VECTOR

MICROSAR Classic Diagnostic Event Manager (DEM)

Technical Reference

Addendum for OBD II, OBD on UDS, WWH-OBD and ZEV on UDS
Version 27.00.00

Document Information

History

Author	Date	Version	Remarks
visdth	2012-08-31	0.01.00	preliminary version
visdth	2012-10-02	1.00.00	beta release
visade	2014-11-03	2.00.00	> Added WWH-OBD (beta) > Added *Figure 2-5* and update of *Figure 2-23* > Removed deprecated API *Dem_GetNextIUMPRRatioData()* Added clarification for 1-trip DTCs in chapter *2.2.4.2*
visanh	2017-05-08	3.00.00	> Fixed ESCAN00091941 (chapter *2.2.4.3*, chapter 2.2.5.1.4)
visanh	2017-10-05	4.00.00	> Added chapter *2.2.12.4 Unavailable DTRs* and adapted chapters *2.3.8 DTR Integration* and *2.2.12 Diagnostic Test Result* > Added clarification in chapters *2.2.3.3*, *2.2.3.7* and *2.2.4.1*
visfrs	2018-04-18	5.00.00	> Adapt update and reset behavior of WWH-OBD freeze frames (chapter 2.2.5.3.1 and 2.2.5.3.2)
visavi	2019-03-19	17.00.00	> Added clarification regarding denominator group DEM_IUMPR_DEN_500MI in chapter 2.3.7.1.1
visavi	2019-05-10	17.01.00	> Corrected figure regarding warm up cycle behavior in chapter 2.2.2.4 (ESCAN00102797)
visfrs	2019-05-24	17.01.00	> Added information about dependency of storage of legislative freeze frame on event storage
visejz	2019-06-27	17.02.00	> Fix ESCAN00102806 (chapter 2.2.11.1.4, 2.3.7.2, 2.4.1)
visern	2019-07-01		> Adapted OBDII Freeze Frame recovery in chapter 2.2.14.2
visern	2019-07-12	17.03.00	> Added Clarification for Non-Volatile write frequency for OBD event qualification status in chapter 2.2.14.1
visavi	2019-08-29	17.04.00	> Corrected Page Alignment
vissat	2019-09-04	18.00.00	> Fixed ESCAN00103808 (chapters *2.2.1.3.3)*
visera	2019-09-18		> Fixed ESCAN00103966 (chapters 2.2.2.4 and 2.2.9.5)
visejz	2019-10-04	18.01.00	> Added information about support of Permanent Memory for WWH-OBD: > Added chapter 2.2.6.2 WWH-OBD

			> Added chapter 2.2.10.2.7 Service 19 55: Report Emission Related DTCs with Permanent Status > Adapted 2.2.1.1 Deviations > Adapted 2.2.10.2.5 Service 19 15: Report DTCs with Permanent Status > Adapted 2.2.14 Non-Volatile Data Management > Adapted 2.3.3.5 Permanent DTCs > Adapted 2.3.6 NVM Integration > Adapted 2.4.1 DemGeneral
visejz	2019-11-14	18.02.00	> Added information regarding event combination type 2 > Adapted 2.2.4.3 Event Displacement > Added 2.2.1.2.1.5 Event Combination Type 2
visern	2019-11-18		> Fixed ESCAN00104233 > Adapted section 2.2.12.2 Conversion of DTR values
visern	2019-11-28	18.03.00	> Added information for: > IUMPR General Individual Denominator Conditions > IUMPR General OBD Denominator Conditions. > Adapted 2.3.7.1.3 General Denominator > Adapted 4.1.2.9 Dem_SetIUMPRDenCondition() > Adapted 2.2.11.1.1: Additional Denominator Conditions can be reported before the General Denominator Condition.
visavi	2020-02-03	18.04.00	> Fixed ESCAN00105311 > Updated information regarding PID $41 calculation for disabled events in chapter 2.2.9.7 > Adapted 4.1.2.1 Dem_SetEventDisabled()
visfrs	2020-03-09	19.00.00	> Fixed ESCAN00105373 > Update description of OBD freeze frame behavior in chapter 2.2.5.1.2 and 2.2.5.1.5.
vissko	2020-04-20	19.01.00	> Added support for independent denominator group > Added additional information to chapter 2.1 introduction
vissat	2020-04-24	19.02.00	> Adapted legal information in chapter 2.1 Introduction
visfrs	2020-04-29	19.02.00	> Adapted limitations for OBD and J1939 (chapter 2.2.1.2.1.2)
visera	2020-05-12	19.03.00	> Adapted recovery of WIR bit for events that store a PDTC and support the MIL but do not currently request it in chapter 2.2.14.2
visern	2020-07-15	19.04.00	> Fixed ESCAN00106517

			> Added Dem_IUMPRLockNumerators() to Exclusive Area 0 in Chapter 2.3.5 Critical Sections
visanh	2020-07-17		> Added chapter 2.4.4 Post-Build Selectable
visavi	2020-08-06	19.05.00	> Added restriction regarding usage of NvM Initialization callbacks in chapter 2.3.6.2.
visera	2020-08-24		> Clarify confirmation of events attached to a MIL Group in chapter 2.2.3.3 MIL Groups
visejz	2020-09-28	20.00.00	> Added clarification regarding usage of NvM Initialization callbacks in chapter 2.3.6.2.
visern	2020-10-01	20.01.00	> Added a new API 4.1.1.18 Dem_SetDTCFilterByReadinessGroup() for Service 0x19 Subfunction 0x56
vissko	2020-11-12	20.02.00	> Updated chapters 2.2.4.2, 2.2.5, 2.2.6.1.2, 2.2.10.1.3, and 2.4.2.2 with respect to delayed external visibility of OBD events > Clarified visibility of permanent DTCs in chapter 2.2.6.1.5
visxli	2021-01-13	20.03.00	> Adapted chapter 2.2.5.1.2 Reporting in Mode $02
vireno	2021-01-14	20.03.00	> Explained how to fulfill 'tested passed since DTC cleared' for DTCs for event combination type 1 in chapter 2.2.9.1.
visern	2021-01-20	20.04.00	> Fixed ESCAN00100513 > Added new parameter DTCFormat for Dem_DcmGetDTCOfOBDFreezeFrame().
visern	2021-01-21	20.04.00	> Fixed ESCAN00108244 > Added fallback handling of OBD relevant events. In chapter 2.2.4.3 Event Displacement.
visern	2021-01-28	20.04.00	> Fixed ESCAN00108379 > Fixed BufSize parameter type of Dem_DcmReadDataOfOBDFreezeFrame().
visxli	2021-02-03	20.04.00	> Fixed ESCAN00108370 > Clarified mapping recommendation of monitors in chapter 2.3.7.1.2.
visern	2021-02-15	20.05.00	> Updated Template
visern	2021-04-06	21.00.00	> Fixed ESCAN00108875 > Added Dem_SetEventDisabled() to Exclusive Area 1 in Chapter 2.3.5 Critical Sections
visera	2021-05-03	21.01.00	> Fixed ESCAN00108707 > Added chapter 2.2.1.3.6 Operation SetEventDisabled of Service Interface Diagnostic Monitor
visejz	2021-05-18	21.02.00	> Adaptation to new template

