezeFrameData**

GetEventFreezeFrameDataEx		
Comments	–	
Variation	–	
Parameters	RecordNumber	–
	Comment	–
	Type	uint8
	Variation	–
	Direction	IN
	DataId	–
	Comment	–
	Type	uint16
	Variation	–
	Direction	IN
	DestBuffer	–
	Comment	–
	Type	Dem_MaxDataValueType

	Variation	–
	Direction	OUT
	BufSize	
	Comment	–
	Type	uint16
	Variation	–
	Direction	INOUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–
	DEM_E_WRONG_BUFFERSIZE	The provided buffer size is too small
	DEM_E_NODATAAVAILABLE	The requested event data is not currently stored (but the request was valid)
	DEM_E_WRONG_RECORDNUMBER	The requested record number is not supported by the event
	DEM_E_WRONG_DIDNUMBER	The requested DID is not supported by the freeze frame

**Table 8.215: Operation GetEventFreezeFrameDataEx**

GetEventStatus	
<b>Comments</b>	–
<b>Variation</b>	–
<b>Parameters</b>	EventStatusByte
	Comment
	Type
	Variation
	Direction
<b>Possible Errors</b>	E_OK
	E_NOT_OK

**Table 8.216: Operation GetEventStatus**

GetEventTested	
<b>Comments</b>	–
<b>Variation</b>	–
<b>Parameters</b>	EventTested
	Comment
	Type
	Variation
	Direction
<b>Possible Errors</b>	E_OK
	E_NOT_OK

**Table 8.217: Operation GetEventTested**

GetFaultDetectionCounter	
<b>Comments</b>	–
<b>Variation</b>	–

<b>Parameters</b>	FaultDetectionCounter	
	Comment	–
	Type	sint8
	Variation	–
	Direction	OUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–
	DEM_E_NO_FDC_AVAILABLE	there is no fault detection counter available for the requested event

**Table 8.218: Operation GetFaultDetectionCounter**

]()

Note: These DiagnosticInfo ports are also available for each BSW event (so that also SW-Cs can access on).

### 8.6.2.21 GeneralDiagnosticInfo

#### [SWS\_Dem\_00600] [

<b>Name</b>	GeneralDiagnosticInfo	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK
	2	DEM_E_NO_DTC_AVAILABLE
	3	DEM_E_NO_FDC_AVAILABLE
	21	DEM_E_WRONG_BUFFERSIZE
	48	DEM_E_NODATAAVAILABLE
	49	DEM_E_WRONG_RECORDNUMBER
	50	DEM_E_WRONG_DIDNUMBER

**Table 8.219: Service Interface GeneralDiagnosticInfo**

### Operations

GetDTCOfEvent	
<b>Comments</b>	–
<b>Variation</b>	–
<b>Parameters</b>	EventId
	Comment
	Type
	Variation
	Direction
	DTCFormat
	Comment
	Type

	Variation	—
	Direction	IN
	DTCOfEvent	
	Comment	—
	Type	uint32
	Variation	—
	Direction	OUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	—
	DEM_E_NO_DTC_AVAILABLE	there is no DTC configured in the requested format

**Table 8.220: Operation GetDTCOfEvent**

GetDebouncingOfEvent		
Comments	—	
<b>Variation</b>		((ecuc(Dem/DemConfigSet/DemEventParameter/DemEventClass/ DemDebounceAlgorithmClass)) instanceof {ecuc(Dem/ DemConfigSet/DemEventParameter/DemEventClass/ DemDebounceAlgorithmClass/DemDebounceCounterBased)})   ((ecuc(Dem/DemConfigSet/DemEventParameter/DemEventClass/ DemDebounceAlgorithmClass)) instanceof {ecuc(Dem/ DemConfigSet/DemEventParameter/DemEventClass/ DemDebounceAlgorithmClass/DemDebounceTimeBase)}))
<b>Parameters</b>	EventId	
	Comment	—
	Type	Dem_EventIdType
	Variation	—
	Direction	IN
	DebouncingState	
	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)
	Type	Dem_DebouncingStateType
	Variation	—
	Direction	OUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	—

**Table 8.221: Operation GetDebouncingOfEvent**

GetEventExtendedDataRecord		
Comments	—	

<b>Variation</b>	–		
<b>Parameters</b>	EventId		
	Comment	–	
	Type	Dem_EventIdType	
	Variation	–	
	Direction	IN	
	RecordNumber		
	Comment	–	
	Type	uint8	
	Variation	–	
	Direction	IN	
<b>Possible Errors</b>	DestBuffer		
	Comment	–	
	Type	Dem_MaxDataValueType	
	Variation	–	
	Direction	OUT	

**Table 8.222: Operation GetEventExtendedDataRecord**

GetEventExtendedDataRecordEx			
<b>Comments</b>	–		
<b>Variation</b>	–		
<b>Parameters</b>	RecordNumber		
	Comment	–	
	Type	uint8	
	Variation	–	
	Direction	IN	
	DestBuffer		
	Comment	–	
	Type	Dem_MaxDataValueType	
	Variation	–	
	Direction	OUT	
<b>Possible Errors</b>	Buflen		
	Comment	–	
	Type	uint16	
	Variation	–	
	Direction	INOUT	
<b>Possible Errors</b>	E_OK	Operation successful	
	E_NOT_OK	–	
	DEM_E_WRONG_BUFFER_SIZE	The provided buffer size is too small	
	DEM_E_NODATAAVAILABLE	The requested event data is not currently stored (but the request was valid)	
	DEM_E_WRONG_RECORDNUMBER	The requested record number is not supported by the event	

**Table 8.223: Operation GetEventExtendedDataRecordEx**

GetEventFailed	
<b>Comments</b>	–
<b>Variation</b>	–
<b>Parameters</b>	EventId
	Comment
	Type
	Variation
	Direction
	EventFailed
	Comment
	Type
	Variation
	Direction
<b>Possible Errors</b>	E_OK
	E_NOT_OK

**Table 8.224: Operation GetEventFailed**

GetEventFreezeFrameData	
<b>Comments</b>	–
<b>Variation</b>	–
<b>Parameters</b>	EventId
	Comment
	Type
	Variation
	Direction
	RecordNumber
	Comment
	Type
	Variation
	Direction
<b>ReportTotalRecord</b>	ReportTotalRecord
	Comment
	Type
	Variation
	Direction
	DataId
	Comment
	Type
	Variation
	Direction
<b>DestBuffer</b>	DestBuffer
	Comment
	Type
	Variation
<b>Direction</b>	IN
	OUT

<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	Operation could not be performed
	DEM_E_NODATAAVAILABLE	The requested event data is not currently stored (but the request was valid)
	DEM_E_WRONG_RECORDNUMBER	The requested record number is not supported by the event
	DEM_E_WRONG_DIDNUMBER	The requested DID is not supported by the freeze frame

**Table 8.225: Operation GetEventFreezeFrameData**

GetEventFreezeFrameDataEx		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	RecordNumber	
	Comment	–
	Type	uint8
	Variation	–
	Direction	IN
	DataId	
	Comment	–
	Type	uint16
	Variation	–
	Direction	IN
	DestBuffer	
	Comment	–
	Type	Dem_MaxDataValueType
	Variation	–
	Direction	OUT
	BufSize	
	Comment	–
	Type	uint16
	Variation	–
	Direction	INOUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–
	DEM_E_WRONG_BUFFERSIZE	The provided buffer size is too small
	DEM_E_NODATAAVAILABLE	The requested event data is not currently stored (but the request was valid)
	DEM_E_WRONG_RECORDNUMBER	The requested record number is not supported by the event
	DEM_E_WRONG_DIDNUMBER	The requested DID is not supported by the freeze frame

**Table 8.226: Operation GetEventFreezeFrameDataEx**

GetEventStatus		
<b>Comments</b>	–	
<b>Variation</b>	–	

<b>Parameters</b>	EventId	
	Comment	–
	Type	Dem_EventIdType
	Variation	–
	Direction	IN
	EventStatusByte	
	Comment	–
	Type	Dem_UdsStatusByteType
	Variation	–
	Direction	OUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.227: Operation GetEventStatus**

GetEventTested		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	EventId	
	Comment	–
	Type	Dem_EventIdType
	Variation	–
	Direction	IN
	EventTested	
	Comment	–
	Type	boolean
	Variation	–
	Direction	OUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.228: Operation GetEventTested**

GetFaultDetectionCounter		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	EventId	
	Comment	–
	Type	Dem_EventIdType
	Variation	–
	Direction	IN
	FaultDetectionCounter	
	Comment	–
	Type	sint8
	Variation	–
	Direction	OUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–
	DEM_E_NO_FDC_AVAILABLE	there is no fault detection counter available for the requested event

---

**Table 8.229: Operation GetFaultDetectionCounter**

]()

### 8.6.2.22 IndicatorStatus

**[SWS\_Dem\_00606]** [The Dem Service Component shall provide the interface IndicatorStatus as defined below, if at least one indicator is configured.

One port of this interface type is provided per indicator by the Dem Service Component. It has IndicatorId as a port-defined argument.

<b>Name</b>	IndicatorStatus	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.230: Service Interface IndicatorStatus**

### Operations

GetIndicatorStatus		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	IndicatorStatus	
	Comment	–
	Type	Dem_IndicatorStatusType
	Variation	–
	Direction	OUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.231: Operation GetIndicatorStatus**

]([SRS\\_Diag\\_04128](#))

### 8.6.2.23 IUMPRDenominator

**[SWS\_Dem\_00611]** [The Dem Service Component shall provide the interface IUMPRDenominator as defined below (to provide the capability to define the number of times the vehicle operation has been fulfilled), if OBD support is configured.

One port of this interface type is provided per ratio Id by the Dem Service Component. It has RatioId as a port-defined argument.

<b>Name</b>	IUMPRDenominator	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.232: Service Interface IUMPRDenominator**

## Operations

ReplIUMPRDenLock		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.233: Operation ReplIUMPRDenLock**

ReplIUMPRDenRelease		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.234: Operation ReplIUMPRDenRelease**

]()

### 8.6.2.24 IUMPRDenominatorCondition

**[SWS\_Dem\_00742]** [The Dem Service Component shall provide the interface IUMPRDenominatorCondition as defined below (to broadcast the status information of the General Denominator and additional denominator conditions among all OBD relevant ECUs), if OBD support is configured.

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.

<b>Name</b>	IUMPRDenominatorCondition	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.235: Service Interface IUMPRDenominatorCondition**

## Operations

GetIUMPRDenCondition		
Comments	–	
Variation	–	
Parameters	ConditionStatus	
	Comment	–
	Type	Dem_lumprDenomCondStatusType
	Variation	–
	Direction	OUT
Possible Errors	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.236: Operation GetIUMPRDenCondition**

SetIUMPRDenCondition		
Comments	–	
Variation	–	
Parameters	ConditionStatus	
	Comment	–
	Type	Dem_lumprDenomCondStatusType
	Variation	–
	Direction	IN
Possible Errors	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.237: Operation SetIUMPRDenCondition**

]()

### 8.6.2.25 IUMPRNumerator

**[SWS\_Dem\_00610]** [The Dem Service Component shall provide the interface IUMPR-Numerator as defined below (to provide the capability to define the number of times a fault could have been found), if OBD support is configured.

One port of this interface type is provided per ratio Id by the Dem Service Component. It has RatioId as a port-defined argument.

Name	IUMPRNumerator	
Comment	–	
IsService	true	
Variation	–	
Possible Errors	0	E_OK
	1	E_NOT_OK

**Table 8.238: Service Interface IUMPRNumerator**

## Operations

ReIUMPRFaultDetect		
Comments	–	
Variation	–	
Possible Errors	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.239: Operation ReIUMPRFaultDetect**

]()

### 8.6.2.26 OperationCycle

**[SWS\_Dem\_00601]** [The Dem Service Component shall provide the interface OperationCycle as defined below (to provide 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 OperationCycId as a port-defined argument.

Name	OperationCycle	
Comment	–	
IsService	true	
Variation	–	
Possible Errors	0	E_OK
	1	E_NOT_OK

**Table 8.240: Service Interface OperationCycle**

## Operations

GetOperationCycleState		
Comments	–	
Variation	–	
Parameters	CycleState	–
	Comment	–
	Type	Dem_OperationCycleStateType
	Variation	–
	Direction	OUT
Possible Errors	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.241: Operation GetOperationCycleState**

SetOperationCycleState		
Comments	–	
Variation	–	

<b>Parameters</b>	CycleState	
	Comment	–
	Type	Dem_OperationCycleStateType
	Variation	–
	Direction	IN
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.242: Operation SetOperationCycleState**

] (*SRS\_Diag\_04076*)

### 8.6.2.27 EvMemOverflowIndication

**[SWS\_Dem\_00607]** [The Dem Service Component shall provide the interface EvMemOverflowIndication as defined below (to provide the status of the *event memory*), if the respective *event memory* is configured.

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.

<b>Name</b>	EvMemOverflowIndication	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.243: Service Interface EvMemOverflowIndication**

### Operations

GetEventMemoryOverflow		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	OverflowIndication	
	Comment	–
	Type	boolean
	Variation	–
	Direction	OUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.244: Operation GetEventMemoryOverflow**

GetNumberOfEventMemoryEntries		
<b>Comments</b>	–	
<b>Variation</b>	–	

<b>Parameters</b>	GetNumberOfEventMemoryEntries	
Comment	–	
Type	uint8	
Variation	–	
Direction	OUT	
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.245: Operation GetNumberOfEventMemoryEntries**

] (*SRS\_Diag\_04105*)

### 8.6.2.28 PfcCycleQualified

**[SWS\_Dem\_00743]** [The Dem Service Component shall provide the interface PfcCycleQualified as defined below, if the interface is used by a SW-C.

<b>Name</b>	PfcCycleQualified	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.246: Service Interface PfcCycleQualified**

### Operations

GetPfcCycleQualified		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	isQualified	
Comment	–	
Type	boolean	
Variation	–	
Direction	OUT	
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.247: Operation GetPfcCycleQualified**

SetPfcCycleQualified		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.248: Operation SetPfcCycleQualified**

]()

### 8.6.2.29 PowerTakeOff

**[SWS\_Dem\_00612]** [The Dem Service Component shall provide the interface PowerTakeOff as defined below (to provide the capability to set the PTO status), if OBD support is configured.

One port of this interface type is provided by the Dem Service Component.

<b>Name</b>	PowerTakeOff	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.249: Service Interface PowerTakeOff**

#### Operations

SetPtoStatus		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	PtoStatus	
	Comment	–
	Type	boolean
	Variation	–
	Direction	IN
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.250: Operation SetPtoStatus**

]()

### 8.6.2.30 SetClearDTC

**[SWS\_Dem\_00744]** [The Dem Service Component shall provide the interface SetClearDTC as defined below, if the interface is used by a SW-C.

<b>Name</b>	SetClearDTC	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.251: Service Interface SetClearDTC**

## Operations

SetClearDTC	
<b>Comments</b>	–
<b>Variation</b>	–
<b>Parameters</b>	DTC
	Comment
	Type
	Variation
	Direction
	DTCFormat
	Comment
	Type
	Variation
	Direction
<b>Possible Errors</b>	DTCOrigin
	Comment
	Type
	Variation
	Direction
E_OK	Operation successful
E_NOT_OK	–

**Table 8.252: Operation SetClearDTC**

」([SRS\\_Diag\\_04030](#))

### 8.6.2.31 GetDataOfPID21

**[SWS\_Dem\_01092]** [The Dem Service Component shall provide the interface GetDataOfPID21 as defined below, if the interface is used by a SW-C.

<b>Name</b>	GetDataOfPID21	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.253: Service Interface GetDataOfPID21**

## Operations

GetDataOfPID21	
<b>Comments</b>	–

<b>Variation</b>	–	
<b>Parameters</b>	PID21value	
	Comment	–
	Type	Dem_PID21valueType
	Variation	–
	Direction	OUT
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.254: Operation GetDataOfPID21**

]([SRS\\_Diag\\_04061](#), [SRS\\_Diag\\_04001](#))

### 8.6.2.32 SetDataOfPID21

**[SWS\_Dem\_00745]** [The Dem Service Component shall provide the interface SetDataOfPID21 as defined below, if the interface is used by a SW-C.

<b>Name</b>	SetDataOfPID21	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.255: Service Interface SetDataOfPID21**

### Operations

SetDataOfPID21		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	PID21value	
	Comment	–
	Type	Dem_PID21valueType
	Variation	–
	Direction	IN
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.256: Operation SetDataOfPID21**

]([SRS\\_Diag\\_04061](#), [SRS\\_Diag\\_04001](#))

### 8.6.2.33 SetDataOfPID31

**[SWS\_Dem\_00746]** [The Dem Service Component shall provide the interface SetDataOfPID31 as defined below, if the interface is used by a SW-C.

<b>Name</b>	SetDataOfPID31	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.257: Service Interface SetDataOfPID31**

### Operations

SetDataOfPID31		
<b>Comments</b>	–	
<b>Variation</b>	–	
<b>Parameters</b>	PID31value	
	Comment	–
	Type	Dem_PID31valueType
	Variation	–
	Direction	IN
<b>Possible Errors</b>	E_OK	Operation successful
	E_NOT_OK	–

**Table 8.258: Operation SetDataOfPID31**

] ([SRS\\_Diag\\_04082](#))

### 8.6.2.34 StorageCondition

**[SWS\_Dem\_00605]** [The Dem Service Component shall provide the interface StorageCondition as defined below (to provide the capability to set an enable 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.

<b>Name</b>	StorageCondition	
<b>Comment</b>	–	
<b>IsService</b>	true	
<b>Variation</b>	–	
<b>Possible Errors</b>	0	E_OK
	1	E_NOT_OK

**Table 8.259: Service Interface StorageCondition**

## Operations

SetStorageCondition		
Comments	–	
Variation	–	
Parameters	ConditionFulfilled	
	Comment	This parameter specifies whether the enable condition assigned to the EnableConditionID is fulfilled (TRUE) or not fulfilled (FALSE).
	Type	boolean
	Variation	–
	Direction	IN
Possible Errors	E_OK	Operation successful
	E_NOT_OK	–

Table 8.260: Operation SetStorageCondition

] (SRS\_Diag\_04095)

### 8.6.3 Implementation Data Types

[SWS\_Dem\_01072] [

Name	Dem_MaxDataValueType		
Kind	Array	Element type	uint8
Size	size of largest Extended data class / Freeze frame record Elements		
Description	–		
Variation	–		

Table 8.261: Implementation Data Type Dem\_MaxDataValueType

] ()

[SWS\_Dem\_01073] [

Name	Dem_PID21valueType		
Kind	Array	Element type	uint8
Size	2 Elements		
Description	–		
Variation	–		

Table 8.262: Implementation Data Type Dem\_PID21valueType

] ()

[SWS\_Dem\_01074] [

Name	Dem_PID31valueType		
Kind	Array	Element type	uint8

<b>Size</b>	2 Elements		
<b>Description</b>	–		
<b>Variation</b>	–		

**Table 8.263: Implementation Data Type Dem\_PID31valueType**

]()

[SWS\_Dem\_01192] [

<b>Name</b>	DataArrayType_{Data}		
<b>Kind</b>	Array	<b>Element type</b>	uint8
<b>Size</b>	{ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementDataSize)} Elements		
<b>Description</b>	–		
<b>Variation</b>	(({ecuc(Dem/DemGeneral/DemDataElementClass)} instanceof {ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass)}) &&{ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass/DemDataElementUsePort)} == true)) Data = {ecuc(Dem/DemGeneral/DemDataElementClass.SHORT-NAME)}		

**Table 8.264: Implementation Data Type DataArrayType\_{Data}**

]()

## 8.6.4 Ports

### 8.6.4.1 Dem\_AgingCycle

[SWS\_Dem\_01001] [

<b>Name</b>	AgingCycle_{Name}		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	AgingCycle
<b>Description</b>	–		
<b>Variation</b>	Name = {ecuc(Dem/DemGeneral/DemOperationCycle.SHORT-NAME)}		

**Table 8.265: Port AgingCycle\_{Name}**

]()

### 8.6.4.2 Dem\_CBDataEvt

[SWS\_Dem\_01003] [

<b>Name</b>	CBDataEvt_{Name}		
-------------	------------------	--	--

<b>Kind</b>	RequiredPort	<b>Interface</b>	CallbackEventDataChanged
<b>Description</b>	–		
<b>Variation</b>	(({ecuc(Dem/DemConfigSet/DemEventParameter/ DemCallbackEventDataChanged)} != NULL) &&({ecuc(Dem/ DemConfigSet/DemEventParameter/ DemCallbackEventDataChanged/ DemCallbackEventDataChangedFnc)} == NULL)) Name = {ecuc(Dem/DemConfigSet/DemEventParameter.SHORT-NAME)}		

**Table 8.266: Port CBDataEvt\_{Name}**

↴()

#### 8.6.4.3 Dem\_CBFaultDetectCtr

**[SWS\_Dem\_01004]** ↴

<b>Name</b>	CBFaultDetectCtr_{Name}		
<b>Kind</b>	RequiredPort	<b>Interface</b>	CallbackGetFaultDetect Counter
<b>Description</b>	–		
<b>Variation</b>	(({ecuc(Dem/DemConfigSet/DemEventParameter/DemEventClass/ DemDebounceAlgorithmClass/DemDebounceMonitorInternal/ DemCallbackGetFDC/DemCallbackGetFDCFnc)} == NULL) Name = {ecuc(Dem/DemConfigSet/DemEventParameter/DemEventClass/ DemDebounceAlgorithmClass/DemDebounceMonitorInternal/ DemCallbackGetFDC.SHORT-NAME)})		

**Table 8.267: Port CBFaultDetectCtr\_{Name}**

↴()

#### 8.6.4.4 Dem\_CBIInitEvt

**[SWS\_Dem\_01005]** ↴

<b>Name</b>	CBIInitEvt_{Name}		
<b>Kind</b>	RequiredPort	<b>Interface</b>	CallbackInitMonitorForEvent
<b>Description</b>	–		
<b>Variation</b>	(({ecuc(Dem/DemConfigSet/DemEventParameter/ DemCallbackInitMForE)} != NULL) &&({ecuc(Dem/DemConfigSet/ DemEventParameter/DemCallbackInitMForE/ DemCallbackInitMForEFnc)} == NULL)) Name = {ecuc(Dem/ DemConfigSet/DemEventParameter.SHORT-NAME)}		

**Table 8.268: Port CBIInitEvt\_{Name}**

]()

#### 8.6.4.5 Dem\_CBStatusDTC

[SWS\_Dem\_01007] [

<b>Name</b>	CBStatusDTC_{Name}		
<b>Kind</b>	RequiredPort	<b>Interface</b>	CallbackDTCStatusChange
<b>Description</b>	–		
<b>Variation</b>	Name = {ecuc(Dem/DemGeneral/DemCallbackDTCStatusChanged. SHORT-NAME)}		

**Table 8.269: Port CBStatusDTC\_{Name}**

]()

#### 8.6.4.6 Dem\_CBStatusEvt

[SWS\_Dem\_01008] [

<b>Name</b>	CBStatusEvt_{EventName}_{CallbackName}		
<b>Kind</b>	RequiredPort	<b>Interface</b>	CallbackEventStatusChange
<b>Description</b>	–		
<b>Variation</b>	EventName = {ecuc(Dem/DemConfigSet/DemEventParameter. SHORT-NAME)} CallbackName = {ecuc(Dem/DemConfigSet/ DemEventParameter/DemCallbackEventStatusChanged. SHORT-NAME)}		

**Table 8.270: Port CBStatusEvt\_{EventName}\_{CallbackName}**

]()

#### 8.6.4.7 Dem\_CBStatusComp

[SWS\_Dem\_01197] [

<b>Name</b>	CBStatusComp_{ComponentName}		
<b>Kind</b>	RequiredPort	<b>Interface</b>	CallbackComponentStatus Changed
<b>Description</b>	–		
<b>Variation</b>	ComponentName = {ecuc(Dem/DemConfigSet/DemComponent. SHORT-NAME)}		

**Table 8.271: Port CBStatusComp\_{ComponentName}**

」(SRS\_Diag\_04142)