			> Moved content from chapter 7 to chapter 2.2.1 Features > Support of AUTOSAR Release 19-11 > Complete update of chapter 2.2.1 Features
visrk	2021-06-02	21.03.00	> Added description of OBD on UDS > Added chapter 2.2.5.2 on Legislative Freeze Frame for legislation OBDonUDS. > Added chapter 2.2.15 OBD Extended Data Records on extended data records for OBD on UDS legislation.
visejz	2021-07-06	21.04.00	> Updated version scheme of referenced AUTOSAR documents.
visern	2021-07-09	21.04.00	> Added new API 4.1.1.10 Dem_DcmReadDataOfPIDF501() > Added newly added Readiness Groups to 4.1.1.18 Dem_SetDTCFilterByReadinessGroup() > Adapted Chapter 2.2.9.1 PID $01 and PID $F501: Monitor status since DTCs cleared > Added information regarding PID $F501
visera	2021-07-21	21.04.00	> Adapt API names in Table 4-46 Port OBDServices
visejz	2021-07-28	21.04.00	> Added a reference where to find DTR conversion formula in chapter 2.2.12.2 Conversion of DTR values > Added DTCClass support for OBD on UDS > Adapted Table 2-2 Comparison of OBD II, OBDonUDS and WWH-OBD > Adapted chapter 2.2.3.5 DTC Class > Extended Table 2-45 DTC configuration
visrk	2021-08-03	21.05.00	> Enabled mode $06 APIs for OBD on UDS, see chapters (2.2.12, 2.2.10.1.5 and 2.2.10.2.8).
visxli	2021-08-05	21.05.00	> Added new parameter DTCFormat in API Dem_SetDTCFilterByReadinessGroup()
vireno	2021-08-06	21.05.00	> Added description of Monitor Activity Ratio (MAR) in chapters 2.2.13 and 2.2.15.3. > Added new API 4.1.1.19 Dem_DcmGetInfoTypeValue79()
visern	2021-08-06	21.05.00	> Added additional information to 2.2.15.2 Extended Data Record 0x92, which describes the limitation of the UASID to 1 byte.
visavi	2021-08-10	21.05.00	> Added information regarding freezing of PID $21, $31 in chapters 2.2.9.4 and 2.2.9.6.
visera	2021-08-19	21.05.00	> Added hint to chapter 2.2.5.1.2 to stress the importance of the configuration of event priorities as described in the Technical Reference

visern	2021-11-12	22.00.00	> Corrected association between OBDII and OBDonUDS to Services $19 15 and $19 55 in the following chapters: > 2.2.6.1.1 Definition > 2.2.6.1.2 Storage > 2.2.10.2.5 Service 19 15: Report DTCs with Permanent Status
visera	2021-11-15	22.00.00	> Clarify meaning of caution box in chapter 2.3.7.1.2.
visfrs	2022-01-07	22.01.00	> Add abbreviations to chapter 5.2 > Describe port availability in chapter 4.2
visern	2022-02-02	22.02.00	> Replaced secondary memory with user defined memory in chapter 4.1.2.10 Dem_SetIUMPRFilter()
eacar	2022-02-08	22.02.00	> Corrected reentrancy description of 4.1.2.9 Dem_SetIUMPRDenCondition()
visern	2022-02-16	22.03.00	> Fixed ESCAN00110167 > Added in Chapter 2.2.15.1 Extended Data Record 0x91 information what is done, if multiple ratios have the same value.
visxli	2022-02-24	22.03.00	> ESCAN00110725 > Added information regarding the configurable calculation of comprehensive component readiness group in PID $F501 in Chapter 2.2.9.1 PID $01 and PID $F501: Monitor status since DTCs cleared
vissko	2022-02-28	22.03.00	> ESCAN00111358 > Added information on (re-)initialization of Permanent DTC NvRAM block in Chapter 2.3.6.2
visxli	2022-02-28	22.03.00	> ESCAN00110661 > Adapted and added information regarding to the time point of MAR update in Chapter 2.2.13 Monitor Activity Ratio
vislan	2022-03-02	22.03.00	> Product name updated to MICROSAR Classic.
visfrs	2022-03-21	23.00.00	> Clarify degration conditions of activation modes in chapter 2.2.7
visabn	2022-03-29	23.00.00	> Updated Chapter 2.2.1.2.1.2 to support DM11 > Added Chapter 2.2.16 to document supported DMs for OBD II on J1939
visavi	2022-03-29	23.00.00	> Updated information regarding DTC Configuration in Chapter 2.4.2.1
visern	2022-04-22	23.01.00	> ESCAN00111651 > Updated chapter 2.2.7.1 Single B1 Counter description. > B1 Counter now needs 3 consecutive DCY without failed report to reset

			> Added info box for single DTC clear
visera	2022-05-05	23.01.00	> Document support for J1939 Diagnostic Messages DM6, DM12, DM23, DM27, DM28 and DM29 for OBD II > Adapted chapter 2.2.1.2.1.2 Incompatible Features > Adapted chapter 2.2.16 J1939
visabn	2022-05-16	23.02.00	> Added new section in 2.2.1.3 Uncommon Feature Behavior regarding PID values.
visera	2022-05-23	23.02.00	> Document support for Diagnostic Message DM20 > Adapted and extended chapter 2.2.16 J1939 > Added chapter 4.1.3 J1939Dcm > Adapted chapter 2.2.1.2 Limitations > Extended chapter 2.3.4 Development Error Codes
visabn	2022-06-28	23.03.00	> Document support for Diagnostic Message DM05 > Adapted and extended chapter 2.2.16 J1939 > Adapted chapter 2.2.1.2 Limitations > Updated Dem_SetDTCFilterByReadinessGroup() with newly supported readiness groups
viswsi	2022-07-19	23.04.00	> Document support for Diagnostic Message DM21 > Adapted and extended chapter 2.2.16 J1939 > Adapted chapter 4.1.3 J1939Dcm > Adapted chapter 2.2.1.2 Limitations > Extended chapter 2.3.4 Development Error Codes
visabn	2022-07-20	23.04.00	> Document support for Power Take-Off > Added new API 4.1.2.21 Dem_SetPtoStatus() > Removed limitation for PTO in 2.2.1.1 Deviations
visxli	2022-08-04	23.05.00	> Document support for Power Take-Off in IUMPR processing > Added chapter 2.2.3.8 Power Take-Off > Added chapter 2.2.11.4 Effect of Power Take-Off
visxli	2022-08-19	23.05.00	> Document support for Power Take-Off Client Server Interface > Added chapter 4.2.1.1.7 PowerTakeOff
eacar	2022-09-13	24.00.00	> Extended descriptions of aging behavior for black MIL DTCs and OBD DTCs which could not be stored in the memory > Adapted chapter 2.2.3.7 Black MIL DTCs > Adapted chapter 2.2.4.1 General Behavior
visabn	2022-09-26	24.01.00	> Document Support for Diagnostic Message DM24 > Adapted chapter 2.2.16 J1939

			> Added Chapter 2.2.16.1.6 Diagnostic Message 24: SPN Support
visabn	2022-10-13	24.01.00	> Document Support for Diagnostic Message DM25 > Adapted chapter 2.2.16 J1939 > Added Chapter 2.2.16.1.7 Diagnostic Message 25: Expanded Freeze Frame
visera	2022-11-21	24.02.00	> Document Support for Diagnostic Message DM26: > Extended chapter 2.2.1.2 Limitations > Adapted chapter 2.3.2 Required Data > Adapted chapter 2.2.16 J1939 > Added chapter 2.2.16.1.8 Diagnostic Message 26: Diagnostic Readiness 3 > Added chapter 4.1.3.4 Dem_J1939DcmReadDiagnosticReadiness3()
visabn	2022-12-21	24.03.00	> Document Support for PTO Activation Timer > Extended chapter 2.2.9.1 > Extended chapter 2.2.3.8 > Extended chapter 2.2.1.2
visanh	2023-01-05	24.03.00	> Extended chapter 2.2.4.3 Event Displacement > Extended chapter 2.2.3.3 MIL Groups > Extended chapter 2.2.7 Activation Mode > Improved description of chapter 4.1.1.17 Dem_DcmGetDTRData() > Adapted chapters 2.2.1.2.1 and 2.4.4 and added chapter 2.2.16.2 related to usage of J1939 in combination with OBD
abjoerkqvist	2023-02-03	24.04.00	> Document OBD UDS DTC Separation: > Adapted chapter 2.2.3 > Added chapter 2.2.3.1 > Extended chapter 2.4.1.9 > Extended chapter 2.4.2.1
visavi	2023-02-15	24.04.00	> Fixed ESCAN00113571 > Adapted chapter 2.2.14.1
visanh	2023-06-16	25.00.00	> Improved description of B1 counter in chapter 2.2.7.1
vsarcmiem	2023-07-14	25.02.00	> Document Event Monitor Disabled > Adapted chapter 2.2.9.1 > Added chapter 4.1.2.22 > Extended chapter 2.4.1.9 > Extended chapter 4.2.1.1.1
visera	2023-08-09	25.03.00	> Fixed ESCAN00115250

			> Adapted chapter 2.4.1.7 Data Element for Distance Information and 2.4.1.8 Data Elements for Time Since Engine Start Minutes or Seconds
visxli	2023-08-17	25.03.00	> Added chapter 2.2.3.4 Event Combination
visera	2023-08-18	25.03.00	> Fixed ESCAN00115010 > Adapted chapter 2.2.7.1 Single B1 Counter
sbappanadu	2023-11-15	26.02.00	> Improved description of Warm-Up cycle in chapter 2.2.1.3.4, chapter 2.2.2.4, chapter 2.3.1.2 > Updated figure in chapter 2.2.2.4 > Added new sub-section 2.2.2.4.1 Delayed Warm-up cycle End
visxli	2023-12-04	26.03.00	> Adapted chapter 2.2.3.4 Event Combination > Adapted behavior for event combination type 1 and type 3 > Added common healing counter
emalekzadeh	2023-12-13	26.03.00	> Updated chapter 2.2.3.4 Event Combination with DTC Pending bit behavior
tfarahani	2024-02-08	26.04.00	> Updated chapter 2.2.14.2 Data Recovery with CDTC bit recovery behavior
visfrs	2024-03-01	26.04.00	> Fixed ESCAN00116678 > Adapted description of readiness since last clear > Added support of Extended Data record 0x93 to Table 2-2
visanh	2024-03-05	26.04.00	> Added several clarifications regarding IUMPR > Adapted chapter 2.3.7.1.3 and 4.1.2.9 regarding inhibited denominator condition status > Adapted chapters 2.2.11.1.4, 2.2.11.2, 2.3.3.3 and 2.3.7.2 regarding locking of IUMPR counters due to a disabled FIM FID > Adapted chapters 2.3.7.1.1 and 4.1.2.9 regarding locking of specific denominator groups by inhibiting the general individual denominator condition status
visanh	2024-03-05	26.04.00	> Added default visibility information for service $19 08 in chapter 2.2.3.6 Dependent Secondary ECU DTCs
vsgeei	2024-03-05	26.04.00	> Moved all Emission OBD related functionality into new chapter 2. > Moved Legal Information to chapter 1.1 > Moved interfaces section into chapter 4. > Added Zero Emission Vehicle OBD chapter 3 > Added chapter 3.1 Introduction > Added chapter 3.2 Functional Description

			> Added chapter 3.3 Integration > Added chapter 3.4 Configuration > Added chapter 3.5 Incompatible Features
tfarahani	2024-03-13	27.00.00	> Added DM22 to the list of supported diagnostic messages in chapter 2.2.16 J1939 > Added chapter 2.2.16.1.5 Diagnostic Message 22: Individual Clear/Reset of Active and Previously Active DTC
tfarahani	2024-03-27	27.00.00	> Added default visibility information for service $19 16, $19 1A, $19 42 and $19 56 in chapter 2.2.3.6 Dependent Secondary ECU DTCs

Reference Documents

No.	Source	Title	Version
[1]	Vector	MICROSAR Classic Diagnostic Event Manager Technical Reference	see delivery
[2]	AUTOSAR	Specification of Diagnostic Event Manager	R19-11
[3]	AUTOSAR	Specification of Diagnostic Communication Manager	R4.2.1
[4]	AUTOSAR	Specification of a Diagnostic Communication Manager for SAE J1939	R19-11
[5]	AUTOSAR	Specification of Function Inhibition Manager	R2.2.0
[6]	SAE	J1979 – E/E Diagnostic Test Modes	SEP2010
[7]	ISO	15031-5 Road Vehicles – Communication between vehicle and external equipment for emissions-related diagnostics – Part 5: Emissions-related diagnostic services	-
[8]	CARB	California Code Regulations, Section 1968.2 (OBD II)	2012-08-07
[9]	ISO	14229-1 Road vehicles – Unified diagnostic services (UDS) – Part 1: Specification and requirements	-
[10]	ISO	27145-3 Road Vehicles – Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements – Part 3: Common message dictionary	-
[11]	SAE	J1979-2 – E/E Diagnostic Test Modes: OBDonUDS	APR2021
[12]	SAE	J1979DA	APR2021
[13]	SAE	J1939-73 – Application Layer Diagnostics	JAN2019
[14]	SAE	J1979-3 – E/E Diagnostic Test Modes: ZEVonUDS	DEC2022
[15]	CARB	California Code Regulations, Section 1962.5 (ZEVonUDS)	AUG2022
[16]	SAE	J1979DA	Draft from NOV2023

❗ **Caution**
We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector´s release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.

❗ **Caution**
This symbol calls your attention to warnings.

Contents

Illustrations

Tables

1 Scope

Introduction
This document is an addendum to the Dem Technical Reference ([1]). It describes OBD specific extensions and deviation from the standard behavior.

Not included herein are topics that are already handled by the regular Technical Reference.

The term “On-Board Diagnostics” in the context of the Dem refers to the detection, evaluation, storage and reporting of emission related faults according to different legislative regulations.

Not all OBD requirements need to be fulfilled by any emission related component. Depending on the functionality, AUTOSAR defines the following types:

Master ECU: only one master ECU exists in the vehicle network. It does not only monitor and store emission related data, but also calculates environmental and status data that is distributed to other emission related ECUs. The Master ECU is also responsible for illuminating the MIL.

Primary ECUs: systems that monitor and store emission related data in their fault memory.

Secondary ECUs: supplementary components with emission related functions that do not store emission related data by itself and do not communicate with a generic scan tool directly. If required, data storage is performed by a related Master or Primary ECU.

This document shall give an overview of these requirements, how they are implemented in the Dem and what need to be done by the application to achieve OBD compliance.

1.1 Legal Information

Caution

The DEM is highly configurable and provides a variety of interfaces. It is therefore possible that certain configurations and usage scenarios that the customer plans, intends or specifies do not comply with applicable laws, statutes, regulations and/or standards, in particular, but not limited to, vehicle emission standards (hereinafter collectively "***Legal Requirements***"). It is the sole responsibility of the customer (i) to configure and use the DEM and its interfaces in such a way that implementation and use of the DEM comply with all applicable Legal Requirements, as amended from time to time, and (ii) to take all measures required by such Legal Requirements for the operation and distribution of the customer system in which the DEM is implemented, in particular, but not limited to, obtaining approvals under regulatory procedures prescribed by Legal Requirements.

2 Emission OBD

2.1 Introduction

Due to heavy smog on the US West Coast in the 1980s the CARB (California Air Resources Board) passed a law prescribing on-board monitoring diagnostic for motor vehicles. This led to the introduction of OBD-I which was replaced by OBD II in the 1990s. In that time also Europe understood the advantages of OBD II. So in 2001 EOBD which is derived from OBD II was introduced in Europe. Since 2001, many other countries have added OBD requirements which are mainly derived from OBD II or EOBD to their regional legislative standards.

The “World Forum for Harmonization of Vehicle Regulations” decided in 2001 to establish a global technical regulation (GTR) for on-board emission diagnostic system for heavy duty vehicles and engines. The idea was to expand this regulation to passenger cars (light duty) as well at a later point in time. One goal of WWH-OBD was to replace the regional legislative standards by a global OBD regulation.

In the year 2021, the SAE (Society of Automotive Engineers) International published “SURFACE VEHICLE STANDARD: E/E Diagnostic Test Modes: OBDonUDS” as J1979-2. This standard will be called OBDonUDS below.

Table 2-1 provides an overview of the standardizations used for the diagnostic communication.

Legislation	SAE International	ISO
OBDII	J1979 ([6])	ISO-15031-5 ([7])
WWH-OBD	-	ISO-27145-3 ([10]), based on 14229-1([9])
OBD on UDS	J1979-2 ([11])	

Table 2-1 Standardizations for legislative diagnostic communication

Chapters without one of the icons below handle all three legislations, OBD II, OBDonUDS and WWH-OBD.

Dem also supports Zero Emission Vehicle OBD (ZEVonUDS) based on SAE J1979-3 which is described in a separate chapter (see Zero Emission Vehicle OBD).