#### 8.6.4.8 Dem\_Cdd

[SWS\_Dem\_01009] 「

<b>Name</b>	Cdd		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	CddIf
<b>Description</b>	–		
<b>Variation</b>	–		

Table 8.272: Port Cdd

」()

#### 8.6.4.9 Dem\_ControlDTCSuppression

[SWS\_Dem\_01010] 「

<b>Name</b>	ControlDTCSuppression		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	DTCSuppression
<b>Description</b>	–		
<b>Variation</b>	{ecuc(Dem/DemGeneral/DemSuppressionSupport)} == DEM_DTC_SUPPRESSION		

Table 8.273: Port ControlDTCSuppression

」()

#### 8.6.4.10 Dem\_ControlEventAvailable

[SWS\_Dem\_01011] 「

<b>Name</b>	ControlEventAvailable		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	EventAvailable
<b>Description</b>	–		
<b>Variation</b>	{ecuc(Dem/DemGeneral/DemAvailabilitySupport)} == DEM_EVENT_AVAILABILITY		

Table 8.274: Port ControlEventAvailable

」()

#### 8.6.4.11 DataServices

[SWS\_Dem\_01012] ↗

<b>Name</b>	DataServices_{Data}		
<b>Kind</b>	RequiredPort	<b>Interface</b>	DataServices_{Data}
<b>Description</b>	–		
<b>Variation</b>	$((\{ecuc(Dem/DemGeneral/DemDataElementClass)\} \text{ instanceof } \{ecuc(Dem/DemGeneral/DemDataElementClass/\\DemExternalCSDDataElementClass)\}) \& (\{ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalCSDDataElementClass/\\DemDataElementUsePort)\} == \text{TRUE})) \mid  (\{ecuc(Dem/DemGeneral/DemDataElementClass)\} \text{ instanceof } \{ecuc(Dem/DemGeneral/DemDataElementClass/DemExternalSRDataElementClass)\}) \text{ Data} = \{ecuc(Dem/DemGeneral/DemDataElementClass.SHORT-NAME)\}$		

Table 8.275: Port DataServices\_{Data}

↳

#### 8.6.4.12 Dem\_Dcm

[SWS\_Dem\_01040] ↗

<b>Name</b>	Dcm		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	DcmIf
<b>Description</b>	–		
<b>Variation</b>	–		

Table 8.276: Port Dcm

↳

#### 8.6.4.13 Dem\_DTR

[SWS\_Dem\_01039] ↗

<b>Name</b>	DTR_{Name}		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	DTRCentralReport
<b>Description</b>	–		
<b>Variation</b>	$(\{ecuc(Dem/DemGeneral.DemOBDSupport)\} != \text{DEM\_OBD\_NO\_OBD\_SUPPORT}) \text{ Name} = \{ecuc(Dem/DemConfigSet/DemDtrs/DemDtr.SHORT-NAME)\}$		

Table 8.277: Port DTR\_{Name}

↳

#### 8.6.4.14 Dem\_EnableCond

[SWS\_Dem\_01038] ↴

<b>Name</b>	EnableCond_{Name}		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	EnableCondition
<b>Description</b>	–		
<b>Variation</b>	Name = {ecuc(Dem/DemGeneral/DemEnableCondition. SHORT-NAME)}		

Table 8.278: Port EnableCond\_{Name}

↳()

#### 8.6.4.15 Dem\_Event

[SWS\_Dem\_01037] ↴

<b>Name</b>	Event_{Name}		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	DiagnosticMonitor
<b>Description</b>	–		
<b>Variation</b>	Name = {ecuc(Dem/DemConfigSet/DemEventParameter. SHORT-NAME)}		

Table 8.279: Port Event\_{Name}

↳()

#### 8.6.4.16 Dem\_EventStatus

[SWS\_Dem\_01036] ↴

<b>Name</b>	EventStatus_{Name}		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	EventStatus
<b>Description</b>	–		
<b>Variation</b>	Name = {ecuc(Dem/DemConfigSet/DemEventParameter. SHORT-NAME)}		

Table 8.280: Port EventStatus\_{Name}

↳()

#### 8.6.4.17 Dem\_EvtInfo

[SWS\_Dem\_01034] ↴

<b>Name</b>	EventInfo_{Name}		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	DiagnosticInfo
<b>Description</b>	–		
<b>Variation</b>	Name = {ecuc(Dem/DemConfigSet/DemEventParameter. SHORT-NAME)}		

**Table 8.281: Port EventInfo\_{Name}**

]()

#### 8.6.4.18 Dem\_GeneralCBDataEvt

[SWS\_Dem\_01041] [

<b>Name</b>	GeneralCBDataEvt		
<b>Kind</b>	RequiredPort	<b>Interface</b>	GeneralCallbackEventData Changed
<b>Description</b>	–		
<b>Variation</b>	({ecuc(Dem/DemGeneral/DemGeneralInterfaceSupport)} == True)		

**Table 8.282: Port GeneralCBDataEvt**

]()

#### 8.6.4.19 Dem\_GeneralCBStatusEvt

[SWS\_Dem\_01032] [

<b>Name</b>	GeneralCBStatusEvt		
<b>Kind</b>	RequiredPort	<b>Interface</b>	GeneralCallbackEventStatus Change
<b>Description</b>	–		
<b>Variation</b>	({ecuc(Dem/DemGeneral/DemGeneralInterfaceSupport)} == True)		

**Table 8.283: Port GeneralCBStatusEvt**

]()

#### 8.6.4.20 Dem\_GeneralEvtInfo

[SWS\_Dem\_01031] [

<b>Name</b>	GeneralEvtInfo		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	GeneralDiagnosticInfo

<b>Description</b>	–
<b>Variation</b>	({ecuc(Dem/DemGeneral/DemGeneralInterfaceSupport)} == True)

**Table 8.284: Port GeneralEvtInfo**

↴()

#### 8.6.4.21 Dem\_IndStatus

[SWS\_Dem\_01030] ↴

<b>Name</b>	IndStatus_{Name}		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	IndicatorStatus
<b>Description</b>	–		
<b>Variation</b>	Name = {ecuc(Dem/DemGeneral/DemIndicator.SHORT-NAME)}		

**Table 8.285: Port IndStatus\_{Name}**

↴()

#### 8.6.4.22 Dem\_IUMPRDenominator

[SWS\_Dem\_01029] ↴

<b>Name</b>	IUMPRDenominator_{Name}		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	IUMPRDenominator
<b>Description</b>	–		
<b>Variation</b>	({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT) Name = {ecuc(Dem/DemGeneral/DemRatio/DemIUMPRDenGroup.SHORT-NAME)}		

**Table 8.286: Port IUMPRDenominator\_{Name}**

↴()

#### 8.6.4.23 Dem\_IUMPRDenominatorCondition

[SWS\_Dem\_01028] ↴

<b>Name</b>	IUMPRDenominatorCondition_{Name}		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	IUMPRDenominatorCondition
<b>Description</b>	–		
<b>Variation</b>	({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT) Name = {ecuc(Dem/DemGeneral/DemRatio/DemIUMPRDenGroup.SHORT-NAME)}		

**Table 8.287: Port IUMPRDenominatorCondition\_{Name}**

]()

#### 8.6.4.24 Dem\_IUMPRNumerator

[SWS\_Dem\_01027] [

<b>Name</b>	IUMPRNumerator_{Name}		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	IUMPRNumerator
<b>Description</b>	–		
<b>Variation</b>	({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT) Name = {ecuc(Dem/DemGeneral/DemRatioId.SHORT-NAME)}		

**Table 8.288: Port IUMPRNumerator\_{Name}**

]()

#### 8.6.4.25 Dem\_OpCycle

[SWS\_Dem\_01026] [

<b>Name</b>	OpCycle_{Name}		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	OperationCycle
<b>Description</b>	–		
<b>Variation</b>	Name = {ecuc(Dem/DemGeneral/DemOperationCycle.SHORT-NAME)}		

**Table 8.289: Port OpCycle\_{Name}**

]()

#### 8.6.4.26 Dem\_OverflowIndMirrorMemory

[SWS\_Dem\_01025] [

<b>Name</b>	OverflowIndMirrorMemory		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	EvMemOverflowIndication
<b>Description</b>	–		
<b>Variation</b>	({ecuc(Dem/DemGeneral/DemMaxNumberEventEntryMirror)} > 0)		

**Table 8.290: Port OverflowIndMirrorMemory**

]()

#### 8.6.4.27 Dem\_OverflowIndPermanentMemory

[SWS\_Dem\_01024] [

<b>Name</b>	OverflowIndPermanentMemory		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	EvMemOverflowIndication
<b>Description</b>	–		
<b>Variation</b>	({ecuc(Dem/DemGeneral/DemMaxNumberEventEntryPermanent)} > 0)		

**Table 8.291: Port OverflowIndPermanentMemory**

]()

#### 8.6.4.28 Dem\_OverflowIndPrimaryMemory

[SWS\_Dem\_01023] [

<b>Name</b>	OverflowIndPrimaryMemory		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	EvMemOverflowIndication
<b>Description</b>	–		
<b>Variation</b>	–		

**Table 8.292: Port OverflowIndPrimaryMemory**

]()

#### 8.6.4.29 Dem\_OverflowIndUserDefinedMemory

[SWS\_Dem\_01022] [

<b>Name</b>	OverflowIndUserDefinedMemory_{Memory}		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	EvMemOverflowIndication
<b>Description</b>	–		
<b>Variation</b>	({ecuc(Dem/DemGeneral/DemUserDefinedMemory)} != NULL) Memory = {ecuc(Dem/DemGeneral/DemUserDefinedMemory. SHORT-NAME)}		

**Table 8.293: Port OverflowIndUserDefinedMemory\_{Memory}**

]()

#### 8.6.4.30 Dem\_PfcCycleQualified

[SWS\_Dem\_01021] ↗

<b>Name</b>	PfcCycleQualified		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	PfcCycleQualified
<b>Description</b>	–		
<b>Variation</b>	{ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT &&{ecuc(Dem/DemGeneral/DemMaxNumberEventEntryPermanent)} > 0)		

Table 8.294: Port PfcCycleQualified

↳()

#### 8.6.4.31 Dem\_PowerTakeOffStatus

[SWS\_Dem\_01020] ↗

<b>Name</b>	PowerTakeOffStatus		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	PowerTakeOff
<b>Description</b>	–		
<b>Variation</b>	{ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT		

Table 8.295: Port PowerTakeOffStatus

↳()

#### 8.6.4.32 Dem\_SetClearDTC\_dependend

[SWS\_Dem\_01019] ↗

<b>Name</b>	SetClearDTC_dependend		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	SetClearDTC
<b>Description</b>	Port is used in dependend / secondary ECUs		
<b>Variation</b>	{ecuc(Dem/DemGeneral.DemOBDSupport)} == DEM_OBD_DEP_SEC_ECU		

Table 8.296: Port SetClearDTC\_dependend

↳()

#### 8.6.4.33 Dem\_SetClearDTC\_master

[SWS\_Dem\_01018] [

<b>Name</b>	SetClearDTC_master		
<b>Kind</b>	RequiredPort	<b>Interface</b>	SetClearDTC
<b>Description</b>	port is used in Master ECU		
<b>Variation</b>	{ecuc(Dem/DemGeneral.DemOBDSupport)} == DEM_OBD_MASTER_ECU		

Table 8.297: Port SetClearDTC\_master

]()

#### 8.6.4.34 Dem\_GetDataOfPID21

[SWS\_Dem\_01094] [

<b>Name</b>	GetDataOfPID21		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	GetDataOfPID21
<b>Description</b>	–		
<b>Variation</b>	({ecuc(Dem/DemGeneral/DemGeneralOBD. DemOBDCentralizedPID21Handling)} == true) && ({ecuc(Dem/DemGeneral.DemOBDSupport)} == DEM_OBD_MASTER_ECU)		

Table 8.298: Port GetDataOfPID21

]()

#### 8.6.4.35 Dem\_SetDataOfPID21

[SWS\_Dem\_01017] [

<b>Name</b>	SetDataOfPID21		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	SetDataOfPID21
<b>Description</b>	–		
<b>Variation</b>	({ecuc(Dem/DemGeneral/DemGeneralOBD. DemOBDCentralizedPID21Handling)} == true) && ({ecuc(Dem/DemGeneral.DemOBDSupport)} == DEM_OBD_PRIMARY_ECU)		

Table 8.299: Port SetDataOfPID21

]()

#### 8.6.4.36 Dem\_SetDataOfPID31

[SWS\_Dem\_01016] ↗

<b>Name</b>	SetDataOfPID31		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	SetDataOfPID31
<b>Description</b>	–		
<b>Variation</b>	({ecuc(Dem/DemGeneral.DemOBDSupport)} != DEM_OBD_NO_OBD_SUPPORT) && ({ecuc(Dem/DemGeneral/DemGeneralOBD.DemOBDCentralizedPID31Handling)} == true)		

Table 8.300: Port SetDataOfPID31

↳()

#### 8.6.4.37 Dem\_StorageCond

[SWS\_Dem\_01015] ↗

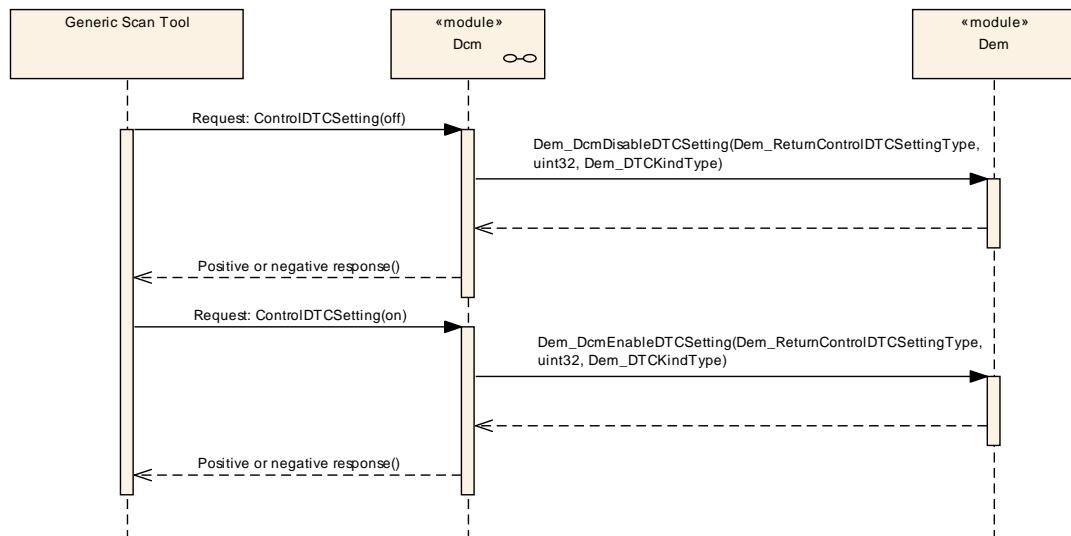
<b>Name</b>	StorageCond_{Name}		
<b>Kind</b>	ProvidedPort	<b>Interface</b>	StorageCondition
<b>Description</b>	–		
<b>Variation</b>	Name = {ecuc(Dem/DemGeneral/DemStorageCondition.SHORT-NAME)}		

Table 8.301: Port StorageCond\_{Name}

↳()

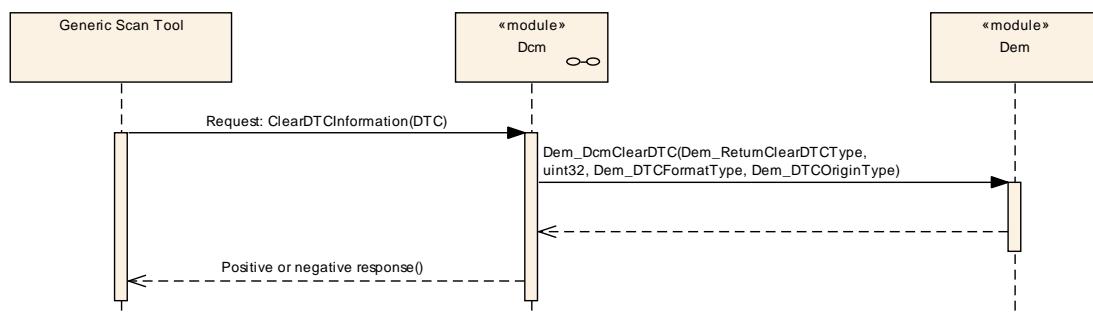
## 9 Sequence Diagrams

### 9.1 ControlDTCSetting

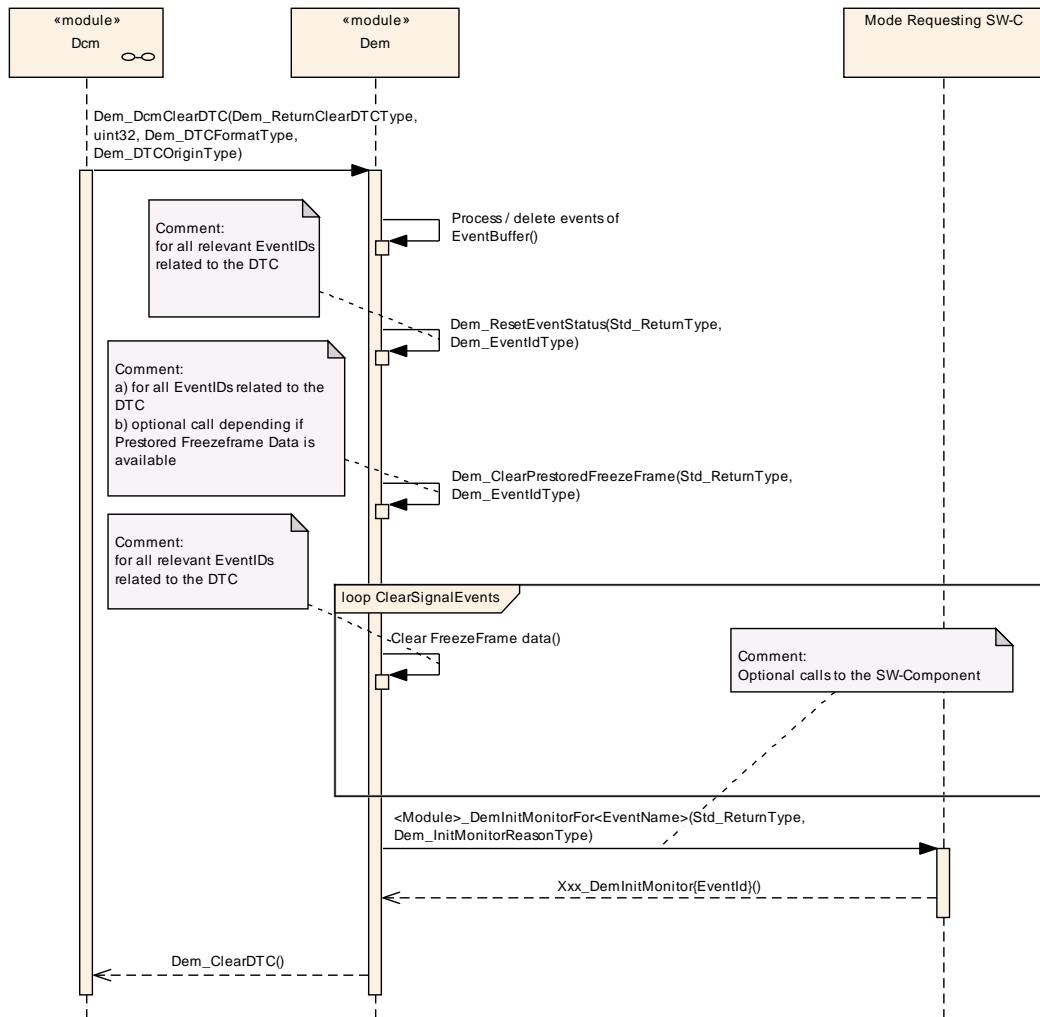


**Figure 9.1: Sequence diagram of ControlDTCSetting**

### 9.2 Dem\_DcmClearDTC

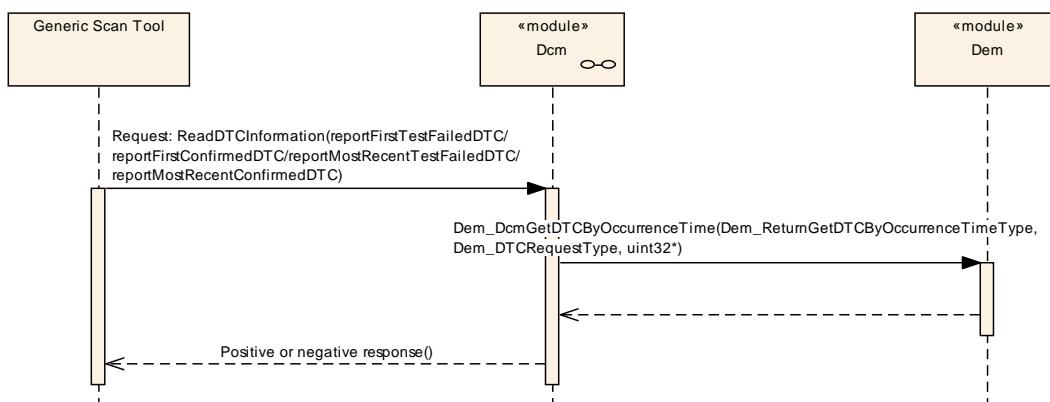


**Figure 9.2: Sequence diagram of Dem\_DcmClearDTC**



**Figure 9.3: Sequence diagram of Dem\_DcmClearDTC clearing a single DTC Dem-internally**

### 9.3 Dem\_DcmGetDTCByOccurrenceTime



**Figure 9.4: Sequence diagram of Dem\_DcmGetDTCByOccurrenceTime**

## 9.4 Dem\_DcmGetExtendedDataRecordByDTC

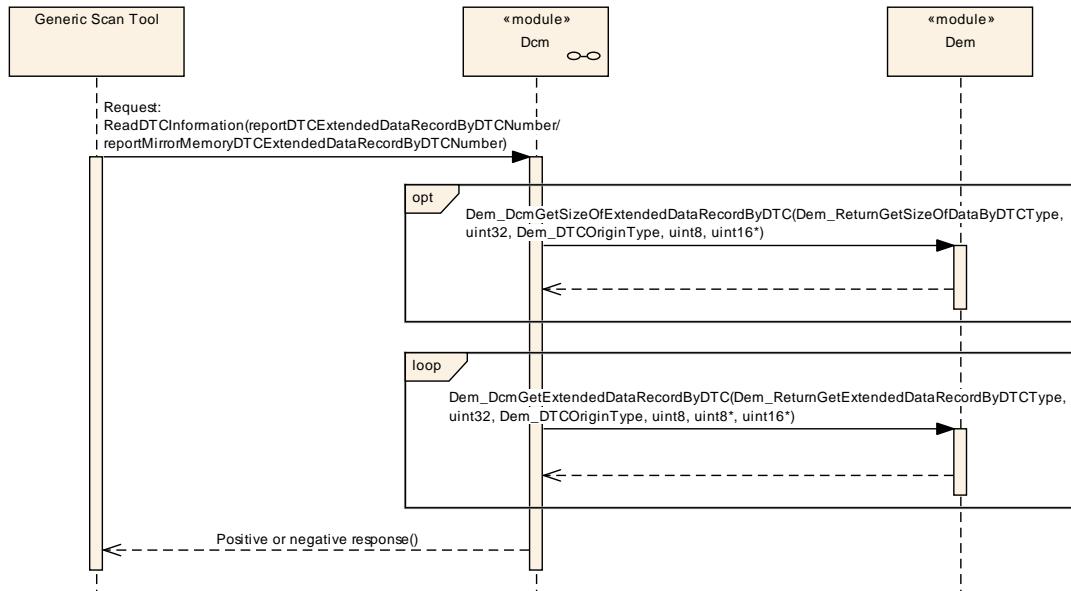


Figure 9.5: Sequence diagram of Dem\_DcmGetExtendedDataRecordByDTC

## 9.5 Dem\_DcmGetStatusOfDTC

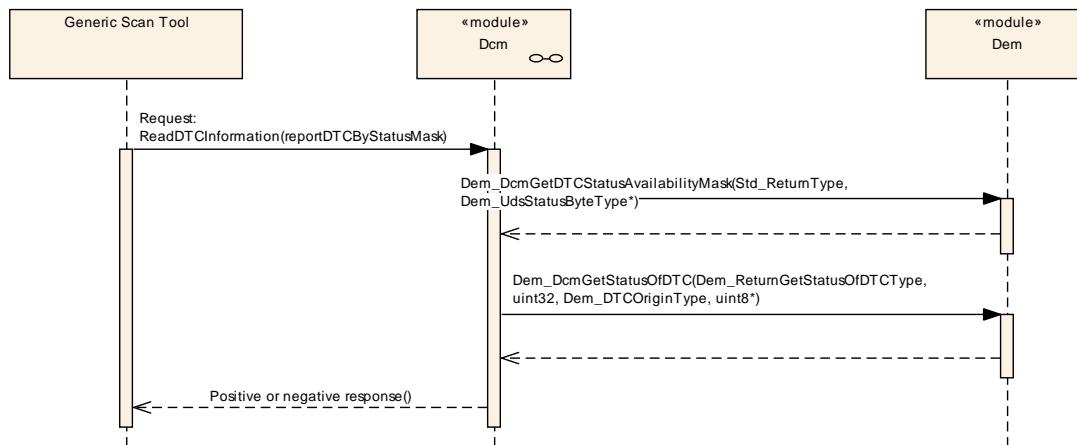
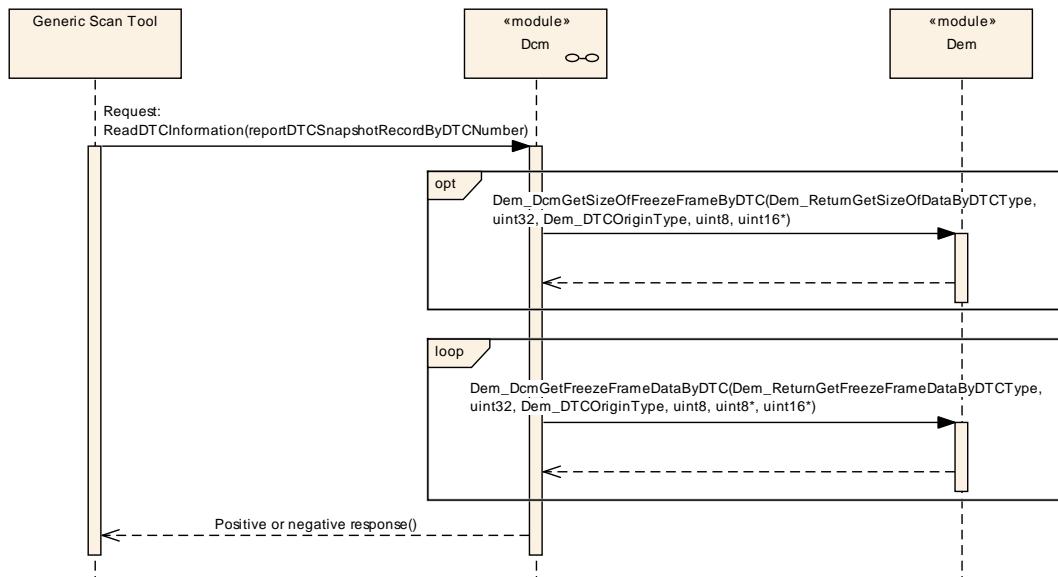