OBDII	This symbol marks a chapter or passage which deals with **OBD II**.
OBD on UDS	This symbol marks a chapter or passage which deals with **OBDonUDS**.
WWH-OBD	**WWH-OBD** related content is marked with this symbol.

2.1.1 Comparison of OBD II, OBDonUDS and WWH-OBD

As this document describes OBD II, OBDonUDS and WWH-OBD the following Table 2-2 shall provide an overview of the main differences and commonalities between those three legislative implementations.

Legislation Topic	OBDII	OBD on UDS	WWH-OBD
DTC byte count	2 byte	3 byte	3 byte
DTC Class	-	■	■
Permanent DTC	■	■	■
Legislative Freeze Frame 0x00	■	■	■
Legislative Freeze Frame 0xF0	-	■	-
Activation Mode	-	-	■
DTC Healing	3 DCYs	3 DCYs	3 DCYs
DTC Aging	40 WUCs	40 WUCs	40 WUCs or 200h
PIDs/ DIDs provided from Dem	$01, $02, $1C, $21, $30, $31, $41, $4D, $4E	$F401, $F41C, $F421, $F430, $F431, $F441, $F44D, $F44E, $F501	$F401 (Byte 0 always 0), $F41C, $F421, $F430, $F431, $F441, $F44D, $F44E, $F491
Legislative Extended data records provided from Dem	-	$91, $92, $93	-
Diagnostic Communication with Dem	Mode $01-$0A	Service $14, $19, $22	Service $14, $19 and $22
In-Use-Monitor Performance Ratio	■	■	■
Diagnostic Test Result	Mode $06	Service $19-06 Erec $92, Service $22 DID $F6xx	Service$22 DID $F6xx
Monitor Activity Ratio	■	Service $19-06 Erec $93, Service $22 DID $F879	■

Table 2-2 Comparison of OBD II, OBDonUDS and WWH-OBD

Note
The Dem can be configured to support OBD II, OBDonUDS and WWH-OBD in same configuration (see 2.4.4 Post-Build Selectable). But it is not supported that more than one legislation is active at the same time.

2.2 Functional Description

2.2.1 Features

This Dem supports the OBD functionality specified in [2], [6], [7], [9], [10] and [11].

The AUTOSAR OBD standard functionality specified in [2] is completely supported except features described in

- Table 2-3 Not supported AUTOSAR standard conform OBD features
- Table 2-4 Not Supported APIs
- Table 2-5 Service Interfaces which are not supported
- Table 2-6 Callbacks which are not supported.

2.2.1.1 Deviations

The following OBD functionality specified in see [2] is not or partially supported from Dem.

Category	Description
Functional	Usage in an OBD Master ECU and OBD Master ECU exclusive Dem functionality like Pfc- and IUMPR cycle management are not supported.
Functional	PID $ 1E not supported
Config	For details about not supported or differently named configuration elements or different configuration structure please refer to the Module Parameter Description (BSWMD).
Config	Multiplicity or ranges of some configuration elements is restricted. For details, please refer to the Module Parameter Description (BSWMD).
Functional	Dem services may report different error code identifier for Det errors.
API	API return values, reentrancy property and service IDs may differ. For details on the supported return values please refer to chapter 4.1 Provided Interfaces.
Functional	Method for grouping of association of events for OBD purpose [ch. 7.7.4.1]: feature named Mil Groups.
Functional	Centralized PID $21 / $31 / $4D / $4E handling [ch. 7.9.2.1]: - Dem uses odometer value instead of PID $0D (= Vehicle speed) to calculate PID $21 and PID $31. - Service Needs are not provided. - PID $21 / $31 / $4D / $4E not provided as Dem internal data element
Functional	In-Use-Monitor Performance Ratio (IUMPR) Support [ch. 7.9.4]: InfoType $08 / InfoType $0B not supported. Instead use application interface (see 2.3.7.3).

Category	Description
Functional	Service $06 - Support of central DTR handling [ch. 7.9.5.5]: Event's enable- and storage conditions are not respected to decide if reporting DTR values are processed if DemDtrUpdateKind == DEM_DTR_UPDATE_ALWAYS.
Functional	Notification of data changes [ch. 7.7.7.5]: No event data change callback for OBD-II legislative freeze frame is called.

Table 2-3 Not supported AUTOSAR standard conform OBD features

2.2.1.1.1 Not supported APIs

The following APIs specified in [2] are not supported from Dem.

Name
Dem_GetIUMPRDenCondition()
Dem_DcmGetInfoTypeValue08()
Dem_DcmGetInfoTypeValue0B()
Dem_SetDataOfPID<NN>()
Dem_GetDataOfPID21(), instead use Dem_DcmReadDataOfPID21()

Table 2-4 Not Supported APIs

2.2.1.1.2 Not supported Service Interfaces

The following service interfaces specified in [2] are not supported from Dem.

Name	Operation(s)
GetDataOfPID21	GetDataOfPID21
IUMPRDenominatorCondition	GetIUMPRDenCondition, please note that SetIUMPRDenCondition is supported.
SetDataOfPID<NN>	SetDataOfPID<NN>

Table 2-5 Service Interfaces which are not supported

2.2.1.1.3 Not supported Callbacks

The following callback functions specified in [2] are not supported from Dem.

Name
OBDDTCStatusChanged

Table 2-6 Callbacks which are not supported

2.2.1.2 Limitations

Limitation	Comment
Diagnostic Test Results (DTR)	DemDtrCompuDenominator0 and DemDtrCumpuNumerator1 cannot be configured to negative values or zero.

Limitation	Comment
Visibility of OBD freeze frame	The visibility of OBD freeze frame in $19 04 is independent of the configured Mode $02 visibility. Therefore, the legislative freeze frame can be reported in $19 04 although it is not visible yet in Mode $02.
Clear single DTC	If the selected format is equal to OBD only the wildcard DTC 0xFFFFFF is allowed. If a single DTC shall be cleared the given format must be UDS.
OBD II DTC without a UDS DTC	Each OBD II DTC must refer at least to one UDS DTC.
Number of J1939NmNodes for OBD II and J1939	For OBD II and J1939 only one J1939NmNode can be configured if clearing active DTCs, reading Monitor Performance Ratios or reading of Diagnostic readiness Information is supported (see chapter 2.2.16 J1939).
Clearing previously active J1939 DTCs for OBD II and J1939	For OBD II and J1939 clearing previously active J1939 DTCs (DM3) is not supported.
Node specific calculation of Diagnostic Readiness	Diagnostic readiness 1 (DM5), 2 (DM21) and 3 (DM26) is not calculated node specific.
Events affected by Power Take-Off	Events which are affected by PTO can only be configured to use Monitor Internal Debouncing.
Additional conditions for completeness calculation	The Dem does not consider additional conditions for completeness (e.g. whether 4,000 engine revolutions have occurred).

Table 2-7 Limitations

2.2.1.2.1 Incompatible Features

2.2.1.2.1.1 Reset TestFailed Bit On Operation Cycle Start

The option ‘Reset TestFailed Bit On Operation Cycle Start’ cannot be enabled in an WWH-OBD configuration. The parameter DemGeneral/DemResetTestFailedOnOperationCycleStart must be set to False.

2.2.1.2.1.2 WWH-OBD and J1939

Features WWH-OBD and J1939 are not allowed to be enabled at the same time.

2.2.1.2.1.3 OBDonUDS and J1939

If both, J1939 and OBDonUDS, are licensed, only DM01, DM02 are supported. Other diagnostic messages may be rejected, or the response message may be incorrect.

2.2.1.2.1.4 OBD II, OBDonUDS and WWH-OBD

Only one of the legislations OBD II, OBDonUDS and WWH-OBD can be supported in a single configuration variant. However, it is possible to have multiple configuration variants with different OBD legislations in parallel (see chapter 2.4.4).

2.2.1.2.1.5 Event Combination Type 2

The usage of event combination type 2 in combination with the following features is currently not supported: WWH-OBD, OBDonUDS, OBD II Freeze Frame behavior other than type 1, reporting of suppressed DTCs in mode $0A.

2.2.1.3 Uncommon Feature Behavior

2.2.1.3.1 Event Status Changed Callback

The event status changed callback will be triggered twice during driving cycle restart for OBD related combined events.

2.2.1.3.2 ResetEventStatus

This chapter is only relevant for **WWH-OBD**.

The API `Dem_ResetEventStatus()` will not accept WWH-OBD events if they are in state ConfirmedDTC. In this case the API will return E_NOT_OK.

2.2.1.3.3 DTC Suppression

DTC Suppression has no effect in calculation of PID $01 and PID $41.

A DTC requesting an indicator (e.g. MIL) will still request that indicator if the DTC is suppressed. Suppression only hides the DTC from the tester. This can lead to the ECU requesting the MIL even without reporting a DTC in Mode $03.

UDS and OBD services will not report the OBD freeze frame if the corresponding DTC for the stored and visible freeze frame is currently suppressed. This also applies if internal other OBD freeze frames are stored by DTCs that are not suppressed.

For these reasons it is discouraged to use DTC suppression for OBD related DTCs. Consider using API `Dem_SetEventAvailable()` instead.

2.2.1.3.4 Warm-Up Cycle

If the Dem is configured to delay the warm-up cycle (see chapter 2.2.2.4.1) to the end of the driving cycle, the warm-up cycle cannot be used as an operation cycle according to AUTOSAR [2]. In the very first driving cycle in which the DEM receives a start trigger for the warm-up cycle from application, the state of the warm-up cycle will remain STOPPED until the end of the driving cycle. In the following driving cycles, the warm-up cycle state will always be reported as STARTED, i.e., even before the start trigger from the application is received. Thus, a DTC using the warm-up cycle as operation cycle can be reported anytime during the driving cycle.

2.2.1.3.5 OBD Ratio (internal data element) attached to a combined event

See [1] for more information.

2.2.1.3.6 Operation SetEventDisabled of Service Interface Diagnostic Monitor

The operation SetEventDisabled must be called from the Dem's master partition. If you want to use the operation for an event, ensure that the event is assigned to the master partition (see DemGeneral/DemMasterOsApplicationRef). Do not use the operation, if it is offered by a DemSatellite SWC running on a partition other than the master partition (see [1]).

2.2.1.3.7 Requesting PID data in an OBD configuration without OBD events

If Dem supports OBD (DemGeneral/DemOBDSupport="PRIMARY_ECU") but no OBD events are configured, requesting PID values using PID $01, PID $41 or PID $F501 (see

DemGeneral/DemPidConfiguration) may return the comprehensive component monitoring bit as enabled, even though no OBD events are configured.

2.2.2 Operation Cycles

To fulfill legislation requirements, AUTOSAR defines some restrictions and extensions to the standard operation cycle handling.

2.2.2.1 Driving Cycle

The OBD driving cycle (DCY) is the operation cycle for an OBD related event defined by legislation. Different to other operation cycles the driving cycle is always started and can only be restarted.

In order to restart the DCY, it must have been qualified, i.e. the vehicle must have been operated under normal conditions (specified by legislative regulations) since the last restart. Restarting the DCY without a previous qualification is not recognized by the Dem.

The qualification state of the DCY is returned by API `Dem_GetOperationCycleState` (see [1]).

The conditions needed to qualify the DCY are usually provided by the Master OBD ECU and distributed over the bus system.

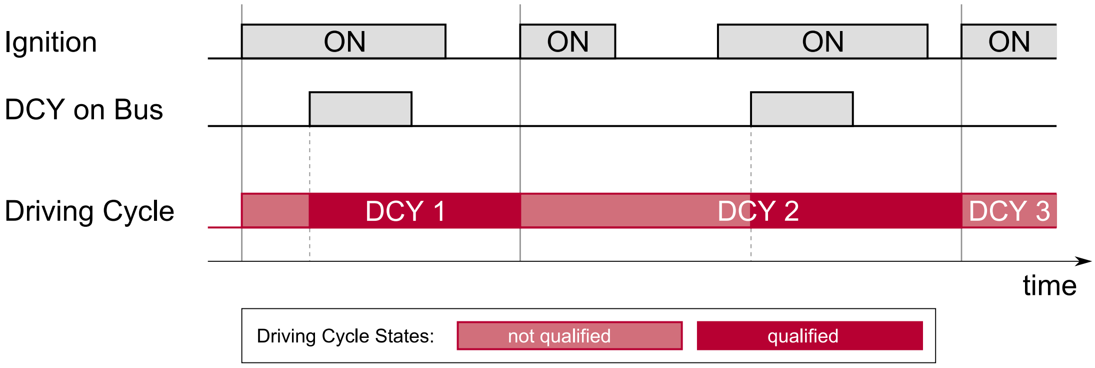

Figure 2-1 OBD Driving Cycle Behavior

2.2.2.2 Ignition Cycle

The OBD ignition cycle is the time between ”key on” and ”key off” after the driving cycle is qualified. This time includes also the shutdown phase of the ECU.

2.2.2.3 Ignition Cycle for Hybrids

For hybrid vehicles an additional ignition cycle must be provided. It indicates that the combustion engine is active.

Note
The hybrid ignition cycle is only intended as trigger for the respective cycle counter and not for usage as operation cycle in context of AUTOSAR.

2.2.2.4 Warm-Up Cycle

The OBD warm-up cycle (WUC) is a phase during engine startup and depends on the coolant temperature. It is calculated by the Master ECU and distributed over the bus system. Primary ECUs must use this information to start their internal warm-up cycle.

2.2.2.4.1 Delayed Warm-Up Cycle End

Using this configuration option, starting the warm-up cycle by application will mark the driving cycle as having met the warm-up conditions. The DTC aging is processed at the end of this driving cycle during which the warm-up conditions have been met (refer to “WUC Delayed” in Figure 2-2). Without using this option, the warm-up cycle must be ended explicitly to process DTC aging (see chapter 2.3.1.2).

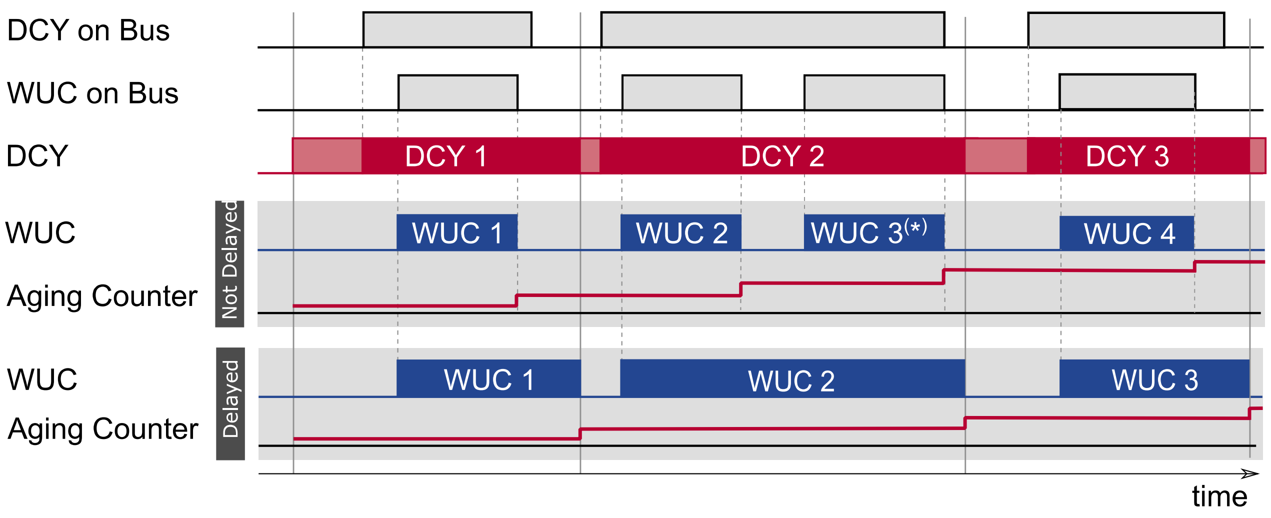

Figure 2-2 Warm-up Cycle Behavior

Caution
(*) Only relevant if WUC is not delayed to the end of the DCY.
Since multiple WUCs within a DCY are not allowed, the application must ensure to start the WUC only once during a DCY.
If WUC is delayed to the end of DCY, DEM will automatically ensure that there is only one WUC during a DCY.