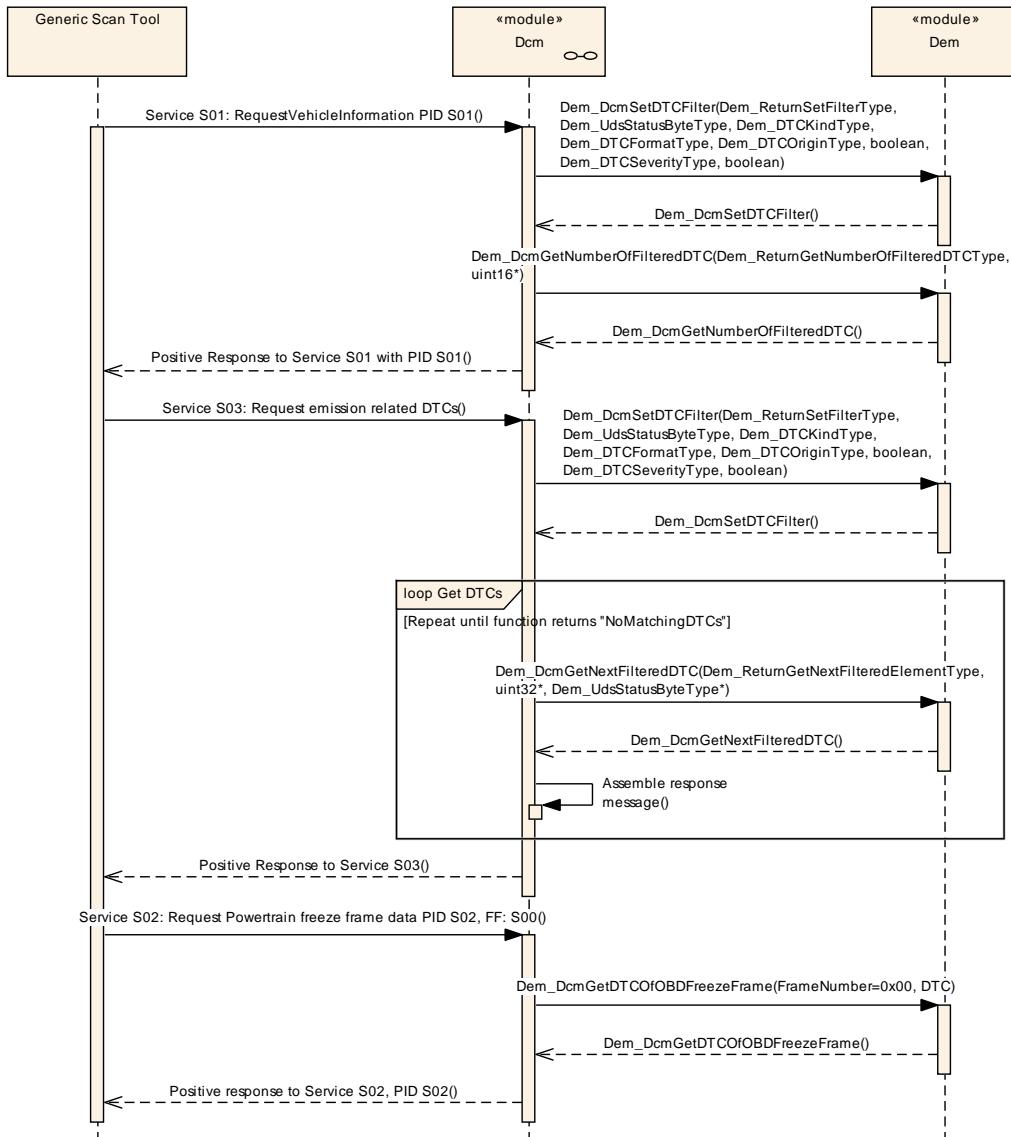
Figure 9.6: Sequence diagram of Dem\_DcmGetStatusOfDTC

## 9.6 Dem\_DcmGetFreezeFrameDataByDTC



**Figure 9.7: Sequence diagram of Dem\_DcmGetFreezeFrameDataByDTC**

## 9.7 GetOBDFaultInformation



**Figure 9.8: Sequence diagram of GetOBDFaultInformation**

## 9.8 ReportDTCByStatusMask

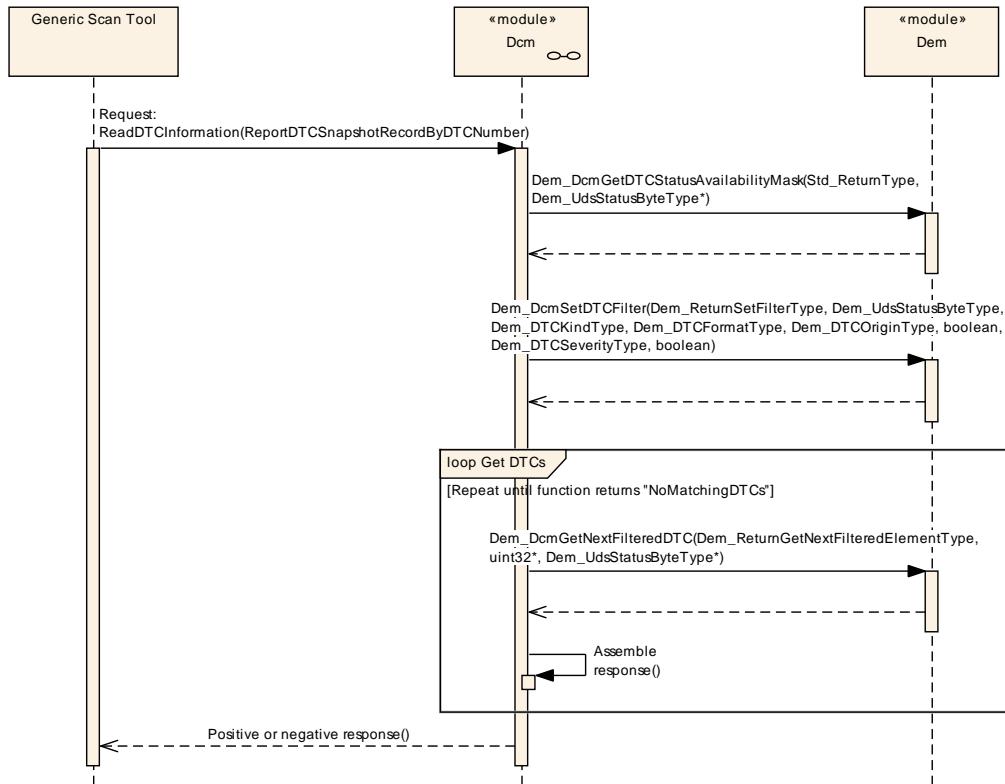


Figure 9.9: Sequence diagram of ReportDTCStatusMask

## 9.9 FiM\_DemTriggerOnEventStatus

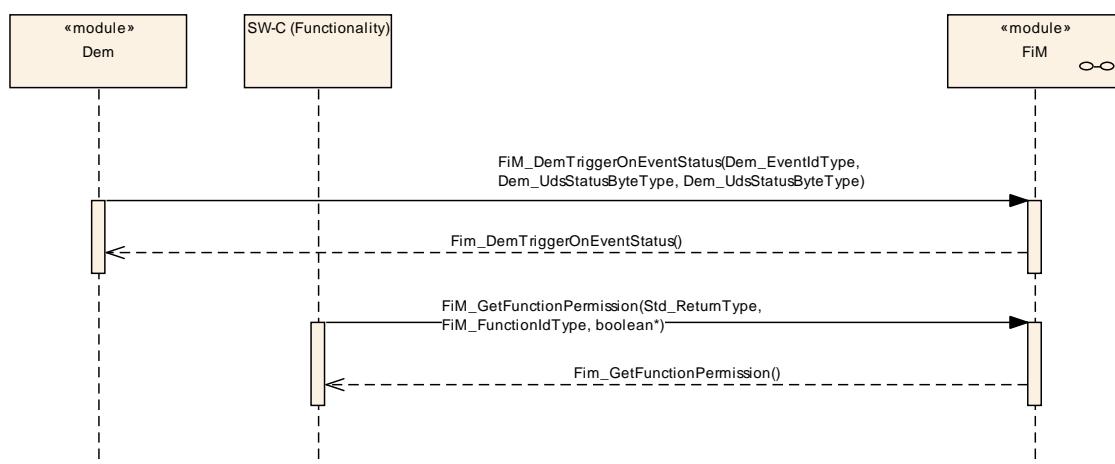


Figure 9.10: Sequence diagram of FiM\_DemTriggerOnEventStatus

## 9.10 ProcessEvent (Example)

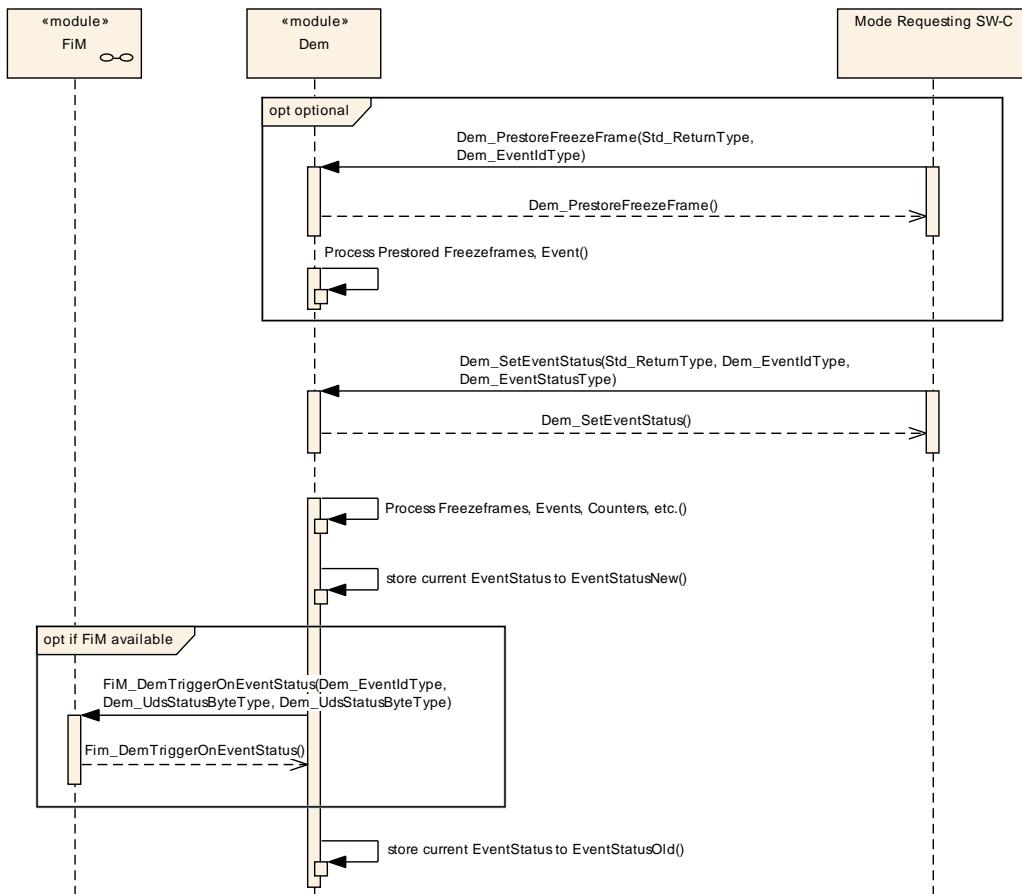


Figure 9.11: Sequence diagram for an example of ProcessEvent Dem-internally

## 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 the chapter 10.1 “Introduction to configuration specification” in SWS\_BSWGeneral

### 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 Variants

The following use-cases and concepts for the Dem are the base for the different Dem configuration variants:

1. Simple ECU configuration, with optimized ROM usage: results in a pre-compile time variant
2. Multiple ECU configuration support: results in a post-build time selectable variant (in combination with a superset method). For this use-case, some DTC values have different values in the different configuration sets. Additionally it is possible, to omit events/DTCs (for the outside world) completely in some of the configuration sets.
3. Support of different PID values for Europe/USA results in a post-build time variant (refer to [DemEventFailureCycleCounterThreshold](#), [DemPidIdentifier](#) and [Dem-PidDataElementClassRef](#))
4. Support of different DTC kind for Europe/USA results in a post-build time variant

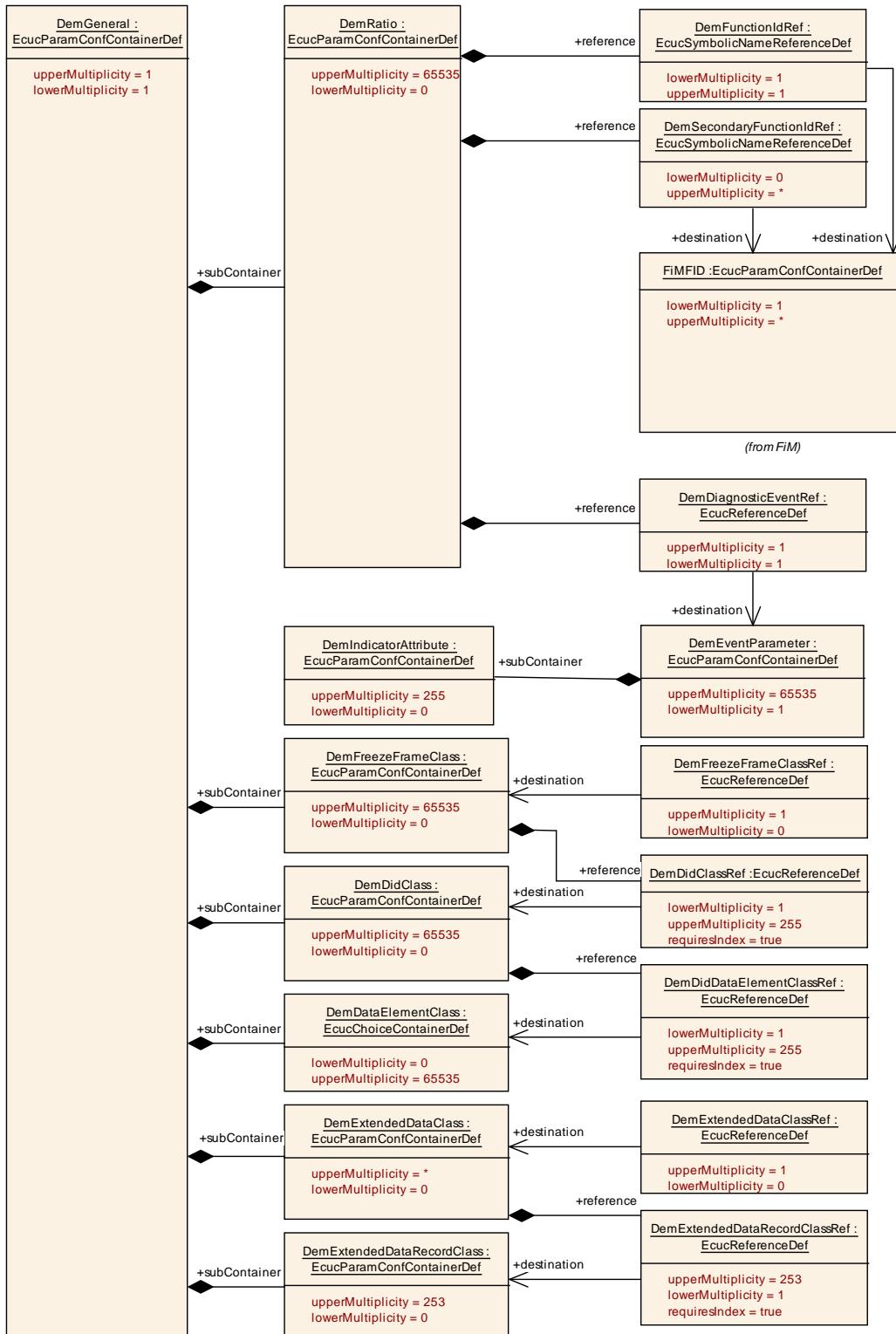
The parameter [DemEventId](#) and the whole debouncing-configuration (including vendor-specific debounce extensions) are always pre-compile time variant, indepen-

dent of configuration concepts above, according to the pre-initialization handling (refer to chapter 7.7).

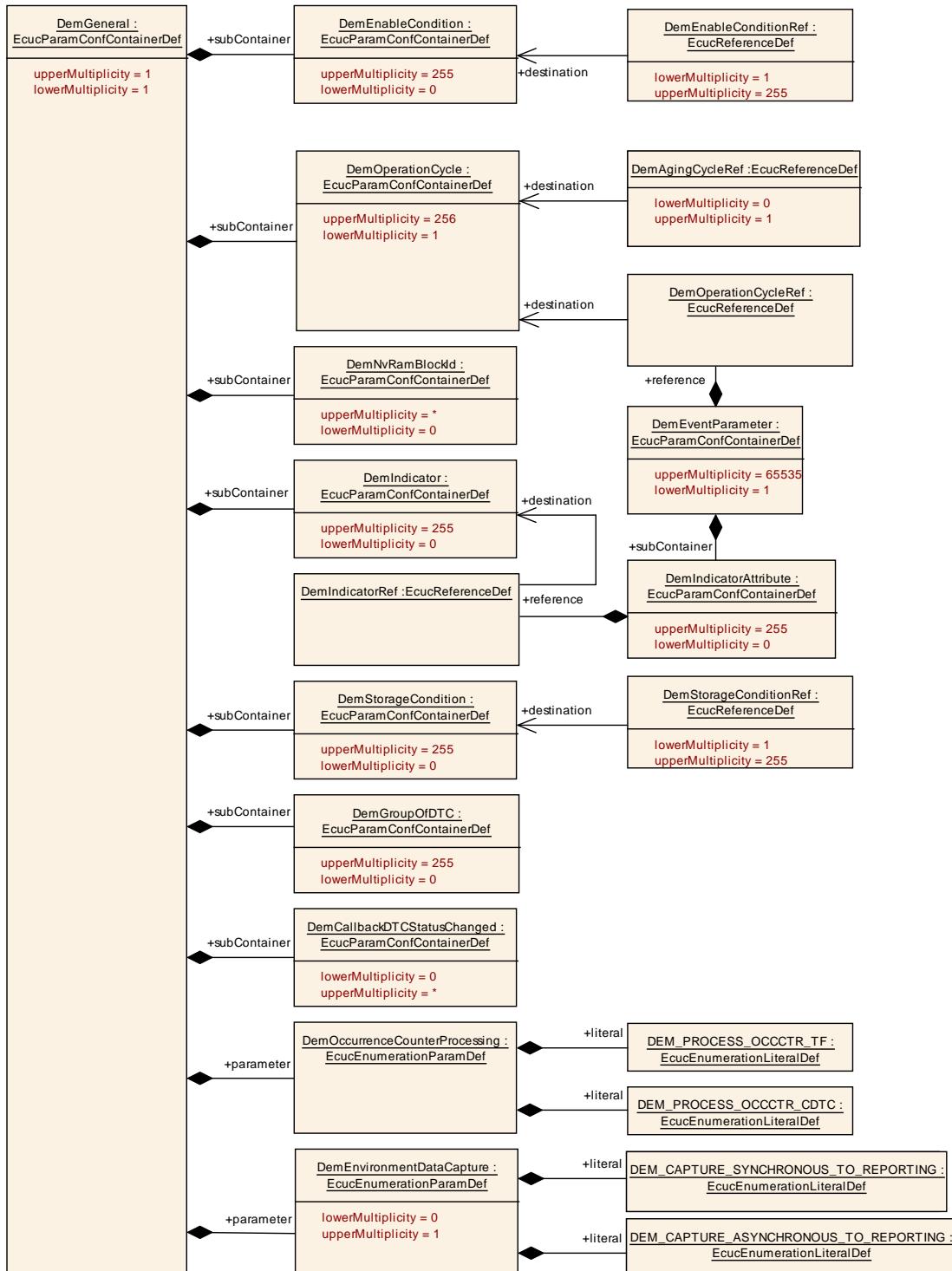
The following configuration parameters shall be available:

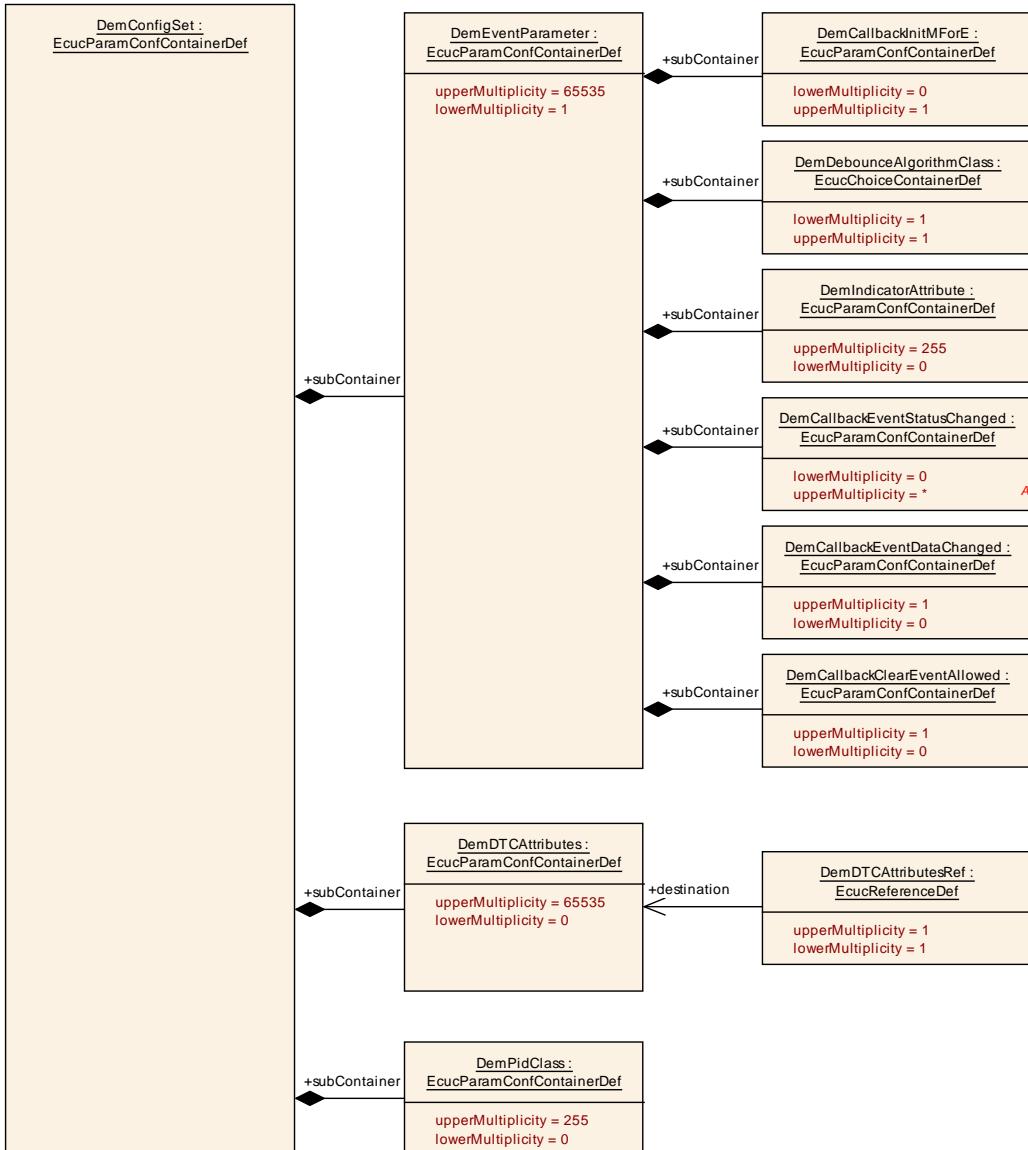
- **[SWS\_Dem\_00267]** 「VARIANT-PRE-COMPILe: only pre-compile time configuration parameters.」(*SRS\_BSW\_00334, SRS\_BSW\_00345, SRS\_BSW\_00396, SRS\_BSW\_00397, SRS\_BSW\_00398, SRS\_BSW\_00399, SRS\_BSW\_00400, SRS\_BSW\_00401, SRS\_BSW\_00404, SRS\_BSW\_00405*)
- **[SWS\_Dem\_00268]** 「VARIANT-POST-BUILD: mix of pre-compile- and post build time-configuration parameters (using post build selectable, refer to chapter 4.1)」(*SRS\_BSW\_00334, SRS\_BSW\_00345, SRS\_BSW\_00396, SRS\_BSW\_00397, SRS\_BSW\_00398, SRS\_BSW\_00399, SRS\_BSW\_00400, SRS\_BSW\_00401, SRS\_BSW\_00404, SRS\_BSW\_00405*)

Link time configurable parameters (VARIANT-LINK-TIME) are not used in this specification.



**Figure 10.1: Configuration overview for DemGeneral (part I)**


**Figure 10.2: Configuration overview for DemGeneral (part II)**



**Figure 10.3: Configuration overview for DemConfigSet**

## 10.2.2 Dem

<b>Module Name</b>	Dem	
<b>Module Description</b>	Configuration of the Dem (Diagnostic Event Manager) module.	
<b>Post-Build Variant Support</b>	true	
<b>Included Containers</b>		
Container Name	Multiplicity	Scope / Dependency
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

### 10.2.3 DemGeneral

#### DemGeneral

<b>SWS Item</b>	[ECUC_Dem_00677]
<b>Container Name</b>	DemGeneral
<b>Description</b>	This container contains the configuration (parameters) of the BSW Dem
<b>Configuration Parameters</b>	

<b>Name</b>	DemAgingCycleCounterProcessing [ECUC_Dem_00603] (Obsolete)		
<b>Description</b>	This configuration switch defines, whether the aging counter is calculated Dem-internally or provided via Dem_SetAgingCycleCounterValue.		
<b>Tags:</b> atp.Status=obsolete atp.StatusComment=This reference is set to obsolete and will be removed in release 4.3. atp.StatusRevisionBegin=4.2.2			
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_PROCESS_AGING CTR_EXTERN	based on API Dem_SetAgingCycleCounterValue	
<b>Post-Build Variant Value</b>	DEM_PROCESS_AGING CTR_INTERN false	based on reported cycle states	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemAgingRequieresTestedCycle [ECUC_Dem_00877]		
<b>Description</b>	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		
<b>Multiplicity</b>			
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	

<b>Scope / Dependency</b>	scope: ECU		
---------------------------	------------	--	--

<b>Name</b>	DemAvailabilitySupport [ECUC_Dem_00878]		
<b>Description</b>	This configuration switch defines, whether support for availability is enabled or not.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_EVENT_AVAILABILITY	Support availability by Event	
	DEM_NO_AVAILABILITY	Availability is not supported	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemBswErrorBufferSize [ECUC_Dem_00625]				
<b>Description</b>	Maximum number of elements in buffer for handling of BSW errors (ref. to SWS_Dem_00207).				
<b>Multiplicity</b>	0..1				
<b>Type</b>	EcucIntegerParamDef				
<b>Range</b> <b>Default Value</b>	0 .. 255				
<b>Post-Build Variant Multiplicity</b>	false				
<b>Post-Build Variant Value</b>	false				
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Scope / Dependency</b>	scope: ECU				

<b>Name</b>	DemClearDTCBehavior [ECUC_Dem_00766]		
<b>Description</b>	Defines the clearing process of diagnostic information for volatile and non-volatile memory and the positive response handling for the Dcm module.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_CLRRESP_NONVOLATILE_FINISH	Return DEM_CLEAR_OK after volatile and non-volatile event memory data cleared.	

<b>Post-Build Variant Value</b>	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 false	Return DEM_CLEAR_OK after volatile event memory data cleared	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemClearDTCLimitation [ECUC_Dem_00790]				
<b>Description</b>	Defines the supported Dem_<...>ClearDTC API scope.				
<b>Multiplicity</b>	1				
<b>Type</b>	EcucEnumerationParamDef				
<b>Range</b>	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.			
<b>Default Value</b>	DEM_ALL_SUPPORTED_DTCs				
<b>Post-Build Variant Value</b>	false				
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Scope / Dependency</b>	scope: ECU				

<b>Name</b>	DemDataElementDefaultEndianness [ECUC_Dem_00858]		
<b>Description</b>	Defines the default endianness of the data belonging to a data element which is applicable if the DemExternalSRDataElementClass does not define a endianness.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	BIG_ENDIAN	Most significant byte shall come at the lowest address.	
	LITTLE_ENDIAN	Most significant byte shall come highest address	
	OPAQUE false	opaque data endianness	
<b>Post-Build Variant Value</b>			

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemDebounceCounterBasedSupport [ECUC_Dem_00724]		
<b>Description</b>	<p>This configuration switch defines, whether support for counter based debouncing is enabled or not.</p> <p>true: counter based debouncing support is enabled false: counter based debouncing support is disabled</p>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDebounceTimeBasedSupport [ECUC_Dem_00725]		
<b>Description</b>	<p>This configuration switch defines, whether support for time based debouncing is enabled or not.</p> <p>true: time based debouncing support is enabled false: time based debouncing support is disabled</p>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDevErrorDetect [ECUC_Dem_00648]		
<b>Description</b>	Switches the Default Error Tracer (Det) detection and notification ON or OFF. <ul style="list-style-type: none"> <li>• true: enabled (ON).</li> <li>• false: disabled (OFF).</li> </ul>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X — —	All Variants
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemDtcStatusAvailabilityMask [ECUC_Dem_00652]		
<b>Description</b>	Mask for the supported DTC status bits by the Dem. This mask is used by UDS service 0x19.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X — —	All Variants
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemEnvironmentDataCapture [ECUC_Dem_00895]		
<b>Description</b>	DemEnvironmentDataCapture defines the point in time, when the data actually is captured.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_CAPTURE_ASYNC_HRONOUS_TO_REPORTING	The data capturing is postponed to the next cycle of the Dem_Mainfunction. (This means that there is a minimum delay between report of the failure and capturing the data).	
	DEM_CAPTURE_SYNCHRONOUS_TO_REPORTING	The data is captured immediately within the reporting function (i.e. in the context of the setEventStatus/reportErrorStatus function).	
<b>Post-Build Variant Multiplicity</b>	false		

<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemEventCombinationSupport [ECUC_Dem_00740]		
<b>Description</b>	This parameter defines the type of event combination supported by the Dem.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_EVCOMB_DISABLE_D	No event combination supported	
	DEM_EVCOMB_ONRETRIEVAL	Event combination on retrieval	
	DEM_EVCOMB_ONSTORAGE	Event combination on storage	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemEventDisplacementStrategy [ECUC_Dem_00742]		
<b>Description</b>	This configuration switch defines, whether support for event displacement is enabled or not, and which displacement strategy is followed.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	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_PARIOCC	Event memory entry displacement is enabled, by consideration of priority and occurrence (but without active/passive status).	
<b>Post-Build Variant Value</b>	false		

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemEventMemoryEntryStorageTrigger [ECUC_Dem_00797]		
<b>Description</b>	Configures the primary trigger to allocate an event memory entry.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_TRIGGER_ON_CO_NFIRMED	Event Memory entries are triggered if the UDS DTC status bit 3 (confirmedDTC) changes from 0 to 1.	
	DEM_TRIGGER_ON_FD_C_THRESHOLD	Event Memory entries are triggered when the FDC threshold is reached.	
	DEM_TRIGGER_ON_PENDING	Event Memory entries are triggered if the UDS DTC status bit 2 (pendingDTC) changes from 0 to 1.	
	DEM_TRIGGER_ON_TEST_FAILED	Event Memory entries are triggered if the UDS DTC status bit 0 (testFailed) changes from 0 to 1.	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemGeneralInterfaceSupport [ECUC_Dem_00880]		
<b>Description</b>	The interfaces GeneralEvtInfo, GeneralCallbackEventDataChanged and GeneralCallbackEventStatusChange are provided if DemGeneralInterfaceSupport is equal to true.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemHeaderFileInclusion [ECUC_Dem_00722]		
<b>Description</b>	Name of the header file(s) to be included by the Dem module containing the used C-callback declarations.		
<b>Multiplicity</b>	0..*		
<b>Type</b>	EcucStringParamDef		
<b>Default Value</b>			
<b>Regular Expression</b>	[a-zA-Z0-9\_\_](a-zA-Z0-9\_\_)*		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemImmediateNvStorageLimit [ECUC_Dem_00738]				
<b>Description</b>	This parameter defines the maximum number of occurrences, a specific event memory entry is allowed, to be stored in NVRAM immediately (refer to DemImmediateNvStorage).				
<b>Multiplicity</b>	0..1				
<b>Type</b>	EcucIntegerParamDef				
<b>Range</b>	1 .. 255				
<b>Default Value</b>					
<b>Post-Build Variant Multiplicity</b>	false				
<b>Post-Build Variant Value</b>	false				
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Scope / Dependency</b>	scope: ECU				

<b>Name</b>	DemMILIndicatorRef [ECUC_Dem_00723]		
<b>Description</b>	<p>This parameter defines the indicator representing the MIL.</p> <p>This parameter is mandatory for ECUs supporting OBD (refer to DemOBDSupport).</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemIndicator		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	<p>scope: ECU</p> <p>dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU</p>		

<b>Name</b>	DemMaxNumberEventEntryEventBuffer [ECUC_Dem_00899]		
<b>Description</b>	Specifies the size of the buffer for storing environmental data (freezeframes and extended data) until they are processed and stored to the event memory.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b> <b>Default Value</b>	1 .. 250		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	<p>scope: ECU</p> <p>dependency: Required for DemEnvironmentDataCapture == DEM_CAPTURE_SYNCHRONOUS_TO_REPORTING</p>		

<b>Name</b>	DemMaxNumberEventEntryPermanent [ECUC_Dem_00689]		
<b>Description</b>	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.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemMaxNumberPrestoredFF [ECUC_Dem_00692]		
<b>Description</b>	Defines the maximum number for prestored freeze frames. If set to 0, then freeze frame prestorage is not supported by the ECU.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemOBDSupport [ECUC_Dem_00698]		
<b>Description</b>	This configuration switch defines OBD support and kind of OBD ECU.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	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: Primary ECU	
<b>Post-Build Variant Value</b>	false		

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemOccurrenceCounterProcessing [ECUC_Dem_00767]		
<b>Description</b>	This configuration switch defines the consideration of the fault confirmation process for the occurrence counter. For OBD and mixed systems (OBD/non OBD, refer to DemOBDSupport) configuration switch shall always set to DEM_PROCESS_OCCCTR_TF.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_PROCESS_OCCCTR_CDTC	the occurrence counter is triggered by the TestFailed 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.	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport, DemGeneralJ1939		

<b>Name</b>	DemOperationCycleStatusStorage [ECUC_Dem_00764]		
<b>Description</b>	Defines if the operation cycle state is available over the power cycle (stored non-volatile) or not. true: the operation cycle state is stored non-volatile false: the operation cycle state is only stored volatile		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemPTOSupport [ECUC_Dem_00704]		
<b>Description</b>	This configuration switch defines, whether PTO support (and therefore PID \$1E support) is enabled or not.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemResetConfirmedBitOnOverflow [ECUC_Dem_00799]		
<b>Description</b>	This configuration switch defines, whether the confirmed bit is reset or not while an event memory entry will be displaced.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>	true		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemStatusBitHandlingTestFailedSinceLastClear [ECUC_Dem_00784]				
<b>Description</b>	This configuration switch defines, whether the aging and displacement mechanism shall be applied to the "TestFailedSinceLastClear" status bits.				
<b>Multiplicity</b>	1				
<b>Type</b>	EcucEnumerationParamDef				
<b>Range</b>	DEM_STATUS_BIT_AGIN G_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_NOR MAL	aging and displacement has no impact on the "TestFailedSinceLastClear" status bits			
<b>Default Value</b>	DEM_STATUS_BIT_NORMAL				
<b>Post-Build Variant Value</b>	false				
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Scope / Dependency</b>	scope: ECU				

<b>Name</b>	DemStatusBitStorageTestFailed [ECUC_Dem_00714]		
<b>Description</b>	Activate/Deactivate the permanent storage of the "TestFailed" status bits.  true: storage activated false: storage deactivated		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemSuppressionSupport [ECUC_Dem_00793]		
<b>Description</b>	This configuration switch defines, whether support for suppression is enabled or not.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_DTC_SUPPRESSIO N	Support suppression by DTC	
	DEM_NO_SUPPRESSIO N	Suppression is not supported	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemTaskTime [ECUC_Dem_00715]											
<b>Description</b>	<p>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.</p> <p>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.</p> <p>min: A negative value is not allowed.</p> <p>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).</p> <p>upperMultiplicity: Exactly one TaskTime must be specified per configuration.</p> <p>lowerMultiplicity: Exactly one TaskTime must be specified per configuration.</p>											
<b>Multiplicity</b>	1											
<b>Type</b>	EcucFloatParamDef											
<b>Range</b>	0.001 .. 0.1											
<b>Default Value</b>												
<b>Post-Build Variant Value</b>	false											
<b>Value Configuration Class</b>	<table border="1"> <tr> <td><b>Pre-compile time</b></td> <td>X</td> <td>All Variants</td> </tr> <tr> <td><b>Link time</b></td> <td>—</td> <td></td> </tr> <tr> <td><b>Post-build time</b></td> <td>—</td> <td></td> </tr> </table>	<b>Pre-compile time</b>	X	All Variants	<b>Link time</b>	—		<b>Post-build time</b>	—			
<b>Pre-compile time</b>	X	All Variants										
<b>Link time</b>	—											
<b>Post-build time</b>	—											
<b>Scope / Dependency</b>	scope: ECU											

<b>Name</b>	DemTriggerDcmReports [ECUC_Dem_00754]											
<b>Description</b>	<p>Activate/Deactivate the notification to the Diagnostic Communication Manager for ROE processing.</p> <p>true: Dcm ROE notification activated false: Dcm ROE notification deactivated</p>											
<b>Multiplicity</b>	1											
<b>Type</b>	EcucBooleanParamDef											
<b>Default Value</b>												
<b>Post-Build Variant Value</b>	false											
<b>Value Configuration Class</b>	<table border="1"> <tr> <td><b>Pre-compile time</b></td> <td>X</td> <td>All Variants</td> </tr> <tr> <td><b>Link time</b></td> <td>—</td> <td></td> </tr> <tr> <td><b>Post-build time</b></td> <td>—</td> <td></td> </tr> </table>	<b>Pre-compile time</b>	X	All Variants	<b>Link time</b>	—		<b>Post-build time</b>	—			
<b>Pre-compile time</b>	X	All Variants										
<b>Link time</b>	—											
<b>Post-build time</b>	—											
<b>Scope / Dependency</b>	scope: ECU											

<b>Name</b>	DemTriggerDltReports [ECUC_Dem_00718]		
<b>Description</b>	Activate/Deactivate the notification to the Diagnostic Log and Trace.  true: Dlt notification activated false: Dlt notification deactivated		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemTriggerFiMReports [ECUC_Dem_00719]		
<b>Description</b>	Activate/Deactivate the notification to the Function Inhibition Manager.  true: FiM notification activated false: FiM notification deactivated		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemTriggerMonitorInitBeforeClearOk [ECUC_Dem_00765]		
<b>Description</b>	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		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemClearDTCBehavior		

<b>Name</b>	DemTypeOfDTCSupported [ECUC_Dem_00720]		
<b>Description</b>	This parameter defines the format returned by Dem_DcmGetTranslationType and does not relate to/influence the supported Dem functionality.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_DTC_TRANSLATIO_N_ISO11992_4	ISO11992-4 DTC format	
	DEM_DTC_TRANSLATIO_N_ISO14229_1	ISO14229-1 DTC format (3 byte format)	
	DEM_DTC_TRANSLATIO_N_ISO15031_6	ISO15031-6 DTC format = SAE_J2012-DA_DTCFormat_00 (2 byte format)	
	DEM_DTC_TRANSLATIO_N_SAEJ1939_73	SAEJ1939-73 DTC format	
	DEM_DTC_TRANSLATIO_N_SAE_J2012_DA_DTCFORMAT_04	SAE_J2012-DA_DTCFormat_00 (3 byte format)	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemTypeOfFreezeFrameRecordNumeration [ECUC_Dem_00778]		
<b>Description</b>	This parameter defines the type of assigning freeze frame record numbers for event-specific freeze frame records.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	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	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemVersionInfoApi [ECUC_Dem_00721]		
<b>Description</b>	Activate/Deactivate the version information API.  true: version information activated false: version information deactivated		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Included Containers</b>			
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>	
DemCallbackDTCStatusChanged	0..*	<p>The presence of this container indicates, that the Dem has access to a "DTCStatusChanged" callback, which the Dem will call to notify other components about the change in the status of a DTC.</p> <p>In case there is a DemCallbackDTCStatusChangedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackDTCStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackDTCStatusChanged whose name is generated by using the unique callback-prefix followed by the event name.</p> <p>Status change notifications are supported for DTCs in primary memory only.</p>	
DemDataElementClass	0..65535	This container contains the configuration (parameters) for an internal/external data element class.	
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.	
DemExtendedDataClass	0..*	This class contains the combinations of extended data records for an extended data class.	
DemExtendedDataRecordClass	0..253	<p>This container contains the configuration (parameters) for an extended data record class.</p> <p>It is assembled out of one or several data elements.</p>	
DemFreezeFrameClass	0..65535	This container contains the combinations of DIDs for a non OBD2 and WWH-OBD relevant freeze frame class.	

DemFreezeFrameRecNumClass	0..255	<p>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.</p> <p>dependency:  <code>DemTypeOfFreezeFrameRecordNumeration = DEM_FF_RECNUM_CONFIGURED</code></p>
DemFreezeFrameRecordClass	0..255	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.
DemGroupOfDTC	0..255	This container contains the configuration (parameters) for DTC groups.
DemIndicator	0..255	This container contains the configuration (parameters) for Indicators.
DemMirrorMemory	0..1	This container contains the mirror event memory specific parameters of the Dem module.
DemNvRamBlockId	0..*	<p>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.</p> <p>The number of blocks which are necessary depends on the implementation and configuration (e.g. number of used event memories) of the Dem module.</p>
DemOperationCycle	1..256	This container holds all parameters that are relevant to configure an operation cycle.
DemPrimaryMemory	1	This container contains the primary event memory specific parameters of the Dem module.
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.
DemUserDefinedMemory	0..30	This container contains the user defined event memory specific parameters of the Dem module.

## 10.2.4 DemGeneralOBD

### DemGeneralOBD

SWS Item	[ECUC_Dem_00756]
Container Name	DemGeneralOBD

<b>Description</b>	This container contains the general OBD-specific configuration (parameters) of the Dem module.
<b>Configuration Parameters</b>	

<b>Name</b>	DemOBDCentralizedPID21Handling [ECUC_Dem_00794]		
<b>Description</b>	Switch to enable the centralized handling of PID \$21.  true: centralized handling of PID \$21 enabled  false: centralized handling of PID \$21 disabled		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemOBDCentralizedPID31Handling [ECUC_Dem_00879]		
<b>Description</b>	Switch to enable the centralized handling of PID \$31.  true: centralized handling of PID \$31 enabled  false: centralized handling of PID \$31 disabled		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemOBDCCompliance [ECUC_Dem_00795]		
<b>Description</b>	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		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	-	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU dependency: (DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU) or (DemJ1939Readiness1Support == true)		

<b>Name</b>	DemOBDDestinationOfEventsRef [ECUC_Dem_00836] (Obsolete)		
<b>Description</b>	The destination of events assigns where the OBD events shall be stored.  <b>Tags:</b> atp.Status=obsolete atp.StatusComment=This reference is set to obsolete and will be removed in release 4.3. atp.StatusRevisionBegin=4.2.2		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Choice Reference to [DemPrimaryMemory,DemUserDefinedMemory]		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemOBDEngineType [ECUC_Dem_00900]		
<b>Description</b>	Switch to provide either Gasoline or Diesel parameters.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM IGNITION COMPRESSION	Diesel engine type	
<b>Post-Build Variant Multiplicity</b>	DEM IGNITION SPARK	Gasoline engine type	
	false		

<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemOBDEventDisplacement [ECUC_Dem_00796]		
<b>Description</b>	Activate/Deactivate a different displacement behavior for OBD events.  OBD events with special Conditions (e.g. Pending, MIL_On...) shall not be displaced.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemOBDInputAcceleratorPedalInformation [ECUC_Dem_00763]		
<b>Description</b>	Input variable for the accelerator pedal information, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Choice Reference to [DemExternalCSDaDataElement-Class,DemExternalSRDataElementClass]		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemOBDInputAmbientPressure [ECUC_Dem_00762]		
<b>Description</b>	Input variable for the ambient pressure, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Choice Reference to [DemExternalCSDDataElement-Class,DemExternalSRDataElementClass]		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemOBDInputAmbientTemperature [ECUC_Dem_00761]		
<b>Description</b>	Input variable for the ambient temperature, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Choice Reference to [DemExternalCSDDataElement-Class,DemExternalSRDataElementClass]		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	

<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU
---------------------------	--

<b>Name</b>	DemOBDInputDistanceInformation [ECUC_Dem_00759]		
<b>Description</b>	Input variable for the distance information, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Choice Reference to [DemExternalCSDDataElement-Class, DemExternalSRDataElementClass]		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemOBDInputEngineSpeed [ECUC_Dem_00757]		
<b>Description</b>	Input variable for the engine speed, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Choice Reference to [DemExternalCSDDataElement-Class, DemExternalSRDataElementClass]		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemOBDInputEngineTemperature [ECUC_Dem_00772]		
<b>Description</b>	Input variable for the engine temperature, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Choice Reference to [DemExternalCSDataElement-Class,DemExternalSRDataElementClass]		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemOBDInputProgrammingEvent [ECUC_Dem_00760]		
<b>Description</b>	Input variable for the programming event, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Choice Reference to [DemExternalCSDataElement-Class,DemExternalSRDataElementClass]		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemOBDInputVehicleSpeed [ECUC_Dem_00758]		
<b>Description</b>	Input variable for the vehicle speed, which is assigned to a specific data element used as interface for the Dem-internal PID calculations.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Choice Reference to [DemExternalCSDDataElement-Class, DemExternalSRDataElementClass]		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemOBDTIMESinceEngineStart [ECUC_Dem_00827]		
<b>Description</b>	Input variable for the Time Since Engine Start information, which is assigned to a specific data element.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Choice Reference to [DemExternalCSDDataElement-Class, DemExternalSRDataElementClass, DemInternalDataElementClass]		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DemCallbackOBDDTCStatusChanged	0..*	<p>The presence of this container indicates, that the Dem has access to a "DTCStatusChanged" callback, which the Dem will call to notify other components about the change in the status of a DTC.</p> <p>In case there is a DemCallbackDTCStatusChangedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackDTCStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackDTCStatusChanged whose name is generated by using the unique callback-prefix followed by the event name.</p> <p>Status change notifications are supported for DTCs in primary memory only.</p>

### 10.2.5 DemGeneralJ19139

#### DemGeneralJ1939

SWS Item	[ECUC_Dem_00864]
Container Name	DemGeneralJ1939
Description	This container contains the general J1939-specific configuration (parameters) of the Dem module. If the container exists the J1939 support is enabled.

#### Configuration Parameters

Name	DemAmberWarningLampIndicatorRef [ECUC_Dem_00821]		
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		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
Scope / Dependency	scope: local		

<b>Name</b>	DemJ1939ClearDtcSupport [ECUC_Dem_00872]		
<b>Description</b>	This configuration switch defines whether clearing J1939 DTCs (DM03 und DM11) is supported or not.  This switches on and off the API Dem_J1939DcmClearDTC.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemJ1939Dm31Support [ECUC_Dem_00868]		
<b>Description</b>	This configuration switch defines whether J1939 DM31 is supported or not.  This switches on and off the APIs Dem_J1939DcmFirstDTCwithLampStatus and Dem_J1939DcmGetNextDTCwithLampStatus. .		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemJ1939ExpandedFreezeFrameSupport [ECUC_Dem_00873]		
<b>Description</b>	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.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	

<b>Scope / Dependency</b>	scope: local
---------------------------	--------------

<b>Name</b>	DemJ1939FreezeFrameSupport [ECUC_Dem_00866]		
<b>Description</b>	<p>This configuration switch defines whether J1939 freeze frames are supported or not.</p> <p>This switches on and off the APIs Dem_J1939DcmSetFreezeFrameFilter and Dem_J1939DcmGetNextFreezeFrame.</p>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X — —	All Variants
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemJ1939RatioSupport [ECUC_Dem_00867]		
<b>Description</b>	<p>This configuration switch defines whether J1939 performance ratios are supported or not.</p> <p>This switches on and off the APIs Dem_J1939DcmSetRatioFilter and Dem_J1939DcmGetNextFilteredRatio.</p>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X — —	All Variants
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemJ1939Readiness1Support [ECUC_Dem_00869]		
<b>Description</b>	<p>This configuration switch defines whether J1939 diagnostic readiness 1 is supported or not.</p> <p>This switches on and off the API Dem_J1939DcmReadDiagnosticReadiness1.</p>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemJ1939Readiness2Support [ECUC_Dem_00870]		
<b>Description</b>	<p>This configuration switch defines whether J1939 diagnostic readiness 2 is supported or not.</p> <p>This switches on and off the API Dem_J1939DcmReadDiagnosticReadiness2.</p>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemJ1939Readiness3Support [ECUC_Dem_00871]		
<b>Description</b>	<p>This configuration switch defines whether J1939 diagnostic readiness 3 is supported or not.</p> <p>This switches on and off the API Dem_J1939DcmReadDiagnosticReadiness3.</p>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemJ1939ReadingDtcSupport [ECUC_Dem_00865]		
<b>Description</b>	<p>This configuration switch defines whether J1939 DTC readout is supported or not.</p> <p>This switches on and off the APIs Dem_J1939DcmSetDTCFilter, Dem_J1939DcmGetNumberOfFilteredDTC and Dem_J1939DcmGetNextFilteredDTC.</p>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X — —	All Variants
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemProtectLampIndicatorRef [ECUC_Dem_00822]		
<b>Description</b>	<p>This parameter defines the indicator representing the ProtectLamp.</p> <p>This parameter may be used for ECUs supporting J1939.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemIndicator		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X — —	All Variants
<b>Value Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X — —	All Variants
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemRedStopLampIndicatorRef [ECUC_Dem_00820]		
<b>Description</b>	<p>This parameter defines the indicator representing the RedStopLamp.</p> <p>This parameter may be used for ECUs supporting J1939.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemIndicator		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X — —	All Variants

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
DemCallbackJ1939DTC StatusChanged	0..*	<p>The presence of this container indicates, that the Dem has access to a "DTCStatusChanged" callback, which the Dem will call to notify other components about the change in the status of a DTC.</p> <p>In case there is a DemCallbackDTCStatusChangedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackDTCStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackDTCStatusChanged whose name is generated by using the unique callback-prefix followed by the event name.</p> <p>Status change notifications are supported for DTCs in primary memory only.</p>
DemJ1939FreezeFrame Class	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.

## 10.2.6 DemOperationCycle

### DemOperationCycle

<b>SWS Item</b>	[ECUC_Dem_00701]
<b>Container Name</b>	DemOperationCycle
<b>Description</b>	This container holds all parameters that are relevant to configure an operation cycle.
<b>Configuration Parameters</b>	

<b>Name</b>	DemOperationCycleAutomaticEnd [ECUC_Dem_00837]
<b>Description</b>	If DemOperationCycleAutomaticEnd is configured to TRUE, Dem shall automatically end the driving cycle at either Dem_Shutdown() or Dem_Init().
<b>Multiplicity</b>	1
<b>Type</b>	EcucBooleanParamDef
<b>Default Value</b>	false
<b>Post-Build Variant Value</b>	false

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECUECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemOperationCycleAutostart [ECUC_Dem_00805]		
<b>Description</b>	The autostart property defines if the operation cycles is automatically (re-)started during Dem_PrelInit.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemOperationCyclegd [ECUC_Dem_00898]				
<b>Description</b>	This parameter's value is used, together with the aggregating container, to define a symbolic name of the operation cycle.				
<b>Multiplicity</b>	0..1				
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)				
<b>Range</b>	0 .. 18446744073709551615				
<b>Default Value</b>					
<b>Post-Build Variant Multiplicity</b>	false				
<b>Post-Build Variant Value</b>	false				
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Scope / Dependency</b>	scope: local				

<b>Name</b>	DemOperationCycleType [ECUC_Dem_00703]														
<b>Description</b>	Operation cycles types for the Dem to be supported by cycle-state APIs.  Further cycle types can be specified as part of the Dem delivery.														
<b>Multiplicity</b>	1														
<b>Type</b>	EcucEnumerationParamDef														
<b>Range</b>	<table border="1"> <tr> <td>DEM_OPCYC_IGNITION</td> <td>Ignition ON / OFF cycle</td> </tr> <tr> <td>DEM_OPCYC_OBD_DCY</td> <td>OBD Driving cycle</td> </tr> <tr> <td>DEM_OPCYC_OTHER</td> <td>further operation cycle</td> </tr> <tr> <td>DEM_OPCYC_POWER</td> <td>Power ON / OFF cycle</td> </tr> <tr> <td>DEM_OPCYC_TIME</td> <td>Time based operation cycle</td> </tr> <tr> <td>DEM_OPCYC_WARMUP</td> <td>OBD OBD Warm up cycle</td> </tr> </table>			DEM_OPCYC_IGNITION	Ignition ON / OFF cycle	DEM_OPCYC_OBD_DCY	OBD Driving cycle	DEM_OPCYC_OTHER	further operation cycle	DEM_OPCYC_POWER	Power ON / OFF cycle	DEM_OPCYC_TIME	Time based operation cycle	DEM_OPCYC_WARMUP	OBD OBD Warm up cycle
DEM_OPCYC_IGNITION	Ignition ON / OFF cycle														
DEM_OPCYC_OBD_DCY	OBD Driving cycle														
DEM_OPCYC_OTHER	further operation cycle														
DEM_OPCYC_POWER	Power ON / OFF cycle														
DEM_OPCYC_TIME	Time based operation cycle														
DEM_OPCYC_WARMUP	OBD OBD Warm up cycle														
<b>Post-Build Variant Value</b>	false														
<b>Value Configuration Class</b>	<table border="1"> <tr> <td><b>Pre-compile time</b></td> <td>X</td> <td>All Variants</td> </tr> <tr> <td><b>Link time</b></td> <td>-</td> <td></td> </tr> <tr> <td><b>Post-build time</b></td> <td>-</td> <td></td> </tr> </table>	<b>Pre-compile time</b>	X	All Variants	<b>Link time</b>	-		<b>Post-build time</b>	-						
<b>Pre-compile time</b>	X	All Variants													
<b>Link time</b>	-														
<b>Post-build time</b>	-														
<b>Scope / Dependency</b>	scope: ECU														

#### No Included Containers

### 10.2.7 DemEnableCondition

#### DemEnableCondition

<b>SWS Item</b>	[ECUC_Dem_00653]
<b>Container Name</b>	DemEnableCondition
<b>Description</b>	This container contains the configuration (parameters) for enable conditions.
<b>Configuration Parameters</b>	

<b>Name</b>	DemEnableConditionId [ECUC_Dem_00654]		
<b>Description</b>	<p>Defines a unique enable condition Id.</p> <p>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.</p>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemEnableConditionStatus [ECUC_Dem_00656]		
<b>Description</b>	<p>Defines the initial status for enable or disable of acceptance of event reports of a diagnostic event.</p> <p>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</p>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

#### No Included Containers

### 10.2.8 DemEnableConditionGroup

#### DemEnableConditionGroup

<b>SWS Item</b>	[ECUC_Dem_00745]
<b>Container Name</b>	DemEnableConditionGroup
<b>Description</b>	This container contains the configuration (parameters) for enable condition groups.
<b>Configuration Parameters</b>	

<b>Name</b>	DemEnableConditionRef [ECUC_Dem_00655]
<b>Description</b>	References an enable condition.
<b>Multiplicity</b>	1..255
<b>Type</b>	Reference to DemEnableCondition
<b>Post-Build Variant Multiplicity</b>	false
<b>Post-Build Variant Value</b>	false

<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

No Included Containers
------------------------

### 10.2.9 DemStorageCondition

#### DemStorageCondition

<b>SWS Item</b>	[ECUC_Dem_00728]
<b>Container Name</b>	DemStorageCondition
<b>Description</b>	This container contains the configuration (parameters) for storage conditions.
<b>Configuration Parameters</b>	

<b>Name</b>	DemStorageConditionId [ECUC_Dem_00730]		
<b>Description</b>	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.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcclIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemStorageConditionReplacementEventRef [ECUC_Dem_00893]		
<b>Description</b>	Specifies the reference to an event which is stored to event memory and supports failure analysis.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemEventParameter		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemStorageConditionStatus [ECUC_Dem_00731]		
<b>Description</b>	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		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

#### No Included Containers

## 10.2.10 DemStorageConditionGroup

### DemStorageConditionGroup

<b>SWS Item</b>	[ECUC_Dem_00773]
<b>Container Name</b>	DemStorageConditionGroup
<b>Description</b>	This container contains the configuration (parameters) for storage condition groups.
<b>Configuration Parameters</b>	

<b>Name</b>	DemStorageConditionRef [ECUC_Dem_00768]		
<b>Description</b>	References an enable condition.		
<b>Multiplicity</b>	1..255		
<b>Type</b>	Reference to DemStorageCondition		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

No Included Containers
------------------------

## 10.2.11 DemIndicator

### DemIndicator

<b>SWS Item</b>	[ECUC_Dem_00680]		
<b>Container Name</b>	DemIndicator		
<b>Description</b>	This container contains the configuration (parameters) for Indicators.		
<b>Configuration Parameters</b>			

<b>Name</b>	DemIndicatorID [ECUC_Dem_00683]		
<b>Description</b>	Unique identifier of an indicator.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b> <b>Default Value</b>	0 .. 255		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

No Included Containers
------------------------

### 10.2.12 DemNvRamBlockId

#### DemNvRamBlockId

<b>SWS Item</b>	[ECUC_Dem_00696]
<b>Container Name</b>	DemNvRamBlockId
<b>Description</b>	<p>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.</p> <p>The number of blocks which are necessary depends on the implementation and configuration (e.g. number of used event memories) of the Dem module.</p>
<b>Configuration Parameters</b>	

<b>Name</b>	DemNvRamBlockIdRef [ECUC_Dem_00697]		
<b>Description</b>	This reference contains the link to a non-volatile memory block. For post build time configurations worst case scenario shall be used.		
<b>Multiplicity</b>	1		
<b>Type</b>	Symbolic name reference to NvMBlockDescriptor		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
<b>Scope / Dependency</b>			

#### No Included Containers

### 10.2.13 DemGroupOfDTC

#### DemGroupOfDTC

<b>SWS Item</b>	[ECUC_Dem_00679]
<b>Container Name</b>	DemGroupOfDTC
<b>Description</b>	This container contains the configuration (parameters) for DTC groups.
<b>Configuration Parameters</b>	

<b>Name</b>	DemGroupDTCs [ECUC_Dem_00678]		
<b>Description</b>	DTC values of the selected group of DTC  (Range: 3 byte, 0xFFFFFFF 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 0xfffff00 to 0xffffffff.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	256 .. 16776959		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X	All Variants
<b>Scope / Dependency</b>	scope: ECU		

#### No Included Containers

### 10.2.14 DemRatio

#### DemRatio

<b>SWS Item</b>	[ECUC_Dem_00734]		
<b>Container Name</b>	DemRatio		
<b>Description</b>	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.		
<b>Configuration Parameters</b>			

<b>Name</b>	DemDiagnosticEventRef [ECUC_Dem_00735]		
<b>Description</b>	This reference contains the link to a diagnostic event.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to DemEventParameter		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X	All Variants
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU		

<b>Name</b>	DemFunctionIdRef [ECUC_Dem_00736]		
<b>Description</b>	This reference contains the link to a function identifier within the FiM which is used as a primary FID.		
<b>Multiplicity</b>	1		
<b>Type</b>	Symbolic name reference to FiMFID		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU		

<b>Name</b>	DemIUMPRDenGroup [ECUC_Dem_00838]		
<b>Description</b>	This parameter specifies the assigned denominator type which is applied in addition to the General Denominator conditions.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_IUMPR_DEN_500M ILL	Additional condition based on definition of 500miles conditions as defined for OBD2.	
	DEM_IUMPR_DEN_COL DSTART	Additional condition based on definition of "cold start" as defined for EU5+	
	DEM_IUMPR_DEN_EVAP	Additional condition based on definition of "EVAP" conditions as defined for OBD2.	
	DEM_IUMPR_DEN_NON E	No further condition. Denominator increments based on General Denominator only.	
	DEM_IUMPR_DEN_PHYS _API	Additional physical condition (component activity) computed within the SW-C and reported via Dem_RepIUMPRDenLock / Dem_RepIUMPRDenRelease.	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemIUMPRGroup [ECUC_Dem_00737]		
<b>Description</b>	This parameter specifies the assigned IUMPR group of the ratio Id.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_IUMPR_BOOSTPRS		
	DEM_IUMPR_CAT1		
	DEM_IUMPR_CAT2		
	DEM_IUMPR_EGR		
	DEM_IUMPR_EGSENSOR		
	DEM_IUMPR_EVAP		
	DEM_IUMPR_FLSYS		
	DEM_IUMPR_NMHCCAT		
	DEM_IUMPR_NOXADSORB		
	DEM_IUMPR_NOXCAT		
	DEM_IUMPR_OXS1		
	DEM_IUMPR_OXS2		
	DEM_IUMPR_PMFILTER		
	DEM_IUMPR_PRIVATE		
	DEM_IUMPR_SAIR		
	DEM_IUMPR_SECOXS1		
	DEM_IUMPR_SECOXS2		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU		

<b>Name</b>	DemRatioId [ECUC_Dem_00787]				
<b>Description</b>	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.				
<b>Multiplicity</b>	1				
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)				
<b>Range</b>	0 .. 65535				
<b>Default Value</b>					
<b>Post-Build Variant Value</b>	false				
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			

<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU
---------------------------	---

<b>Name</b>	DemRatioKind [ECUC_Dem_00741]		
<b>Description</b>	This parameter defines whether the ratio will be calculated API or observer based.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_RATIO_API	API based ratio Id	
	DEM_RATIO_OBSERVER	Observer based ratio Id	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemSecondaryFunctionIdRef [ECUC_Dem_00782]		
<b>Description</b>	This reference contains the link to a function identifier within the FiM which is used as a secondary FID.  The "primary" and all "secondary" FID inhibitions are combined by "OR".		
<b>Multiplicity</b>	0..*		
<b>Type</b>	Symbolic name reference to FiMFID		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU		

#### No Included Containers

### 10.2.15 DemPrimaryMemory

#### DemPrimaryMemory

<b>SWS Item</b>	[ECUC_Dem_00901]
<b>Container Name</b>	DemPrimaryMemory
<b>Description</b>	This container contains the primary event memory specific parameters of the Dem module.
<b>Configuration Parameters</b>	

<b>Name</b>	DemMaxNumberEventEntryPrimary [ECUC_Dem_00690]		
<b>Description</b>	Maximum number of events which can be stored in the primary memory		
<b>Multiplicity</b>	1		
<b>Type</b>	EcclIntegerParamDef		
<b>Range</b>	1 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: ECU		

#### No Included Containers

### 10.2.16 DemMirrorMemory

#### DemMirrorMemory

<b>SWS Item</b>	[ECUC_Dem_00902]
<b>Container Name</b>	DemMirrorMemory
<b>Description</b>	This container contains the mirror event memory specific parameters of the Dem module.
<b>Configuration Parameters</b>	

<b>Name</b>	DemMaxNumberEventEntryMirror [ECUC_Dem_00688]		
<b>Description</b>	Maximum number of events which can be stored in the mirror memory		
<b>Multiplicity</b>	1		
<b>Type</b>	EcclIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	

<b>Scope / Dependency</b>	scope: ECU
---------------------------	------------

<b>No Included Containers</b>
-------------------------------

## 10.2.17 DemUserDefinedMemory

### DemUserDefinedMemory

<b>SWS Item</b>	[ECUC_Dem_00910]
<b>Container Name</b>	DemUserDefinedMemory
<b>Description</b>	This container contains the user defined event memory specific parameters of the Dem module.

<b>Configuration Parameters</b>
---------------------------------

<b>Name</b>	DemMaxNumberEventEntryUserDefined [ECUC_Dem_00691]		
<b>Description</b>	Maximum number of events which can be stored in the user defined memory.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemUserDefinedMemoryIdentifier [ECUC_Dem_00903]		
<b>Description</b>	Identifier used by external tester to identify the User defined event memory.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	16 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

**No Included Containers**

### 10.2.18 DemCallbackDTCStatusChanged

#### DemCallbackDTCStatusChanged

<b>SWS Item</b>	[ECUC_Dem_00626]
<b>Container Name</b>	DemCallbackDTCStatusChanged
<b>Description</b>	<p>The presence of this container indicates, that the Dem has access to a "DTCStatusChanged" callback, which the Dem will call to notify other components about the change in the status of a DTC.</p> <p>In case there is a DemCallbackDTCStatusChangedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackDTCStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackDTCStatusChanged whose name is generated by using the unique callback-prefix followed by the event name.</p> <p>Status change notifications are supported for DTCs in primary memory only.</p>

**Configuration Parameters**

<b>Name</b>	DemCallbackDTCStatusChangedFnc [ECUC_Dem_00627]		
<b>Description</b>	<p>Function name of prototype "DTCStatusChanged".</p> <p>Note: If the parameter DemTriggerDcmReports is enabled, this parameter shall not be "Dcm_DemTriggerOnDTCStatus".</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: ECU dependency: DemTriggerDcmReport		

**No Included Containers**

### 10.2.19 DemCallbackOBDDTCStatusChanged

#### **DemCallbackOBDDTCStatusChanged**

<b>SWS Item</b>	[ECUC_Dem_00825]
<b>Container Name</b>	DemCallbackOBDDTCStatusChanged
<b>Description</b>	<p>The presence of this container indicates, that the Dem has access to a "DTCStatusChanged" callback, which the Dem will call to notify other components about the change in the status of a DTC.</p> <p>In case there is a DemCallbackDTCStatusChangedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackDTCStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackDTCStatusChanged whose name is generated by using the unique callback-prefix followed by the event name.</p> <p>Status change notifications are supported for DTCs in primary memory only.</p>

**Configuration Parameters**

<b>Name</b>	DemCallbackDTCStatusChangedFnc [ECUC_Dem_00826]		
<b>Description</b>	<p>Function name of prototype "DTCStatusChanged".</p> <p>Note: If the parameter DemTriggerDcmReports is enabled, this parameter shall not be "Dcm_DemTriggerOnDTCStatus".</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: ECU dependency: DemTriggerDcmReport		

**No Included Containers**

### 10.2.20 DemCallbackJ1939DTCStatusChanged

#### DemCallbackJ1939DTCStatusChanged

<b>SWS Item</b>	[ECUC_Dem_00823]
<b>Container Name</b>	DemCallbackJ1939DTCStatusChanged
<b>Description</b>	<p>The presence of this container indicates, that the Dem has access to a "DTCStatusChanged" callback, which the Dem will call to notify other components about the change in the status of a DTC.</p> <p>In case there is a DemCallbackDTCStatusChangedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackDTCStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackDTCStatusChanged whose name is generated by using the unique callback-prefix followed by the event name.</p> <p>Status change notifications are supported for DTCs in primary memory only.</p>

**Configuration Parameters**

<b>Name</b>	DemCallbackDTCStatusChangedFnc [ECUC_Dem_00824]		
<b>Description</b>	<p>Function name of prototype "DTCStatusChanged".</p> <p>Note: If the parameter DemTriggerDcmReports is enabled, this parameter shall not be "Dcm_DemTriggerOnDTCStatus".</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: ECU dependency: DemTriggerDcmReport		

**No Included Containers**

### 10.2.21 DemConfigSet

#### **DemConfigSet**

<b>SWS Item</b>	[ECUC_Dem_00634]
<b>Container Name</b>	DemConfigSet
<b>Description</b>	This container contains the configuration parameters and sub containers of the Dem module supporting multiple configuration sets.

**Configuration Parameters**

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
DemComponent	0..*	This container configures the monitored components and system dependencies.
DemDTC	0..65535	This container contains the configuration (parameters) for DemUdsDTC.
DemDTCAttributes	0..65535	This container contains the configuration (parameters) for DemDTCAttributes.
DemDebounceCounter BasedClass	0..65535	This container contains the configuration of Debounce Counter Based Class
DemDebounceTimeBase Class	0..65535	This container contains the configuration of Debounce Counter Based Class
DemDtrs	0..1	This container holds the configuration of DTRs collection.
DemEventParameter	1..65535	This container contains the configuration (parameters) for events.
DemJ1939Node	0..255	Contains the parameters for the support of a logical J1939 node.
DemObdDTC	0..65535	This container contains the configuration (parameters) for DemObdDTC.
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.