2.2.3 DTCs and Events

For OBD II, DTCs with specific 2 byte DTC numbers are defined to access emission related diagnostic events. Only these DTCs can be accessed by the OBD II diagnostic tool.

Thereto an OBD II relevant Dem event is not only referenced by an UDS DTC, but also by a 2 byte OBD DTC.

Since OBDonUDS and WWH-OBD use the UDS protocol for diagnostic communication the 3 byte UDS DTC is used as OBD DTC number.

Note
For OBDonUDS, it is also possible to configure a separate OBD DTC number (see chapter 2.2.3.1).

Figure 2-3 shows the different types of Dem events and their relation to DTCs.

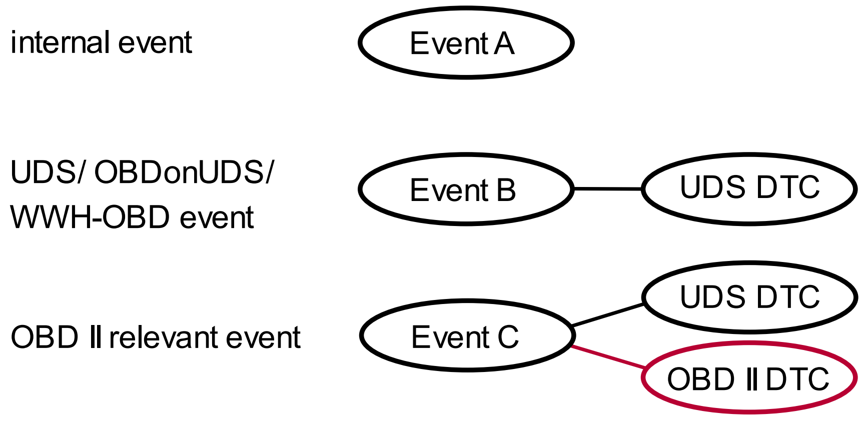

Figure 2-3 Relation between Events and DTCs

Additionally to these basic relations, Dem also supports DTC separation and DTC grouping which is described in the following subchapters.

2.2.3.1 OBD UDS DTC Separation

This chapter is only relevant for **OBDonUDS**.

The DTC separation feature allows a separate 3 byte OBD DTC to be reported for the following OBDonUDS services (refer to [11]):

- $19 04
- $19 06
- $19 1A
- $19 42
- $19 55
- $19 56

With the feature enabled, all OBD relevant events must be configured with the 3 byte OBD DTC.

This DTC is selected and reported separately from the UDS DTCs by calling the relevant services and related Dem APIs with format:

- DEM_DTC_FORMAT_OBD_3BYTE

Note
The DTC format DEM_DTC_FORMAT_OBD_3BYTE is only available with the feature enabled and only for the above stated services.

2.2.3.2 DTC Combination

OBD II

This chapter is only relevant for **OBD II**.

Combined DTCs means, that one OBD II DTC refers to multiple OBD relevant events, where each has its own UDS DTC.

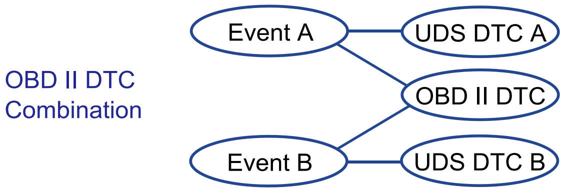

Figure 2-4 Relation between Events and DTCS with DTC combination

The combination affects the reporting with the OBD Modes $03, $07 and $0A. An OBD DTC is reported if at least one referenced Event / UDS DTC is confirmed ($03), pending ($07) or permanent ($0A).

Note
Because the internal handling of the failure codes is mainly based on events, with DTC combination single OBD DTCs may be reported several times.

2.2.3.3 MIL Groups

OBD related Events that are attached to a MIL Group share a common trip counter, i.e., the MIL indicator status and the confirmed status is calculated from the failed test result of all associated events. Still for the confirmation of each individual event at least one failed result from its respective monitor is required when the common trip counter has reached the trip target.

The results of associated events only affect the detection of a failure. Healing and Aging are still performed individually for each event.

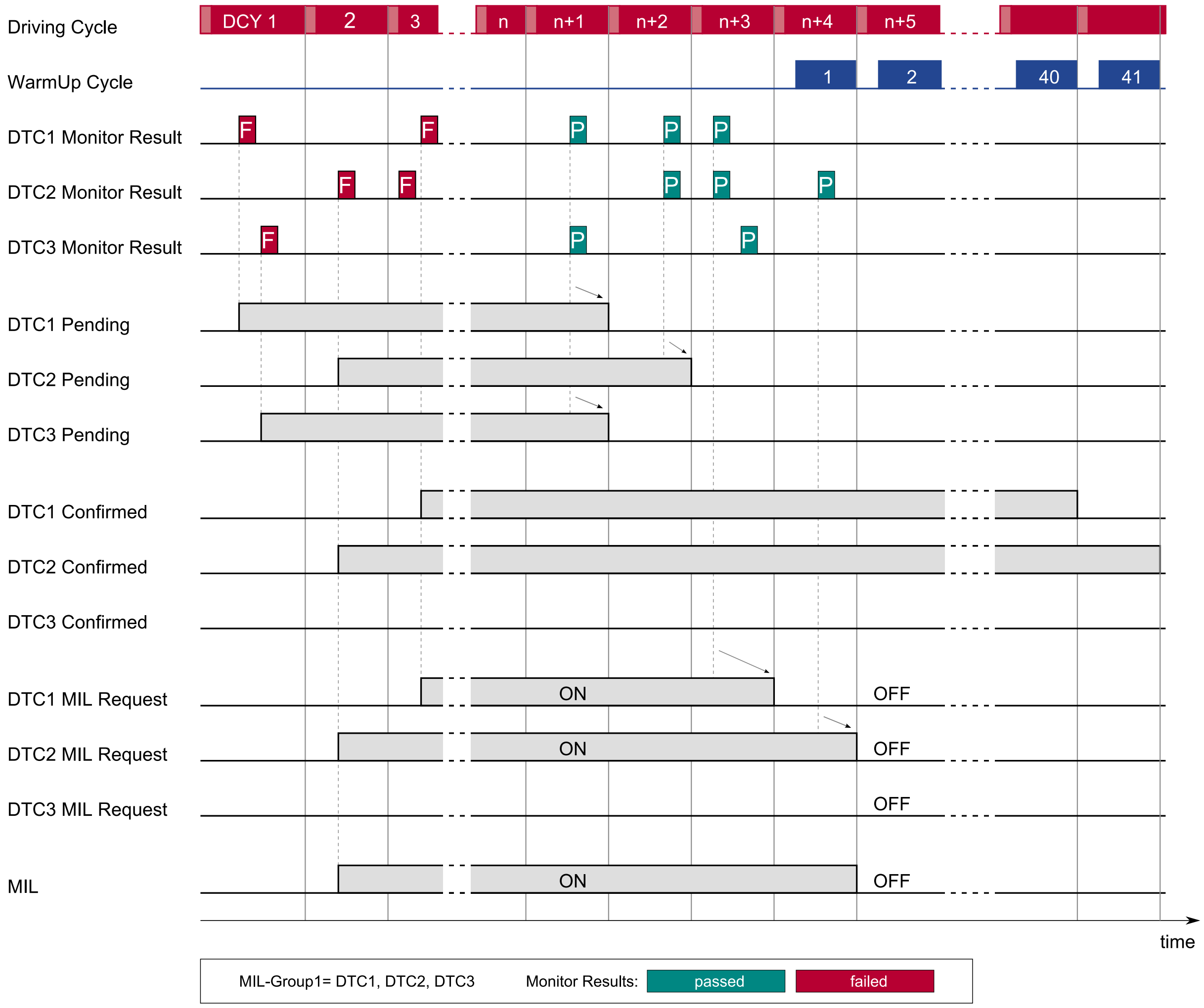

Figure 2-5 MIL Group behavior

In Figure 2-5 the following preconditions are assumed:

- Two-trip behavior
- Healing after 3 driving cycles
- Aging after 40 warm-up cycles.