### 10.2.22 DemComponent

#### **DemComponent**

<b>SWS Item</b>	[ECUC_Dem_00904]
<b>Container Name</b>	DemComponent
<b>Description</b>	This container configures the monitored components and system dependencies.

**Configuration Parameters**

<b>Name</b>	DemComponentFailedCallbackFnc [ECUC_Dem_00906]		
<b>Description</b>	Specifies the function to be called on component failed status changes.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemComponentIgnoresPriority [ECUC_Dem_00905]		
<b>Description</b>	This configuration switch defines, whether the priority of events at this component shall be ignored.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucBooleanParamDef		
	false		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemImmediateChildComponentRef [ECUC_Dem_00907]		
<b>Description</b>	Reference to all immediate children of the current component.		
<b>Multiplicity</b>	0..*		
<b>Type</b>	Reference to DemComponent		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		

<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

No Included Containers
------------------------

### 10.2.23 DemOBDTC

#### DemOBDTC

<b>SWS Item</b>	[ECUC_Dem_00884]
<b>Container Name</b>	DemOBDTC
<b>Description</b>	This container contains the configuration (parameters) for DemOBDTC.
<b>Configuration Parameters</b>	

<b>Name</b>	DemConsiderPtoStatus [ECUC_Dem_00602]		
<b>Description</b>	This parameter is TRUE, when the event is affected by the Dem PTO handling.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemDtcValue [ECUC_Dem_00885]				
<b>Description</b>	Unique Diagnostic Trouble Code value for OBD				
<b>Multiplicity</b>	0..1				
<b>Type</b>	EcucIntegerParamDef				
<b>Range</b>	1 .. 65535				
<b>Default Value</b>					
<b>Post-Build Variant Multiplicity</b>	true				

<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	—	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	—	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemEventOBDReadinessGroup [ECUC_Dem_00755]	
<b>Description</b>	This parameter specifies the Event OBD Readiness group for PID \$01 and PID \$41 computation. This parameter is only applicable for emission-related ECUs.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	EcucEnumerationParamDef	
<b>Range</b>	DEM_OBD_RDY_AC	A/C system component - spark
	DEM_OBD_RDY_BOOST_PR	Boost Pressure System - compr.
	DEM_OBD_RDY_CAT	Catalyst - spark
	DEM_OBD_RDY_CMPPRC_MPT	Comprehensive component - spark, compr.
	DEM_OBD_RDY_EGSEN_S	Exhaust Gas Sensor - compr.
	DEM_OBD_RDY_ERG	EGR system - spark, compr.
	DEM_OBD_RDY_EVAP	Evaporative system - spark
	DEM_OBD_RDY_FLSYS	Fuel system - spark, compr.
	DEM_OBD_RDY_FLSYS_NONCONT	Non Contious Fuel system - spark, compr
	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_NOXCAT	NOx Catalyst - compr.
<b>Post-Build Variant Multiplicity</b>	DEM_OBD_RDY_O2SEN_S	Oxygen sensor - spark
	DEM_OBD_RDY_O2SEN_SHT	Oxygen sensor heater - spark
<b>Post-Build Variant Value</b>	DEM_OBD_RDY_PMFLT	Particle Matters Filter - compr.
	DEM_OBD_RDY_SECAIR	Secondary air system - spark

<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU		

<b>Name</b>	DemJ1939DTCValue [ECUC_Dem_00892]		
<b>Description</b>	Unique Diagnostic Trouble Code value for J1939 (consiting of SPN and FMI)		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	1 .. 16777214		
<b>Default Value</b>			
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	—	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	—	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU		

#### No Included Containers

## 10.2.24 DemDTC

### DemDTC

<b>SWS Item</b>	[ECUC_Dem_00886]
<b>Container Name</b>	DemDTC
<b>Description</b>	This container contains the configuration (parameters) for DemUdsDTC.
<b>Configuration Parameters</b>	

<b>Name</b>	DemDTCAttributesRef [ECUC_Dem_00642]		
<b>Description</b>	This parameter defines the DTC Attributes associated with the DemDTC.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to DemDTCAttributes		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
<b>Scope / Dependency</b>			

<b>Name</b>	DemDTCFunctionalUnit [ECUC_Dem_00643]		
<b>Description</b>	DTCFunctionalUnit 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.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
<b>Value Configuration Class</b>	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDTCSeverity [ECUC_Dem_00645]		
<b>Description</b>	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.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_SEVERITY_CHECK_AT_NEXT_HALT		Check at next halt

<b>Post-Build Variant Multiplicity</b>	DEM_SEVERITY_CHECK_IMMEDIATELY	Check immediately	
	DEM_SEVERITY_MAINTENANCE_ONLY	Maintenance required	
	DEM_SEVERITY_NO_SEVERITY false	No severity information available	
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDtcValue [ECUC_Dem_00887]				
<b>Description</b>	Unique Diagnostic Trouble Code value for UDS  (Range: 0x000000 and 0xFFFF are reserved for DTC groups by ISO 14229-1)				
<b>Multiplicity</b>					
<b>Type</b>	EcucIntegerParamDef				
<b>Range</b> <b>Default Value</b>	1 .. 16777214				
<b>Post-Build Variant Multiplicity</b>	true				
<b>Post-Build Variant Value</b>	true				
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL		
	<b>Link time</b>	—			
	<b>Post-build time</b>	X	VARIANT-POST-BUILD		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL		
	<b>Link time</b>	—			
	<b>Post-build time</b>	X	VARIANT-POST-BUILD		
<b>Scope / Dependency</b>	scope: ECU				

<b>Name</b>	DemObdDTCRef [ECUC_Dem_00889]		
<b>Description</b>	This parameter defines the OBD DTC configuration associated with the DemDTC.  It is allowed to have events without a OBD DTC.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemObdDTC		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>			

<b>Name</b>	DemWWHOBDDTCClass [ECUC_Dem_00912]		
<b>Description</b>	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.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_DTC_WWHOBD_C_LASS_A	X	All Variants
	DEM_DTC_WWHOBD_C_LASS_B1	X	All Variants
	DEM_DTC_WWHOBD_C_LASS_B2	X	All Variants
	DEM_DTC_WWHOBD_C_LASS_C	X	All Variants
	DEM_DTC_WWHOBD_C_LASS_NOCLASS	X	No Class information
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

**No Included Containers**
**10.2.25 DemJ1939Node**
**DemJ1939Node**

<b>SWS Item</b>	[ECUC_Dem_00817]		
<b>Container Name</b>	DemJ1939Node		
<b>Description</b>	Contains the parameters for the support of a logical J1939 node.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	—	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Configuration Parameters</b>			

<b>Name</b>	DemJ1939NmNodeRef [ECUC_Dem_00818]		
<b>Description</b>	Reference to the corresponding J1939Nm node.		
<b>Multiplicity</b>	1..255		
<b>Type</b>	Symbolic name reference to J1939NmNode		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	—	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	—	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU		

**No Included Containers**
**10.2.26 DemPidClass**
**DemPidClass**

<b>SWS Item</b>	[ECUC_Dem_00729]
<b>Container Name</b>	DemPidClass

<b>Description</b>	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.		
<b>Configuration Parameters</b>			

<b>Name</b>	DemPidIdentifier [ECUC_Dem_00705]				
<b>Description</b>	identifier of the PID				
<b>Multiplicity</b>	1				
<b>Type</b>	EcucIntegerParamDef				
<b>Range</b>	0 .. 255				
<b>Default Value</b>					
<b>Post-Build Variant Value</b>	true				
<b>Value Configuration Class</b>	Pre-compile time	X	VARIANT-PRE-COMPIL		
	Link time	-			
	Post-build time	X	VARIANT-POST-BUILD		
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU				

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
DemPidDataElement	1..255	This container contains the different data elements contained in the specific PID.

## 10.2.27 DemPidDataElement

### DemPidDataElement

<b>SWS Item</b>	[ECUC_Dem_00896]		
<b>Container Name</b>	DemPidDataElement		
<b>Description</b>	This container contains the different data elements contained in the specific PID.		
<b>Configuration Parameters</b>			

<b>Name</b>	DemPidDataElementClassRef [ECUC_Dem_00733]		
<b>Description</b>	This reference contains the link to a data element class.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to DemDataElementClass		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	Pre-compile time	X	VARIANT-PRE-COMPIL
	Link time	-	
	Post-build time	X	VARIANT-POST-BUILD

<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU
---------------------------	--

**No Included Containers**

### 10.2.28 DemDTCAttributes

#### DemDTCAttributes

<b>SWS Item</b>	[ECUC_Dem_00641]
<b>Container Name</b>	DemDTCAttributes
<b>Description</b>	This container contains the configuration (parameters) for DemDTCAttributes.
<b>Configuration Parameters</b>	

<b>Name</b>	DemAgingAllowed [ECUC_Dem_00622]		
<b>Description</b>	Switch to allow aging/unlearning of the event or not.  true: aging allowed false: aging not allowed		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemAgingCycleCounterThreshold [ECUC_Dem_00623]				
<b>Description</b>	Number of aging cycles needed to unlearn/delete the event.				
<b>Multiplicity</b>	0..1				
<b>Type</b>	EcucIntegerParamDef				
<b>Range</b>	1 .. 256				
<b>Default Value</b>					
<b>Post-Build Variant Multiplicity</b>	false				
<b>Post-Build Variant Value</b>	false				
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemAgingAllowed		

<b>Name</b>	DemAgingCycleCounterThresholdForTFSLC [ECUC_Dem_00897]		
<b>Description</b>	Number of aging cycles needed to reset the testFailedSinceLastClear Bit.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	1 .. 256		
<b>Default Value</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemStatusBitHandlingTestFailedSinceLastClear == DEM_STATUS_BIT_AGING_AND_DISPLACEMENT		

<b>Name</b>	DemAgingCycleRef [ECUC_Dem_00624]		
<b>Description</b>	Reference to the cycle which is triggering the aging of the event.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemOperationCycle		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemAgingAllowed, DemOperationCycleRef, DemAgingCycleCounterProcessing		

<b>Name</b>	DemDTCPriority [ECUC_Dem_00662]		
<b>Description</b>	Priority of the event/dtc, in view of full event buffer. A lower value means higher priority.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	1 .. 256		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDTCSignificance [ECUC_Dem_00779]		
<b>Description</b>	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.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_EVENT_SIGNIFICANCE_FAULT	failure, which affects the component/ECU itself	
<b>Post-Build Variant Multiplicity</b>	DEM_EVENT_SIGNIFICANCE_OCCURRENCE	issue, which indicates additional information concerning insufficient system behavior	
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemEventMemoryEntryFdcThresholdStorageValue [ECUC_Dem_00798]		
<b>Description</b>	Threshold to allocate an event memory entry and to capture the Freeze Frame.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	1 .. 126		
<b>Default Value</b>			

<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local dependency: DemFreezeFrameRecordTrigger, DemExtendedDataRecordTrigger, DemEventMemoryEntryStorageTrigger = DEM_STORAGE_ON_FDC_THRESHOLD		

<b>Name</b>	DemExtendedDataClassRef [ECUC_Dem_00667]		
<b>Description</b>	This reference defines the link to an extended data class sampler.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemExtendedDataClass		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>			

<b>Name</b>	DemFreezeFrameClassRef [ECUC_Dem_00674]		
<b>Description</b>	These references define the links to a freeze frame class sampler.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemFreezeFrameClass		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>			

<b>Name</b>	DemFreezeFrameRecNumClassRef [ECUC_Dem_00776]		
<b>Description</b>	<p>This parameter defines the list of dedicated freeze frame record numbers associated with the diagnostic event. These record numbers are assigned to the freeze frame records (instead of calculated record numbers).</p> <p>This parameter is only required for configured record numeration (refer to DemTypeOfFreezeFrameRecordNumeration).</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemFreezeFrameRecNumClass		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemTypeOfFreezeFrameRecordNumeration = DEM_FF_RECNUM_CONFIGURED		

<b>Name</b>	DemImmediateNvStorage [ECUC_Dem_00739]		
<b>Description</b>	<p>Switch to enable immediate storage triggering of an according event memory entry persistently to NVRAM.</p> <p>true: immediate non-volatile storage triggering enabled false: immediate non-volatile storage triggering disabled</p>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemJ1939DTC_J1939NodeRef [ECUC_Dem_00819]		
<b>Description</b>	Reference to a J1939 Node		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemJ1939Node		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	—	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	—	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>			

<b>Name</b>	DemJ1939ExpandedFreezeFrameClassRef [ECUC_Dem_00834]		
<b>Description</b>	These references define the links to a J1939 freeze frame class sampler.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemJ1939FreezeFrameClass		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>			

<b>Name</b>	DemJ1939FreezeFrameClassRef [ECUC_Dem_00835]		
<b>Description</b>	These references define the links to a J1939 freeze frame class sampler.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemJ1939FreezeFrameClass		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>			

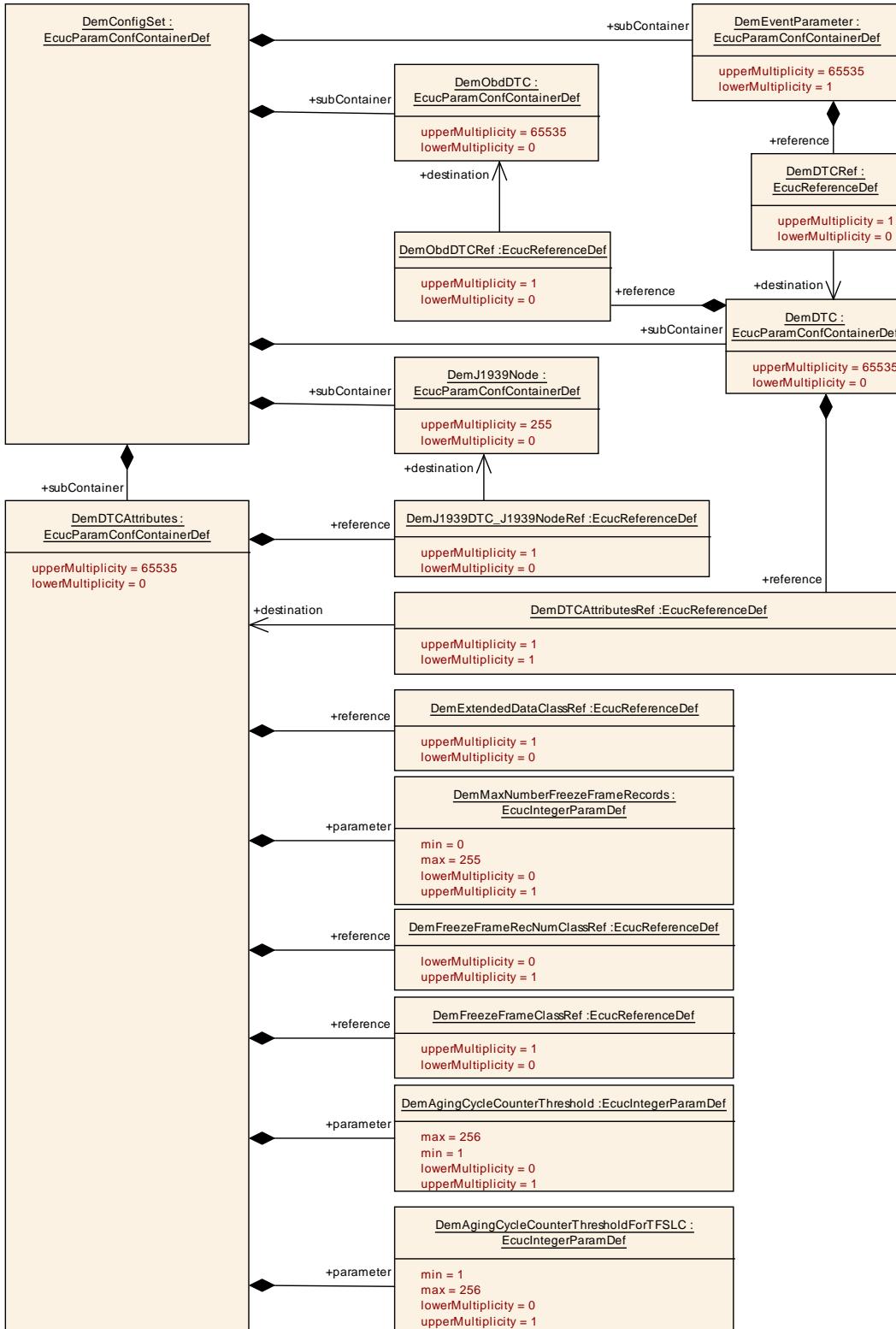
<b>Name</b>	DemMaxNumberFreezeFrameRecords [ECUC_Dem_00605]		
<b>Description</b>	<p>This parameter defines the number of according freeze frame records, which can maximal be stored for this event. Therefore all these freeze frame records have the same freeze frame class.</p> <p>This parameter is only required for calculated record numeration (refer to DemTypeOfFreezeFrameRecordNumeration).</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemTypeOfFreezeFrameRecordNumeration = DEM_FF_RECNUM_CALCULATED		

<b>Name</b>	DemMemoryDestinationRef [ECUC_Dem_00890]		
<b>Description</b>	The event destination assigns events to none, one or two origins. If no event destination is assigned to a specific event, the event is handled internally and is not visible externally to the Dcm. If more than one event destination is assigned to a specific event, the event can be present in the corresponding origins.		
<b>Multiplicity</b>	0..2		
<b>Type</b>	Choice Reference to [DemMirrorMemory,DemPrimaryMemory,DemUserDefinedMemory]		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	

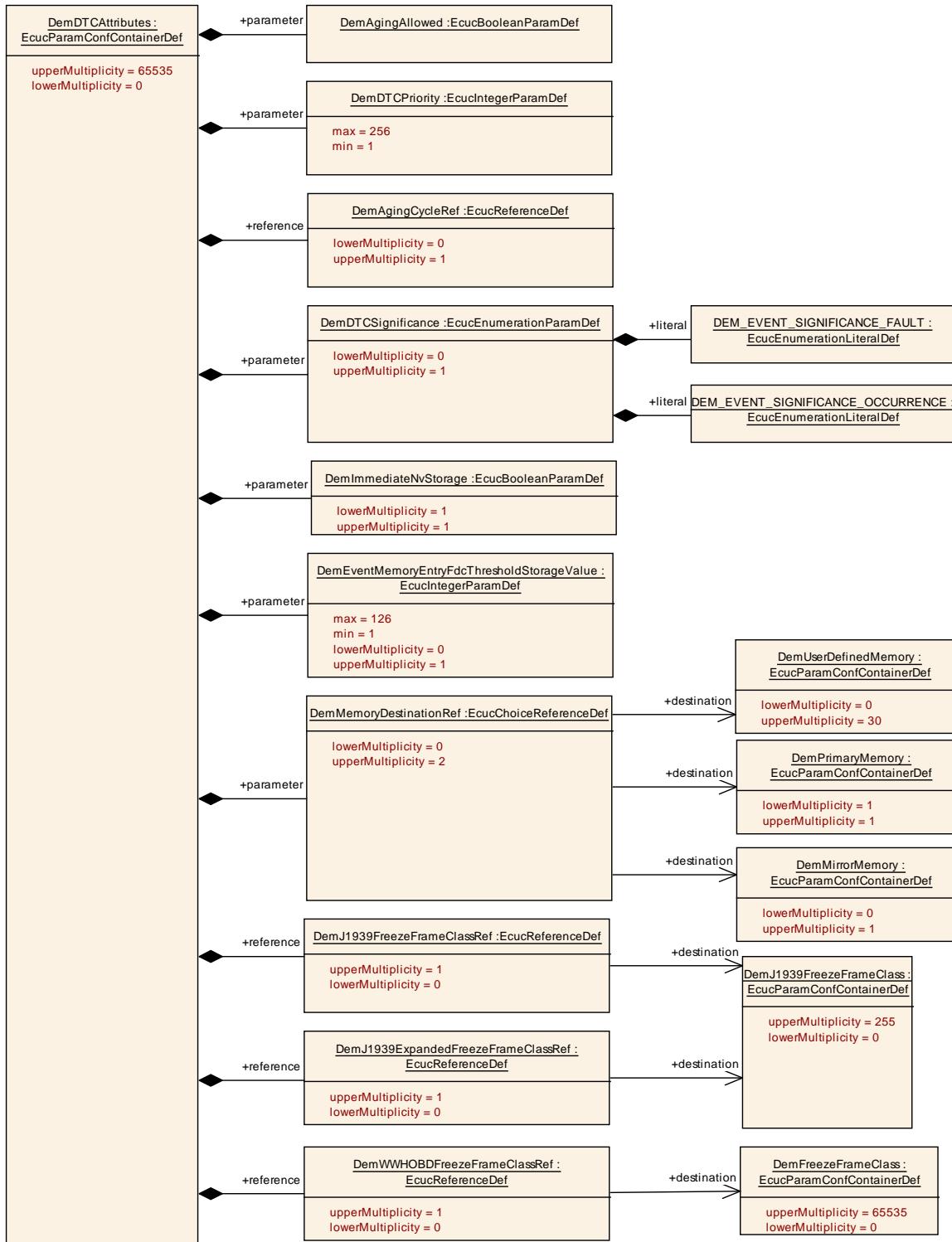
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemWWHOBDFreezeFrameClassRef [ECUC_Dem_00911]		
<b>Description</b>	This reference defines the link to a WWH-OBD freeze frame class sampler.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemFreezeFrameClass		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

**No Included Containers**



**Figure 10.4: DTC attribute configuration (part I)**


**Figure 10.5: DTC attribute configuration (part II)**

### 10.2.29 DemEventParameter

#### DemEventParameter

<b>SWS Item</b>	[ECUC_Dem_00661]
<b>Container Name</b>	DemEventParameter
<b>Description</b>	This container contains the configuration (parameters) for events.
<b>Configuration Parameters</b>	

<b>Name</b>	DemComponentClassRef [ECUC_Dem_00908]		
<b>Description</b>	Reference to the monitored component.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemComponent		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemComponentPriority [ECUC_Dem_00909]		
<b>Description</b>	Specifies the priority within the component. A lower value means higher priority.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range Default Value</b>	0 .. 255		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemDTCRef [ECUC_Dem_00888]		
<b>Description</b>	<p>This parameter defines the DTC configuration (typically Uds) associated with the diagnostic event.</p> <p>It is allowed to have events without a DTC (e.g. for ECU-internal events triggering safety reactions without being reported via diagnostic communication). The same DemDTCAttributes can be used from several events, to combine these (refer to chapter "Combination of diagnostic event").</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemDTC		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>			

<b>Name</b>	DemEnableConditionGroupRef [ECUC_Dem_00746]		
<b>Description</b>	References an enable condition group.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemEnableConditionGroup		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemEventAvailable [ECUC_Dem_00792]		
<b>Description</b>	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		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			

<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	—	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemEventFailureCycleCounterThreshold [ECUC_Dem_00753]		
<b>Description</b>	Defines the number of failure cycles for the event based fault confirmation. If this parameter is enabled, fault confirmation of the event is enabled accordingly.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default Value</b>	0		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	—	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	—	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemEventId [ECUC_Dem_00659]		
<b>Description</b>	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.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	1 .. 65535		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemEventKind [ECUC_Dem_00660]						
<b>Description</b>	This parameter is used to distinguish between SW-C and BSW events. SW-C events are reported by Dem_SetEventStatus API and BSW events are reported by Dem_ReportErrorStatus API.						
<b>Multiplicity</b>	1						
<b>Type</b>	EcucEnumerationParamDef						
<b>Range</b>	<table border="1"> <tr> <td>DEM_EVENT_KIND_BSW</td> <td>The event is assigned to a BSW module</td> </tr> <tr> <td>DEM_EVENT_KIND_SWC</td> <td>The event is assigned to a SW-C</td> </tr> </table>			DEM_EVENT_KIND_BSW	The event is assigned to a BSW module	DEM_EVENT_KIND_SWC	The event is assigned to a SW-C
DEM_EVENT_KIND_BSW	The event is assigned to a BSW module						
DEM_EVENT_KIND_SWC	The event is assigned to a SW-C						
<b>Post-Build Variant Value</b>	false						
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants				
	<b>Link time</b>	—					
	<b>Post-build time</b>	—					
<b>Scope / Dependency</b>							

<b>Name</b>	DemFFPrestorageSupported [ECUC_Dem_00671]		
<b>Description</b>	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.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemOBDGroupingAssociativeEventsRef [ECUC_Dem_00839]		
<b>Description</b>	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 referenced to only one group of associate events.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemEventParameter		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemOperationCycleRef [ECUC_Dem_00702]		
<b>Description</b>	Kind of operation cycle for the event (e.g. power cycle, driving cycle, ...)		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to DemOperationCycle		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemReportBehavior [ECUC_Dem_00894]				
<b>Description</b>	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.				
<b>Multiplicity</b>	0..1				
<b>Type</b>	EcucEnumerationParamDef				
<b>Range</b>	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().			
<b>Default Value</b>	REPORT_BEFORE_INIT				
<b>Post-Build Variant Multiplicity</b>	false				
<b>Post-Build Variant Value</b>	false				
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Scope / Dependency</b>	scope: local dependency: DemEventKind == DEM_EVENT_KIND_BSW				

<b>Name</b>	DemStorageConditionGroupRef [ECUC_Dem_00769]		
<b>Description</b>	References a storage condition group.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemStorageConditionGroup		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

Included Containers			
Container Name	Multiplicity	Scope / Dependency	
DemCallbackClearEventAllowed	0..1	<p>The presence of this container indicates that the Dem has access to a "ClearEventAllowed" callback.</p> <p>In case there is a DemCallbackClearEventAllowedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>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.</p>	
DemCallbackEventDataChanged	0..1	<p>The presence of this container indicates that the Dem has access to an "EventDataChanged" callback.</p> <p>In case there is a DemCallbackEventDataChangedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>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.</p>	

DemCallbackEventStatusChanged	0..*	<p>The presence of this container indicates, that the Dem has access to an "EventStatusChanged" callback, which the Dem will call to notify other components about the change in the status of an event.</p> <p>In case there is a DemCallbackEvenStatusChangedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackEvenStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackEventStatusChanged, whose name is generated by using the unique callback-prefix followed by the event name.</p>
DemCallbackInitMForE	0..1	<p>The presence of this container indicates, that the Dem has access to an "InitMonitorForEvent" callback, which the Dem will call to initialize a monitor.</p> <p>In case the container has a DemCallbackInitMForEFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackInitMForEFnc, the Dem will have an R-Port requiring the interface CallbackInitMonitorForEvent, whose name is generated by using the unique callback-prefix followed by the event name.</p>
DemDebounceAlgorithmClass	1	Debounce algorithm class: counter based, time based, or monitor internal.
DemIndicatorAttribute	0..255	This container contains the event specific configuration of Indicators.

### 10.2.30 DemIndicatorAttribute

#### DemIndicatorAttribute

SWS Item	[ECUC_Dem_00681]
Container Name	DemIndicatorAttribute
Description	This container contains the event specific configuration of Indicators.
Configuration Parameters	

Name	DemIndicatorBehaviour [ECUC_Dem_00682]	
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.

<b>Post-Build Variant Value</b>	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 event has status FAILED	
	DEM_INDICATOR_FAST_FLASH	Flash Indicator Lamp should be set to 'Fast Flash'	
	DEM_INDICATOR_SLOW_FLASH true	Flash Indicator Lamp should be set to 'Slow Flash'	
<b>Value Configuration Class</b>	Pre-compile time	X	VARIANT-PRE-COMPIL
	Link time	—	
	Post-build time	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemIndicatorFailureCycleCounterThreshold [ECUC_Dem_00750]				
<b>Description</b>	Defines the number of failure cycles for the WarningIndicatorOnCriteria.				
<b>Multiplicity</b>	0..1				
<b>Type</b>	EcclIntegerParamDef				
<b>Range</b>	0 .. 255				
<b>Default Value</b>					
<b>Post-Build Variant Multiplicity</b>	false				
<b>Post-Build Variant Value</b>	false				
<b>Multiplicity Configuration Class</b>	Pre-compile time	X	All Variants		
	Link time	—			
	Post-build time	—			
<b>Value Configuration Class</b>	Pre-compile time	X	All Variants		
	Link time	—			
	Post-build time	—			
<b>Scope / Dependency</b>	scope: ECU				

<b>Name</b>	DemIndicatorHealingCycleCounterThreshold [ECUC_Dem_00748]		
<b>Description</b>	Defines the number of healing cycles for the WarningIndicatorOffCriteria.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcclIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemIndicatorRef [ECUC_Dem_00687]		
<b>Description</b>	Reference to the used indicator.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to DemIndicator		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

#### No Included Containers

### 10.2.31 DemDebounceAlgorithmClass

#### DemDebounceAlgorithmClass

<b>SWS Item</b>	[ECUC_Dem_00604]		
<b>Container Name</b>	DemDebounceAlgorithmClass		
<b>Description</b>	Debounce algorithm class: counter based, time based, or monitor internal.		
<b>Configuration Parameters</b>			

<b>Container Choices</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
DemDebounceCounter Based	0..1	This container contains the configuration (parameters) for counter based debouncing.
DemDebounceMonitor Internal	0..1	This container contains the configuration (parameters) for monitor internal debouncing.
DemDebounceTimeBase	0..1	This container contains the configuration (parameters) for time based debouncing.

### 10.2.32 DemDebounceCounterBased

#### DemDebounceCounterBased

<b>SWS Item</b>	[ECUC_Dem_00711]
-----------------	------------------

<b>Container Name</b>	DemDebounceCounterBased
<b>Description</b>	This container contains the configuration (parameters) for counter based debouncing.
<b>Configuration Parameters</b>	

<b>Name</b>	DemDebounceCounterBasedClassRef [ECUC_Dem_00883]		
<b>Description</b>	This reference selects the DemDebounceCounterBasedClass applied for the debouncing of the DemEventParameter.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to DemDebounceCounterBasedClass		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>			

#### No Included Containers

### 10.2.33 DemDebounceCounterBasedClass

#### DemDebounceCounterBasedClass

<b>SWS Item</b>	[ECUC_Dem_00881]
<b>Container Name</b>	DemDebounceCounterBasedClass
<b>Description</b>	This container contains the configuration of Debounce Counter Based Class
<b>Configuration Parameters</b>	

<b>Name</b>	DemDebounceBehavior [ECUC_Dem_00786]		
<b>Description</b>	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.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_DEBOUNCE_FREEZE  DEM_DEBOUNCE_RESET	The event debounce counter will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and ControlDTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. SetEventStatus).	
<b>Post-Build Variant Value</b>			
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemDebounceCounterDecrementStepSize [ECUC_Dem_00635]		
<b>Description</b>	Defines the step size for decrementation of the internal debounce counter (PREPASSED).		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	1 .. 32768		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDebounceCounterFailedThreshold [ECUC_Dem_00618]		
<b>Description</b>	Defines the value of the internal debounce counter, which indicates the failed status.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	1 .. 32767		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDebounceCounterIncrementStepSize [ECUC_Dem_00637]		
<b>Description</b>	Defines the step size for incrementation of the internal debounce counter (PREFAILED).		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	1 .. 32767		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDebounceCounterJumpDown [ECUC_Dem_00685]		
<b>Description</b>	Switch for the activation of Jump-Down.  true: Jump-Down activated false: Jump-Down deactivated		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDebounceCounterJumpDownValue [ECUC_Dem_00638]		
<b>Description</b>	Jump-Down value of the internal debounce counter which is taken as initialization value for the counter when the respective step-down occurs.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	-32768 .. 32767		
<b>Default Value</b>	0		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemDebounceCounterJumpDown		

<b>Name</b>	DemDebounceCounterJumpUp [ECUC_Dem_00686]		
<b>Description</b>	Switch for the activation of Jump-Up.  true: Jump-Up activated false: Jump-Up deactivated		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDebounceCounterJumpUpValue [ECUC_Dem_00639]		
<b>Description</b>	Jump-Up value of the internal debounce counter which is taken as initialization value for the counter when the respective step-up occurs.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	-32768 .. 32767		
<b>Default Value</b>	0		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemDebounceCounterJumpUp		

<b>Name</b>	DemDebounceCounterPassedThreshold [ECUC_Dem_00636]				
<b>Description</b>	Defines the value of the internal debounce counter, which indicates the passed status.				
<b>Multiplicity</b>	1				
<b>Type</b>	EcucIntegerParamDef				
<b>Range</b>	-32768 .. -1				
<b>Default Value</b>					
<b>Post-Build Variant Value</b>	false				
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Scope / Dependency</b>	scope: ECU				

<b>Name</b>	DemDebounceCounterStorage [ECUC_Dem_00791]		
<b>Description</b>	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		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local dependency: DemOperationCycleStatusStorage = True		

#### No Included Containers

### 10.2.34 DemDebounceTimeBase

#### DemDebounceTimeBase

<b>SWS Item</b>	[ECUC_Dem_00713]
<b>Container Name</b>	DemDebounceTimeBase
<b>Description</b>	This container contains the configuration (parameters) for time based debouncing.
<b>Configuration Parameters</b>	

<b>Name</b>	DemDebounceTimeBaseRef [ECUC_Dem_00891]		
<b>Description</b>	This reference selects the DemDebounceTimeBaseClass applied for the debouncing of the DemEventParameter.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to DemDebounceTimeBaseClass		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
<b>Scope / Dependency</b>			

No Included Containers
------------------------

## 10.2.35 DemDebounceTimeBaseClass

### DemDebounceTimeBaseClass

<b>SWS Item</b>	[ECUC_Dem_00882]
<b>Container Name</b>	DemDebounceTimeBaseClass
<b>Description</b>	This container contains the configuration of Debounce Counter Based Class
<b>Configuration Parameters</b>	

<b>Name</b>	DemDebounceBehavior [ECUC_Dem_00789]		
<b>Description</b>	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.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_DEBOUNCE_FREE ZE	The event debounce timer will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and ControlDTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. SetEventStatus).	

	DEM_DEBOUNCE_RESET	The event debounce timer will be reset to initial value if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. The qualification of the event will be restarted with the next valid event report.	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemDebounceTimeFailedThreshold [ECUC_Dem_00716]		
<b>Description</b>	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.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	0.001 .. 3600		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDebounceTimePassedThreshold [ECUC_Dem_00717]		
<b>Description</b>	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.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	0.001 .. 3600		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

**No Included Containers**

### 10.2.36 DemDebounceMonitorInternal

#### DemDebounceMonitorInternal

<b>SWS Item</b>	[ECUC_Dem_00712]
<b>Container Name</b>	DemDebounceMonitorInternal
<b>Description</b>	This container contains the configuration (parameters) for monitor internal debouncing.
<b>Configuration Parameters</b>	

**Included Containers**

<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
DemCallbackGetFDC	0..1	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.

### 10.2.37 DemCallbackGetFDC

#### DemCallbackGetFDC

<b>SWS Item</b>	[ECUC_Dem_00630]
<b>Container Name</b>	DemCallbackGetFDC
<b>Description</b>	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.
<b>Configuration Parameters</b>	

<b>Name</b>	DemCallbackGetFDCFnc [ECUC_Dem_00631]		
<b>Description</b>	This parameter defines the name of the function that the Dem will call to retrieve the fault-detection counter value from a complex driver.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

#### No Included Containers

### 10.2.38 DemCallbackClearEventAllowed

#### DemCallbackClearEventAllowed

<b>SWS Item</b>	[ECUC_Dem_00607]
<b>Container Name</b>	DemCallbackClearEventAllowed
<b>Description</b>	<p>The presence of this container indicates that the Dem has access to a "ClearEventAllowed" callback.</p> <p>In case there is a DemCallbackClearEventAllowedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>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.</p>
<b>Configuration Parameters</b>	

<b>Name</b>	DemCallbackClearEventAllowedFnc [ECUC_Dem_00609]
<b>Description</b>	Function name of prototype "ClearEventAllowed".
<b>Multiplicity</b>	0..1
<b>Type</b>	EcucFunctionNameDef
<b>Default Value</b>	
<b>Regular Expression</b>	

<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemClearEventAllowedBehavior [ECUC_Dem_00788]				
<b>Description</b>	Defines the resulting UDS status byte for the related event, which must not be cleared according to the ClearEventAllowed callback.				
<b>Multiplicity</b>	1				
<b>Type</b>	EcucEnumerationParamDef				
<b>Range</b>	DEM_NO_STATUS_BYTE_CHANGE	The event status byte keeps unchanged.			
	DEM_ONLY_THIS_CYCLE_AND_READINESS	The <...>ThisOperationCycle and readiness bits of the event status byte are reset.			
<b>Default Value</b>	DEM_NO_STATUS_BYTE_CHANGE				
<b>Post-Build Variant Value</b>	false				
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Scope / Dependency</b>	scope: local				

#### No Included Containers

### 10.2.39 DemCallbackEventDataChanged

#### DemCallbackEventDataChanged

<b>SWS Item</b>	[ECUC_Dem_00606]
<b>Container Name</b>	DemCallbackEventDataChanged

<b>Description</b>	<p>The presence of this container indicates that the Dem has access to an "EventDataChanged" callback.</p> <p>In case there is a DemCallbackEventDataChangedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>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.</p>
<b>Configuration Parameters</b>	

<b>Name</b>	DemCallbackEventDataChangedFnc [ECUC_Dem_00608]		
<b>Description</b>	Function name of prototype "EventDataChanged"		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

#### No Included Containers

### 10.2.40 DemCallbackEventStatusChanged

#### DemCallbackEventStatusChanged

<b>SWS Item</b>	[ECUC_Dem_00628]
<b>Container Name</b>	DemCallbackEventStatusChanged

<b>Description</b>	<p>The presence of this container indicates, that the Dem has access to an "EventStatusChanged" callback, which the Dem will call to notify other components about the change in the status of an event.</p> <p>In case there is a DemCallbackEvenStatusChangedFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackEvenStatusChangedFnc, the Dem will have an R-Port requiring the interface CallbackEventStatusChanged, whose name is generated by using the unique callback-prefix followed by the event name.</p>
--------------------	---

#### Configuration Parameters

<b>Name</b>	DemCallbackEventStatusChangedFnc [ECUC_Dem_00629]		
<b>Description</b>	Function name of prototype "EventStatusChanged"		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X	All Variants
		-	
		-	
<b>Value Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X	All Variants
		-	
		-	
<b>Scope / Dependency</b>	scope: ECU		

#### No Included Containers

### 10.2.41 DemCallbackInitMForE

#### DemCallbackInitMForE

<b>SWS Item</b>	[ECUC_Dem_00632]
<b>Container Name</b>	DemCallbackInitMForE

<b>Description</b>	<p>The presence of this container indicates, that the Dem has access to an "InitMonitorForEvent" callback, which the Dem will call to initialize a monitor.</p> <p>In case the container has a DemCallbackInitMForEFnc, this parameter defines the name of the function that the Dem will call.</p> <p>In case there is no DemCallbackInitMForEFnc, the Dem will have an R-Port requiring the interface CallbackInitMonitorForEvent, whose name is generated by using the unique callback-prefix followed by the event name.</p>
--------------------	--

#### Configuration Parameters

<b>Name</b>	DemCallbackInitMForEFnc [ECUC_Dem_00601]		
<b>Description</b>	Function name of prototype "InitMonitorForEvent".		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X	All Variants
<b>Value Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X	All Variants
<b>Scope / Dependency</b>			

#### No Included Containers

### 10.2.42 DemDtrs

#### DemDtrs

<b>SWS Item</b>	[ECUC_Dem_00913]
<b>Container Name</b>	DemDtrs
<b>Description</b>	This container holds the configuration of DTRs collection.
<b>Configuration Parameters</b>	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DemDtr	0..65535	This container holds the configuration of one individual DTR.

### 10.2.43 DemDtr

#### DemDtr

SWS Item	[ECUC_Dem_00806]
Container Name	DemDtr
Description	This container holds the configuration of one individual DTR.
Configuration Parameters	

Name	DemDtrCompuDenominator0 [ECUC_Dem_00815]		
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			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

Name	DemDtrCompuNumerator0 [ECUC_Dem_00813]		
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			
Post-Build Variant Value	false		

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDtrCompuNumerator1 [ECUC_Dem_00814]		
<b>Description</b>	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.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	-INF .. INF		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemDtrEventRef [ECUC_Dem_00808]		
<b>Description</b>	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.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to DemEventParameter		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemDtrId [ECUC_Dem_00807]		
<b>Description</b>	The index identifier value assigned to this DTR. The value is generated during the Dem configuration process.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 65535		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemDtrMid [ECUC_Dem_00809]		
<b>Description</b>	The OBDMID of the DTR.  The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, 0xE0 are reserved.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	—	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemDtrTid [ECUC_Dem_00810]		
<b>Description</b>	The OBDTID of the DTR.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	—	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD

<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU
---------------------------	--

<b>Name</b>	DemDtrUasid [ECUC_Dem_00811]		
<b>Description</b>	The UaSId the DTR data shall be scaled to, and reported together with the rescaled DTR data.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPIL
	<b>Link time</b>	-	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU		

<b>Name</b>	DemDtrUpdateKind [ECUC_Dem_00812]		
<b>Description</b>	Update conditions applied by the Dem to reports of DTR values. Only supported if a related Event is configured. If no related Event is configured, the Dem behaves as if DemDtrUpdateKind is configured to "DEM_DTR_UPDATE_ALWAYS".		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_DTR_UPDATE_ALWAYS	Any DTR result reported by the monitor is used by the Dem.	
<b>Post-Build Variant Multiplicity</b>	DEM_DTR_UPDATE_STEADY	The Dem accepts reported DTRs only when the configured debouncing mechanism is stable at the FAIL or PASS limit.	
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	-	
	<b>Post-build time</b>	-	
<b>Scope / Dependency</b>	scope: local dependency: DemOBDSupport == DEM_OBD_MASTER_ECU    DEM_OBD_PRIMARY_ECU, DemDtrEventRef		

**No Included Containers**

### 10.2.44 DemFreezeFrameClass

#### DemFreezeFrameClass

<b>SWS Item</b>	[ECUC_Dem_00673]
<b>Container Name</b>	DemFreezeFrameClass
<b>Description</b>	This container contains the combinations of DIDs for a non OBD2 and WWH-OBD relevant freeze frame class.
<b>Configuration Parameters</b>	

<b>Name</b>	DemDidClassRef [ECUC_Dem_00707]		
<b>Description</b>	Reference to the DID elements which shall be contained in the freeze frame.		
<b>Multiplicity</b>	1..255		
<b>Type</b>	Reference to DemDidClass		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

**No Included Containers**

### 10.2.45 DemDidClass

#### DemDidClass

<b>SWS Item</b>	[ECUC_Dem_00706]
<b>Container Name</b>	DemDidClass
<b>Description</b>	This container contains the configuration (parameters) for a data Id class. It is assembled out of one or several data elements.

**Configuration Parameters**

<b>Name</b>	DemDidDataElementClassRef [ECUC_Dem_00617]		
<b>Description</b>	This reference contains the link to a data element class.  <b>Attributes:</b> requiresIndex=true		
<b>Multiplicity</b>	1..255		
<b>Type</b>	Reference to DemDataElementClass		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDidIdentifier [ECUC_Dem_00650]		
<b>Description</b>	Identifier of the Data ID.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b> <b>Default Value</b>	0 .. 65535		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>			

**No Included Containers**

### 10.2.46 DemJ1939FreezeFrameClass

#### DemJ1939FreezeFrameClass

<b>SWS Item</b>	[ECUC_Dem_00828]
<b>Container Name</b>	DemJ1939FreezeFrameClass

<b>Description</b>	This container contains the combinations of SPNs s for a J1939 relevant freeze frame.
<b>Configuration Parameters</b>	

<b>Name</b>	DemSPNClassRef [ECUC_Dem_00829]		
<b>Description</b>	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.  <b>Attributes:</b> requiresIndex=true		
<b>Multiplicity</b>	1..255		
<b>Type</b>	Reference to DemSPNClass		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

#### No Included Containers

### 10.2.47 DemSPNClass

#### DemSPNClass

<b>SWS Item</b>	[ECUC_Dem_00830]
<b>Container Name</b>	DemSPNClass
<b>Description</b>	This container contains the configuration (parameters) for a SPN.
<b>Configuration Parameters</b>	

<b>Name</b>	DemSPNDataElementClassRef [ECUC_Dem_00832]
<b>Description</b>	This reference contains the link to a data element class.
<b>Multiplicity</b>	1
<b>Type</b>	Reference to DemDataElementClass
<b>Post-Build Variant Value</b>	false

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemSPNId [ECUC_Dem_00831]				
<b>Description</b>	Suspect parameter number				
<b>Multiplicity</b>	1				
<b>Type</b>	EcucIntegerParamDef				
<b>Range</b>	0 .. 524287				
<b>Default Value</b>					
<b>Post-Build Variant Value</b>	false				
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Scope / Dependency</b>	scope: local				

#### No Included Containers

### 10.2.48 DemFreezeFrameRecordClass

#### DemFreezeFrameRecordClass

<b>SWS Item</b>	[ECUC_Dem_00801]		
<b>Container Name</b>	DemFreezeFrameRecordClass		
<b>Description</b>	This container contains a list of dedicated, different freeze frame record numbers.		
<b>Configuration Parameters</b>			

<b>Name</b>	DemFreezeFrameRecordNumber [ECUC_Dem_00777]				
<b>Description</b>	This parameter defines a record number for a freeze frame record. This record number is unique per freeze frame record number class.				
<b>Multiplicity</b>	1				
<b>Type</b>	EcucIntegerParamDef				
<b>Range</b>	0 .. 254				
<b>Default Value</b>					
<b>Post-Build Variant Value</b>	false				
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Scope / Dependency</b>	scope: ECU				

<b>Name</b>	DemFreezeFrameRecordTrigger [ECUC_Dem_00803]		
<b>Description</b>	Defines the trigger to store the FreezeFrameRecordTrigger.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_TRIGGER_ON_CO_NFIRMED	Event Memory entries are triggered if the UDS DTC status bit 3 (confirmedDTC) changes from 0 to 1.	
	DEM_TRIGGER_ON_FD_C_THRESHOLD	Event Memory entries are triggered when the FDC threshold is reached.	
	DEM_TRIGGER_ON_PENDING	Event Memory entries are triggered if the UDS DTC status bit 2 (pendingDTC) changes from 0 to 1.	
	DEM_TRIGGER_ON_TEST FAILED	Event Memory entries are triggered if the UDS DTC status bit 0 (testFailed) changes from 0 to 1.	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local dependency: DemTypeOfFreezeFrameRecordNumeration == DEM_FF_RECNUM_CONFIGURED		

<b>Name</b>	DemFreezeFrameRecordUpdate [ECUC_Dem_00802]		
<b>Description</b>	This parameter defines the case, when the freeze frame record is stored/updated.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_UPDATE_RECORD_NO	This record is only captured for new event memory entries.	
	DEM_UPDATE_RECORD_YES	This record is captured every time.	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

#### No Included Containers

### 10.2.49 DemFreezeFrameRecNumClass

#### DemFreezeFrameRecNumClass

<b>SWS Item</b>	[ECUC_Dem_00775]
<b>Container Name</b>	DemFreezeFrameRecNumClass
<b>Description</b>	<p>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.</p> <p>dependency: DemTypeOfFreezeFrameRecordNumeration = DEM_FF_RECNUM_CONFIGURED</p>
<b>Configuration Parameters</b>	

<b>Name</b>	DemFreezeFrameRecordClassRef [ECUC_Dem_00800]		
<b>Description</b>	This parameter references record number(s) for a freeze frame record.		
	<b>Attributes:</b> requiresIndex=true		
<b>Multiplicity</b>	1..254		
<b>Type</b>	Reference to DemFreezeFrameRecordClass		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X	All Variants
<b>Value Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X	All Variants
<b>Scope / Dependency</b>	scope: ECU dependency: DemTypeOfFreezeFrameRecordNumeration = DEM_FF_RECNUM_CONFIGURED		

#### No Included Containers

### 10.2.50 DemExtendedDataClass

#### DemExtendedDataClass

<b>SWS Item</b>	[ECUC_Dem_00664]
<b>Container Name</b>	DemExtendedDataClass
<b>Description</b>	This class contains the combinations of extended data records for an extended data class.
<b>Configuration Parameters</b>	

<b>Name</b>	DemExtendedDataRecordClassRef [ECUC_Dem_00774]		
<b>Description</b>	This reference contains the link to an extended data class record.		
<b>Attributes:</b> requiresIndex=true			
<b>Multiplicity</b>	1..253		
<b>Type</b>	Reference to DemExtendedDataRecordClass		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>			

#### No Included Containers

### 10.2.51 DemExtendedDataRecordClass

#### DemExtendedDataRecordClass

<b>SWS Item</b>	[ECUC_Dem_00665]
<b>Container Name</b>	DemExtendedDataRecordClass
<b>Description</b>	This container contains the configuration (parameters) for an extended data record class.  It is assembled out of one or several data elements.
<b>Configuration Parameters</b>	

<b>Name</b>	DemDataElementClassRef [ECUC_Dem_00771]
<b>Description</b>	This reference contains the link to a data element class.
<b>Attributes:</b> requiresIndex=true	
<b>Multiplicity</b>	1..255
<b>Type</b>	Reference to DemDataElementClass
<b>Post-Build Variant Multiplicity</b>	false
<b>Post-Build Variant Value</b>	false

<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>			

<b>Name</b>	DemExtendedDataRecordNumber [ECUC_Dem_00666]		
<b>Description</b>	<p>This configuration parameter specifies an unique identifier for an extended data record.</p> <p>One or more extended data records can be assigned to one diagnostic event/DTC.</p> <p>0x00 is reserved by ISO (therefore the minimal value equals 1)</p> <p>0xF0 to 0xFF are reserved by ISO (therefore the maximal value equals 239)</p>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	1 .. 239		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemExtendedDataRecordTrigger [ECUC_Dem_00804]		
<b>Description</b>	Defines the trigger to store the ExtendedDataRecord.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_TRIGGER_ON_CO_NFIRMED	ExtendedDataRecord will be stored when the event status confirmed bit changes from 0 to 1.	
	DEM_TRIGGER_ON_FD_C_THRESHOLD	ExtendedDataRecord will be stored when the FDC reaches its threshold.	
	DEM_TRIGGER_ON_MIR_ROR	ExtendedDataRecord will be stored when the event is transferred to the mirror memory.	
	DEM_TRIGGER_ON_PASSED	ExtendedDataRecord will be stored when the event is reported as passed (testFailed bit transition 1 → 0).	