The following applies to the common MIL group trip counter:

- Until the trip target is reached, the counter is incremented at the end of the driving cycle, if at least one associated event is pending and was tested failed during this driving cycle.
- The counter is reset at the end of the driving cycle if none of the associated events is pending. I.e., if the trip target is reached and at least one of the associated events is still pending, a single failed result for another associated event leads to the confirmation of this other event.

Note

For the configuration of MIL groups, the following restrictions apply:

- The trip counter is still configured individually for each event. To avoid an unpredictable tripping behavior, all associated events must be configured with the same trip target.
- At most 127 events can be associated to one MIL group.

2.2.3.4 Event Combination

This chapter is only relevant for **OBDonUDS**.

For OBDonUDS it is supported to combine the results of multiple events to a single DTC on storage with event combination type 1 or type 3 (see [1]).

If event combination type 1 is enabled in an OBDonUDS configuration, the DTC status is derived from the status of multiple events as defined by Autosar (see [1]).

If event combination type 3 is enabled in an OBDonUDS configuration, the DTC status of an OBD related DTC is calculated according to [11]. As described in Table 2-8, the Pending-, Confirmed- and WIR bit are not an OR combination of all events.

Combined DTC Status Bit	
Bit 0 – TestFailed	OR (Event[i].Bit0)
Bit 1 – Test Failed This Operation Cycle	OR (Event[i].Bit1)
Bit 2 – PendingDTC	No combination/DTCStatusbit2
Bit 3 – ConfirmedDTC	No combination/DTCStatusbit3
Bit 4 – Test not Completed Since Last Clear	OR (Event[i].Bit4) AND NOT Bit5
Bit 5 – Test Failed Since Last Clear	OR (Event[i].Bit5)
Bit 6 – Test not Completed This Operation Cycle	OR (Event[i].Bit6) AND NOT Bit1
Bit 7 – Warning Indicator Requested	No combination/DTCStatusbit7

Table 2-8 OBD related DTC status calculation with event combination type 3

All events that are combined to a single OBD related DTC share a common trip counter and a common healing counter, which are used to calculate the Confirmed and WIR bit of the DTC’s status (refer to [11]).

The common trip counter is calculated the same way as the trip counter that is shared by a MIL group (see chapter 2.2.3.3). Once the common trip counter reaches the trip target, the OBD related DTC becomes confirmed.

The common healing counter is incremented when all events of the combination group have been tested only ‘passed’ in the same operation cycle. When the common healing counter reaches the healing target, the OBD related DTC is healed and its WIR bit is reset.

The Pending bit of DTC status is set if at least one event of the combination group has tested 'failed'. For the Pending bit of DTC status to be reset, all events of the combination group must be tested only 'passed' in the same operation cycle.

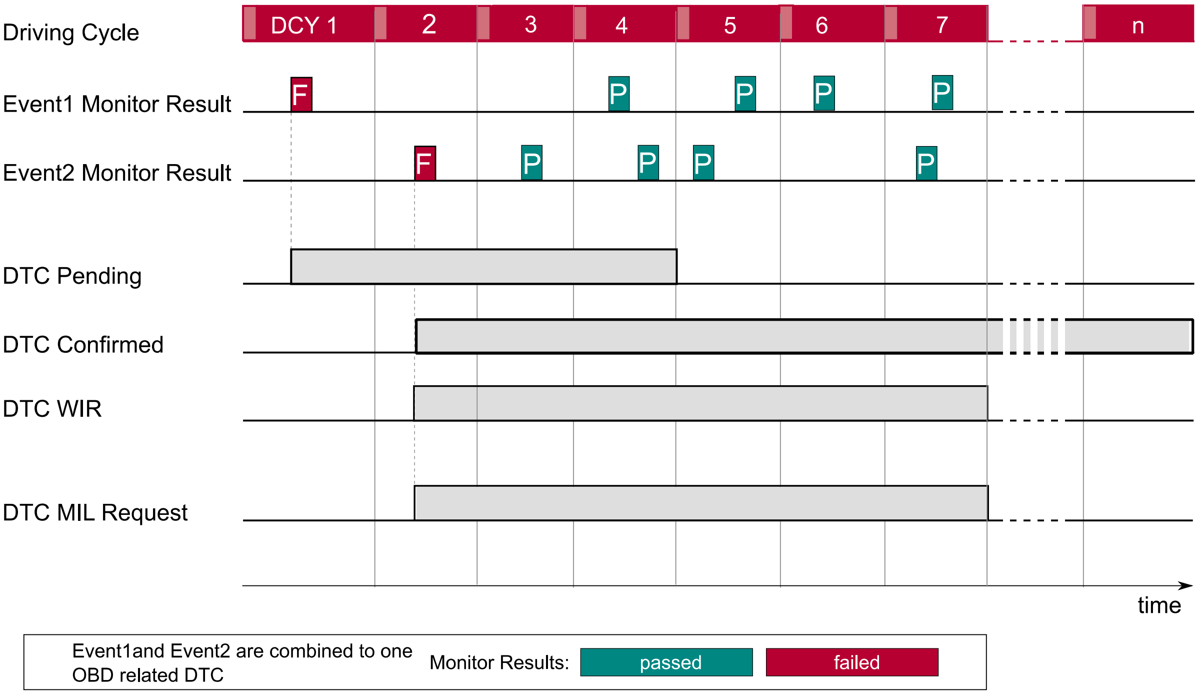

Figure 2-6 Combined DTC status of event combination type 3

In Figure 2-6 the following preconditions are assumed:

- Two-trip behavior
- Healing after 3 driving cycles

Note

In an OBDonUDS configuration with event combination type 3, the following restrictions apply if multiple events are combined to an OBD related DTC:

- At most 127 events can be combined to the OBD related DTC and all events must have the same trip target.
- The OBD related DTC is not allowed to share a trip counter with other DTCs.
- All events that are combined to the same OBD related DTC must have the same healing target and the healing target must be greater than 0.

Note

In an OBDonUDS configuration with event combination type 3, the following applies to the status of the events which are combined to an OBD related DTC:

- When an event sets the WIR bit in its event status, the WIR bit is not reset until the DTC is healed.
- Pending bit in each combined event's status is only reset if the DTC Pending bit can be reset.

2.2.3.5 DTC Class

This chapter is only relevant for **WWH-OBD** and **OBDonUDS**.

DTCs can be classified dependent on their effect to the emissions if the fault occurs.

In case of WWH-OBD, the DTC class of a fault directly influences the malfunction indicator illumination scheme which can be derived from the activation mode described in chapter 2.2.7.

Naming Convention		
WWH-OBD DTC Class	**OBDonUDS DTC Class**	**Classification**
A	1	If this DTC occurs it is assumed that the OBD threshold limits (OTL) are exceeded.
B1	2	The DTC has the potential to lead to emissions above the OBD threshold limits.
B2	3	The malfunction is assumed to influence the emissions but not in a level exceeding the OBD threshold limits.
C	4	The DTC is assumed to influence the emissions but not in a level above the regulated emission limits.
None	0	The DTC does not influence the emissions.

Table 2-9 DTC Class

2.2.3.6 Dependent Secondary ECU DTCs

Using the primary/secondary dependent ECU concept, the Dem of the primary ECU has to administrate the OBD related faults from the dependent secondary ECUs. These faults shall be available in the diagnostic services used for OBD.

For some use cases these dependent DTCs shall not appear in UDS $19 sub-services. The Dem uses the configuration parameter *DemEventSignificance* to classify the event. Events which are configured as *Occurrence* will be visible in the different diagnostic services as defined in Table 2-10.