<b>Post-Build Variant Value</b>	DEM_TRIGGER_ON_PENDING	ExtendedDataRecord will be stored when the event status pending bit changes from 0 to 1.	
	DEM_TRIGGER_ON_TEST_FAILED	ExtendedDataRecord will be stored when the event status test failed bit changes from 0 to 1.	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: local		

<b>Name</b>	DemExtendedDataRecordUpdate [ECUC_Dem_00621]		
<b>Description</b>	This extended data record is captured if the configured trigger condition in "DemExtendedDataRecordTrigger" is fulfilled.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	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.	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

#### No Included Containers

### 10.2.52 DemDataElementClass

#### DemDataElementClass

<b>SWS Item</b>	[ECUC_Dem_00610]
<b>Container Name</b>	DemDataElementClass
<b>Description</b>	This container contains the configuration (parameters) for an internal/external data element class.
<b>Configuration Parameters</b>	

Container Choices		
Container Name	Multiplicity	Scope / Dependency
DemExternalCSDDataElementClass	0..1	<p>This container contains the configuration (parameters) for an external client/server based data element class.</p> <p>It defines, how the Dem can obtain the value of the data element from either a SW-C or another BSW module. Whether a client/server port or a C function-call is used, is defined by DemDataElementUsePort.</p>
DemExternalSRDataElementClass	0..1	<p>This container contains the configuration (parameters) for an external sender/receiver based data element class. It defines, how the Dem can obtain the value of the data element from a SW-C, by using a sender/receiver port.</p>
DemInternalDataElementClass	0..1	<p>This container contains the configuration (parameters) for an internal data element class.</p>

### 10.2.53 DemDataElementInstance

#### DemDataElementInstance

SWS Item	[ECUC_Dem_00875]
Container Name	DemDataElementInstance
Description	Instance Reference to the primitive data in a port where the data element is typed with an ApplicationPrimitiveDataType or an ImplementationDataType.
Configuration Parameters	

Name	DemDataElementInstanceRef [ECUC_Dem_00861]		
Description	<p>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).</p> <p>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</p>		
Multiplicity	1		
Type	Instance reference to AUTOSAR-DATA-PROTOTYPE context: ROOT-SW-COMPOSITION-PROTOTYPE SW-COMPONENT-PROTOTYPE PORT-PROTOTYPE		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency			

**No Included Containers**

### 10.2.54 DemInternalDataElementClass

#### DemInternalDataElementClass

<b>SWS Item</b>	[ECUC_Dem_00684]
<b>Container Name</b>	DemInternalDataElementClass
<b>Description</b>	This container contains the configuration (parameters) for an internal data element class.
<b>Configuration Parameters</b>	

<b>Name</b>	DemDataElementDataSize [ECUC_Dem_00614]				
<b>Description</b>	Defines the size of the data element in bytes.				
<b>Multiplicity</b>	1				
<b>Type</b>	EcucIntegerParamDef				
<b>Range</b>	1 .. 255				
<b>Default Value</b>					
<b>Post-Build Variant Value</b>	false				
<b>Value Configuration Class</b>	Pre-compile time	X	All Variants		
	Link time	—			
	Post-build time	—			
<b>Scope / Dependency</b>	scope: ECU dependency: DemInternalDataElement				

<b>Name</b>	DemInternalDataElement [ECUC_Dem_00616]		
<b>Description</b>	This parameter defines the Dem-internal data value, which is mapped to the data element.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEM_AGINGCTR_DOWN_CNT	map down-counting Dem-internal aging counter, max. range: 1 byte	
	DEM_AGINGCTR_UPCN_T	map up-counting Dem-internal aging counter, max. range: 1 byte	
	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_FAILED_CYCLES	map Dem-internal Operation Cycle Counter - Failed cycles, max. range: 1 byte	

<b>Post-Build Variant Value</b>	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_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)	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

**No Included Containers**

### 10.2.55 DemExternalCSDDataElementClass

#### DemExternalCSDDataElementClass

<b>SWS Item</b>	[ECUC_Dem_00668]
<b>Container Name</b>	DemExternalCSDDataElementClass
<b>Description</b>	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.
<b>Configuration Parameters</b>	

<b>Name</b>	DemDataElementDataSize [ECUC_Dem_00646]	
<b>Description</b>	Defines the size of the data element in bytes.	
<b>Multiplicity</b>	1	
<b>Type</b>	EcucIntegerParamDef	
<b>Range</b>	1 .. 255	
<b>Default Value</b>		
<b>Post-Build Variant Value</b>	false	

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDataElementReadFnc [ECUC_Dem_00619]		
<b>Description</b>	In case of DemDataElementUsePort is false, this parameter defines the prototype of the C function "ReadDataElement" used to get the according value.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default Value</b>			
<b>Regular Expression</b>			
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDataElementUsePort [ECUC_Dem_00647]		
<b>Description</b>	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 DemDataElementReadFnc.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

#### No Included Containers

## 10.2.56 DemExternalSRDataElementClass

### DemExternalSRDataElementClass

<b>SWS Item</b>	[ECUC_Dem_00669]
<b>Container Name</b>	DemExternalSRDataElementClass
<b>Description</b>	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.
<b>Configuration Parameters</b>	

<b>Name</b>	DemDataElementDataSize [ECUC_Dem_00615]				
<b>Description</b>	Defines the size of the data element in bits.				
<b>Multiplicity</b>	1				
<b>Type</b>	EcucIntegerParamDef				
<b>Range</b>	1 .. 255				
<b>Default Value</b>					
<b>Post-Build Variant Value</b>	false				
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Scope / Dependency</b>	scope: ECU				

<b>Name</b>	DemDataElementDataType [ECUC_Dem_00840]				
<b>Description</b>	Provide the implementation data type of data belonging to a external data.				
<b>Multiplicity</b>	1				
<b>Type</b>	EcucEnumerationParamDef				
<b>Range</b>	BOOLEAN	type of the data is boolean.			
	SINT16	type of the data is sint16.			
	SINT32	type of the data is sint32.			
	SINT8	type of the data is sint8.			
	UINT16	type of the data is uint16.			
	UINT32	type of the data is uint32.			
	UINT8	type of the data is uint8.			
<b>Post-Build Variant Multiplicity</b>	false				
<b>Post-Build Variant Value</b>	false				
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU dependency: DemDataSize		

<b>Name</b>	DemDataElementEndianness [ECUC_Dem_00841]				
<b>Description</b>	Defines the endianness of the data belonging to an external data.  If no DemDataElementEndianness is defined the value of DemDataElementDefaultEndianness is applicable.				
<b>Multiplicity</b>	0..1				
<b>Type</b>	EcucEnumerationParamDef				
<b>Range</b>	BIG_ENDIAN	Most significant byte shall come at the lowest address.			
	LITTLE_ENDIAN	Most significant byte shall come highest address			
	OPAQUE false	opaque data endianness			
<b>Post-Build Variant Multiplicity</b>	false				
<b>Post-Build Variant Value</b>	false				
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants		
	<b>Link time</b>	—			
	<b>Post-build time</b>	—			
<b>Scope / Dependency</b>	scope: local dependency: DemDataElementDefaultEndianness				

<b>Included Containers</b>			
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>	
DemDiagnosisScaling	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation. Out of this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.	
DemSRDataElement Class	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 DemSubElementInDataElementInstance OR DemDataElementInstance OR DemSubElementInImplDataElementInstance reference.	

### 10.2.57 DemSRDataElementClass

#### DemSRDataElementClass

<b>SWS Item</b>	[ECUC_Dem_00842]
<b>Container Name</b>	DemSRDataElementClass
<b>Description</b>	<p>This container defines the source of data in a provided port which shall be read for a external data element</p> <p>This container shall contain either one DemSubElementInDataElementInstance OR DemDataElementInstance OR DemSubElementInImplDataElementInstance reference.</p>
<b>Configuration Parameters</b>	

Container Choices		
Container Name	Multiplicity	Scope / Dependency
DemDataElement Instance	0..1	Instance Reference to the primitive data in a port where the data element is typed with an ApplicationPrimitiveDataType or an ImplementationDataType.
DemSubElementInDataElementInstance	0..1	Instance Reference to the primitive sub-element (at any level) of composite data in a port where the data element is typed with an ApplicationCompositeDataType.
DemSubElementInImplDataElementInstance	0..1	Instance Reference to the primitive sub-element (at any level) of composite data in a port where the data element is typed with an ImplementationDataType of category STRUCTURE or ARRAY.

### 10.2.58 DemSubElementInDataElementInstance

#### DemSubElementInDataElementInstance

<b>SWS Item</b>	[ECUC_Dem_00874]
<b>Container Name</b>	DemSubElementInDataElementInstance
<b>Description</b>	Instance Reference to the primitive sub-element (at any level) of composite data in a port where the data element is typed with an ApplicationCompositeDataType.
<b>Configuration Parameters</b>	

<b>Name</b>	DemSubElementInDataElementInstanceRef [ECUC_Dem_00860]		
<b>Description</b>	Instance Reference to the primitive sub-element (at any level) of composite data in a port 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 ApplicationCompositeDataType.		
<b>Multiplicity</b>	1		
<b>Type</b>	Instance reference to AUTOSAR-DATA-PROTOTYPE context: ROOT-SW-COMPOSITION-PROTOTYPE SW-COMPONENT-PROTOTYPE PORT-PROTOTYPE AUTOSAR-DATA-PROTOTYPE APPLICATION-COMPOSITE-ELEMENT-DATA-PROTOTYPE*		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X	All Variants
<b>Scope / Dependency</b>			

No Included Containers
------------------------

## 10.2.59 DemSubElementInImplIDataElementInstance

### DemSubElementInImplIDataElementInstance

<b>SWS Item</b>	[ECUC_Dem_00876]
<b>Container Name</b>	DemSubElementInImplIDataElementInstance
<b>Description</b>	Instance Reference to the primitive sub-element (at any level) of composite data in a port where the data element is typed with an ImplementationDataType of category STRUCTURE or ARRAY.
<b>Configuration Parameters</b>	

<b>Name</b>	DemSubElementInImplDataElementInstanceRef [ECUC_Dem_00862]		
<b>Description</b>	<p>Instance Reference to the primitive sub-element (at any level) of composite data in a port which shall be read or written.</p> <p>Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only).</p> <p>This reference is applicable if the AutosarDataPrototype is typed with a ImplementationDataType of category STRUCTURE or ARRAY.</p> <p>Please note that in case of ARRAY the index attribute in the target reference has to be set to select a single array element.</p>		
<b>Multiplicity</b>	1		
<b>Type</b>	Instance reference to IMPLEMENTATION-DATA-TYPE-ELEMENT context: ROOT-SW-COMPOSITION-PROTOTYPE SW-COMPONENT-PROTOTYPE PORT-PROTOTYPE AUTOSAR-DATA-PROTOTYPE I MPLEMENTATION-DATA-TYPE-ELEMENT*		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
<b>Scope / Dependency</b>			

#### No Included Containers

### 10.2.60 DemDiagnosisScaling

#### DemDiagnosisScaling

<b>SWS Item</b>	[ECUC_Dem_00843]		
<b>Container Name</b>	DemDiagnosisScaling		
<b>Description</b>	This container contains the configuration (parameters) of an alternative Diagnosis Representation. Out of this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.		
<b>Configuration Parameters</b>			

Container Choices		
Container Name	Multiplicity	Scope / Dependency
DemAlternativeDataInterface	0..1	<p>This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of a VariableDataPrototype in a DataInterface.</p> <p>Additionally a reference to PortInterfaceMapping can be defined which provide already the mapping rules between the VariableDataPrototype in a DataInterface used by the software component (DemSRDataElementClass) and the intended Diagnosis Representation defined by DemExternalSRDataElementClass.</p>

DemAlternativeDataProps	0..1	<p>This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of ECU configuration parameters.</p> <p>The physical unit of the alternative data representation is defined by the DataPrototype referenced by DemSRDataElementClass.</p> <p>Additionally the definition of a text table mapping can be defined for DemDataTypeCategory TEXTTABLE and SCALE_LINEAR_AND_TEXTTABLE</p>
DemAlternativeDataType	0..1	<p>This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of an ApplicationDataType.</p> <p>Additionally the definition of a text table mapping can be defined for ApplicationDataTypes of category TEXTTABLE and SCALE_LINEAR_AND_TEXTTABLE.</p>

### 10.2.61 DemAlternativeDataInterface

#### DemAlternativeDataInterface

SWS Item	[ECUC_Dem_00844]
Container Name	DemAlternativeDataInterface
Description	<p>This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of a VariableDataPrototype in a DataInterface.</p> <p>Additionally a reference to PortInterfaceMapping can be defined which provide already the mapping rules between the VariableDataPrototype in a DataInterface used by the software component (DemSRDataElementClass) and the intended Diagnosis Representation defined by DemExternalSRDataElementClass.</p>
Configuration Parameters	

Name	DemDataElement [ECUC_Dem_00845]		
Description	Alternative Diagnosis Representation for the data defined by the means of a VariableDataPrototype in a DataInterface.		
Multiplicity	1		
Type	Foreign reference to VARIABLE-DATA-PROTOTYPE		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
Scope / Dependency			

<b>Name</b>	DemPortInterfaceMapping [ECUC_Dem_00846]		
<b>Description</b>	Optional reference to PortInterfaceMapping which defines the mapping rules.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Foreign reference to PORT-INTERFACE-MAPPING		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>			

#### No Included Containers

### 10.2.62 DemAlternativeDataType

#### DemAlternativeDataType

<b>SWS Item</b>	[ECUC_Dem_00847]		
<b>Container Name</b>	DemAlternativeDataType		
<b>Description</b>	<p>This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of an ApplicationDataType.</p> <p>Additionally the definition of a text table mapping can be defined for ApplicationDataTypes of category TEXTTABLE and SCALE_LINEAR_AND_TEXTTABLE.</p>		
<b>Configuration Parameters</b>			

<b>Name</b>	DemApplicationDataType [ECUC_Dem_00848]		
<b>Description</b>	Alternative Diagnosis Representation for the data defined by the means of a ApplicationPrimitiveDataType of category VALUE or BOOLEAN.		
<b>Multiplicity</b>	1		
<b>Type</b>	Foreign reference to APPLICATION-PRIMITIVE-DATA-TYPE		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>			

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DemTextTableMapping	0..*	<p>This container contains the configuration (parameters) of the mapping a DataPrototype typed by AutosarDataType that refer to a CompuMethods of category TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.</p> <p>Each DemTextTableMapping defines a value pair which is used to map the ECU internal value (DemInternalDataValue) to the value used in the diagnosis representation (DemDiagnosisRepresentationDataValue) and vice versa.</p> <p>The set of all DemTextTableMapping defines the whole mapping of an data.</p>

### 10.2.63 DemAlternativeDataProps

#### DemAlternativeDataProps

SWS Item	[ECUC_Dem_00852]
Container Name	DemAlternativeDataProps
Description	<p>This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of ECU configuration parameters.</p> <p>The physical unit of the alternative data representation is defined by the DataPrototype referenced by DemSRDataElementClass.</p> <p>Additionally the definition of a text table mapping can be defined for DemDataTypeCategory TEXTTABLE and SCALE_LINEAR_AND_TEXTTABLE</p>
Configuration Parameters	

Name	DemDataTypeCategory [ECUC_Dem_00857]		
Description	Data category of the alternative Diagnosis Representation.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	SCALE_LINEAR_AND_TEXTTABLE		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DemLinearScale	0..1	This container contains the configuration (parameters) of an linear scale of the alternative Diagnosis Representation.
DemTextTableMapping	0..*	<p>This container contains the configuration (parameters) of the mapping a DataPrototype typed by AutosarDataType that refer to a CompuMethods of category TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.</p> <p>Each DemTextTableMapping defines a value pair which is used to map the ECU internal value (DemInternalDataValue) to the vale used in the diagnosis representation (DemDiagnosisRepresentationDataValue) and vice versa.</p> <p>The set of all DemTextTableMapping defines the whole mapping of an data.</p>

## 10.2.64 DemLinearScale

### DemLinearScale

SWS Item	[ECUC_Dem_00859]
Container Name	DemLinearScale
Description	This container contains the configuration (parameters) of an linear scale of the alternative Diagnosis Representation.
Configuration Parameters	

Name	DemDiagnosisRepresentationDataLowerRange [ECUC_Dem_00856]				
Description	Lower Range for this scale of the data in the alternative Diagnosis Representation.				
Multiplicity	1				
Type	EcucFloatParamDef				
Range	-INF .. INF				
Default Value					
Post-Build Variant Value	false				
Value Configuration Class	Pre-compile time	X	All Variants		
	Link time	—			
	Post-build time	—			
Scope / Dependency	scope: ECU				

Name	DemDiagnosisRepresentationDataOffset [ECUC_Dem_00854]		
Description	Data offset of the alternative Diagnosis Representation for this scale.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	0 .. INF		

<b>Default Value</b>	0		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDiagnosisRepresentationDataResolution [ECUC_Dem_00853]		
<b>Description</b>	Data resolution of the alternative Diagnosis Representation for this scale.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range Default Value</b>	0 .. INF		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemDiagnosisRepresentationDataUpperRange [ECUC_Dem_00855]		
<b>Description</b>	Upper Range for this scale of the data in the alternative Diagnosis Representation.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range Default Value</b>	-INF .. INF		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	—	
	<b>Post-build time</b>	—	
<b>Scope / Dependency</b>	scope: ECU		

#### No Included Containers

### 10.2.65 DemTextTableMapping

#### DemTextTableMapping

<b>SWS Item</b>	[ECUC_Dem_00849]
-----------------	------------------

<b>Container Name</b>	DemTextTableMapping
<b>Description</b>	<p>This container contains the configuration (parameters) of the mapping a DataPrototype typed by AutosarDataType that refer to a CompuMethods of category TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.</p> <p>Each DemTextTableMapping defines a value pair which is used to map the ECU internal value (DemInternalDataValue) to the value used in the diagnosis representation (DemDiagnosisRepresentationDataValue) and vice versa.</p> <p>The set of all DemTextTableMapping defines the whole mapping of an data.</p>
<b>Configuration Parameters</b>	

<b>Name</b>	DemDiagnosisRepresentationDataValue [ECUC_Dem_00851]		
<b>Description</b>	The data value in the diagnosis representation.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 18446744073709551615		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X	All Variants
<b>Scope / Dependency</b>	scope: ECU		

<b>Name</b>	DemInternalDataValue [ECUC_Dem_00850]		
<b>Description</b>	The ECU internal data value.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 18446744073709551615		
<b>Default Value</b>			
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b> <b>Link time</b> <b>Post-build time</b>	X	All Variants
<b>Scope / Dependency</b>	scope: ECU		

#### No Included Containers

## 10.3 Published information

For details refer to the chapter [10.3 “Published Information”](#) in SWS\_BSWGeneral.

## 11 Not Applicable Requirements

**[SWS\_Dem\_00999]** [ These requirements are not applicable to this specification.  
]([SRS\\_BSW\\_00005](#), [SRS\\_BSW\\_00161](#), [SRS\\_BSW\\_00162](#), [SRS\\_BSW\\_00164](#),  
[SRS\\_BSW\\_00168](#), [SRS\\_BSW\\_00170](#), [SRS\\_BSW\\_00171](#), [SRS\\_BSW\\_00324](#),  
[SRS\\_BSW\\_00326](#), [SRS\\_BSW\\_00327](#), [SRS\\_BSW\\_00331](#), [SRS\\_BSW\\_00337](#),  
[SRS\\_BSW\\_00338](#), [SRS\\_BSW\\_00339](#), [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\\_00361](#), [SRS\\_BSW\\_00369](#),  
[SRS\\_BSW\\_00374](#), [SRS\\_BSW\\_00375](#), [SRS\\_BSW\\_00379](#), [SRS\\_BSW\\_00382](#),  
[SRS\\_BSW\\_00387](#), [SRS\\_BSW\\_00433](#), [SRS\\_BSW\\_00434](#), [SRS\\_Diag\\_04005](#),  
[SRS\\_Diag\\_04006](#), [SRS\\_Diag\\_04007](#), [SRS\\_Diag\\_04015](#), [SRS\\_Diag\\_04016](#),  
[SRS\\_Diag\\_04019](#), [SRS\\_Diag\\_04020](#), [SRS\\_Diag\\_04021](#), [SRS\\_Diag\\_04032](#),  
[SRS\\_Diag\\_04033](#), [SRS\\_Diag\\_04036](#), [SRS\\_Diag\\_04059](#), [SRS\\_Diag\\_04064](#),  
[SRS\\_Diag\\_04072](#), [SRS\\_Diag\\_04075](#), [SRS\\_Diag\\_04086](#), [SRS\\_Diag\\_04087](#),  
[SRS\\_Diag\\_04089](#), [SRS\\_Diag\\_04090](#), [SRS\\_Diag\\_04091](#), [SRS\\_Diag\\_04097](#),  
[SRS\\_Diag\\_04098](#), [SRS\\_Diag\\_04100](#), [SRS\\_Diag\\_04101](#), [SRS\\_Diag\\_04119](#),  
[SRS\\_Diag\\_04120](#), [SRS\\_Diag\\_04121](#), [SRS\\_Diag\\_04135](#), [SRS\\_Diag\\_04136](#),  
[SRS\\_Diag\\_04138](#), [SRS\\_Diag\\_04139](#), [SRS\\_Diag\\_04142](#), [SRS\\_Diag\\_04143](#),  
[SRS\\_Diag\\_04144](#), [SRS\\_Diag\\_04145](#), [SRS\\_Diag\\_04146](#))