Dependent Secondary ECU DTC Availability Diagnostic Service	Visible/Accessible	Not Visible	NRC 0x31
$19 01 – ReportNumberOfDTCByStatusMask		■	
$19 02 – ReportDTCByStatusMask		■	
$19 03 – ReportDTCSnapshotIdentification		■	
$19 04 – ReportDTCSnapshotRecordByDTCNumber	■		
$19 05 – ReportDTCSnapshotRecordByRecordNumber	■		
$19 06 – ReportDTCExtendedDataRecordByDTCNumber	■		
$19 07 – ReportNumberOfDTCBySeverityMaskRecord		■	
$19 08 – ReportDTCBySeverityMaskRecord		■	
$19 09 – ReportSeverityInformationOfDTC			■
$19 0A – ReportSupportedDTC		■	
$19 0B – ReportFirstTestFailedDTC		■	
$19 0C – ReportFirstConfirmedDTC		■	
$19 0D – ReportMostRecentTestFailedDTC		■	
$19 0E – ReportMostRecentConfirmedDTC		■	
$19 12 – ReportNumberOfEmissionsRelatedOBDDTCByStatusMask	■		
$19 13 – ReportEmissionsRelatedOBDDTCByStatusMask	■		
$19 14 – ReportDTCFaultDetectionCounter		■	
$19 15 – ReportDTCWithPermanentStatus	■		
$19 16 – ReportDTCExtDataRecordByRecordNumber	■		
$19 1A – ReportSupportedDTCExtDataRecord	■		
$19 42 – ReportWWHOBDDTCByMaskRecord		■	
$19 55 – ReportWWHOBDDTCWithPermanentStatus	■		
$19 56 – ReportDTCInformationByDTCReadinessGroupIdentifier	■		
$14 – ClearDiagnosticInformation	■		

Table 2-10 Dependent Secondary ECU DTC behavior in diagnostic services

In addition the Dem provides the services interface `Dem_SetHideObdOccurrences()` to allow the application to overwrite[1] the predefined behavior of the diagnostic services listed

[1] Diagnostic services which behaves per default as „Not Visible“ or „NRC 0x31“ can be set to „Visible/Accessible“

above. This feature also allows the implementation of user defined diagnostic services whereas the application can decide if the predefined behavior shall be overwritten[1] or not.

2.2.3.7 Black MIL DTCs

The Dem supports the configuration of OBD relevant DTCs that do not trigger the MIL. If such a DTC is tested as failed it will store a freeze frame and it becomes also available in Mode $03, Mode $07, Service $19-13 and Service $19-42.

Since a black MIL DTC cannot trigger the MIL indicator no permanent DTC will be stored and therefore no respective Mode $0A, Service $19-15 nor Service $19-55 entry is available.

Request emission-related DTCs:

- Mode $03 (OBD II),
- Service $19-13 with CDTC (UDS),
- Service$19-42 (WWH-OBD, OBDonUDS).

Request emission-related DTCs detected during current or last completed driving cycle:

- Mode $07 (OBD II),
- Service $19-13 with PDTC (UDS),
- Service $19-42 (WWH-OBD, OBDonUDS).

Request emission-related DTCs with permanent status:

- Mode $0A (OBD II),
- Service $19-55 (WWH-OBD, OBDonUDS).

A black MIL DTC always needs to heal before aging can start (if they have their ConfirmedDTC status bit set). This also holds for black MIL DTCs not assigned to any indicator.

WWH-OBD

Note

In case of WWH-OBD a DTC is considered as black MIL DTC, if it has DTC class ‘None’ and a legislative freeze frame configured and is not assigned to the MIL indicator.

Caution

The legislation requires that OBD relevant DTCs trigger the MIL as soon as they become confirmed DTCs. Nevertheless, in some rare special cases and after consultation of the authority a black MIL DTC may be supported in the OBD system.

2.2.3.8 Power Take-Off (PTO)

Events can be configured to be affected by Dem PTO handling. The Dem supports the handlings described below for the events.

2.2.3.8.1 Consider PTO status

While PTO is active, the following processing is disabled for the events, which are affected by PTO:

- Event reporting via `Dem_SetEventStatus()`, i.e., no event status reports will be accepted
- Incrementing of IUMPR counters (refer to chapter 2.2.11.4)
- Changing Event availability via `Dem_SetEventAvailable()`.

2.2.3.8.2 PTO Activation Timer

For each affected event which is assigned to a readiness group, the Dem maintains a timer which calculates the cumulative engine runtime with PTO active since the last qualified test result of the event. The value of the timer latches at 750 minutes. The timer is reset to 0 in the following cases:

- Event is reported with a qualified test result
- Event is set to unavailable
- Event is cleared by ClearDTC.

The PTO activation timer affects the readiness calculations for:

- PID $01 and PID $F501: Monitor status since DTCs cleared (see 2.2.9.1)
- Diagnostic Message 5: Diagnostic Readiness 1 (see 2.2.16.1.1).

2.2.4 Behavior of OBD Relevant Events

2.2.4.1 General Behavior

The behavior of OBD relevant events differs in some aspects from the behavior specified by UDS, especially concerning event status bit transitions. Figure 2-7 gives an overview of the OBD behavior.

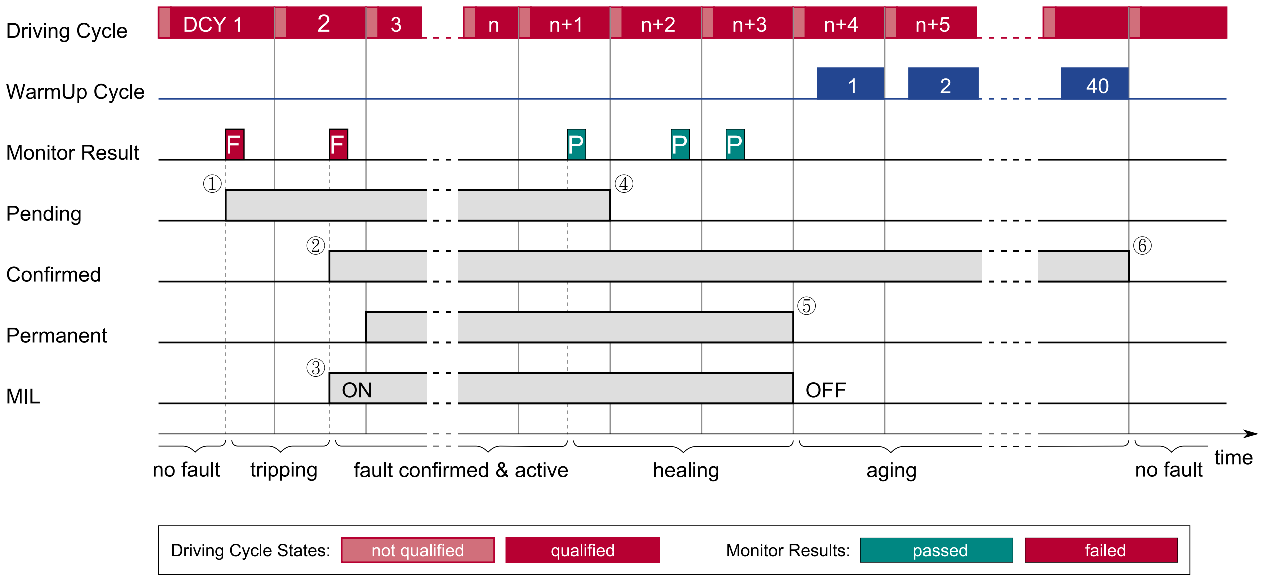

Figure 2-7 OBD failure life cycle

The following preconditions are assumed:

- Two-trip behavior
- Healing after 3 driving cycles
- Aging after 40 warm-up cycles
- No other event commands the MIL indicator
- Neither DTC Combination nor MIL Groups are used. For OBDonUDS, the event is not a combined event.
- OBD Permanent DTC behavior: storage at driving cycle restart

① The event becomes pending if it is reported failed for the first time.

② The failure becomes confirmed if the trip condition is fulfilled (DCY 2 in the example). OBD relevant events require a two-trip behavior (three-trip in EU), i.e. the event must be reported as failed for two (resp. three) driving cycles.

③ The confirmed failure activates the MIL indicator and triggers the permanent DTC storage at the end of the driving cycle.

④ When the failure condition disappears and the event is only reported passed (and not failed) for one complete driving cycle, the pending status is reset at the end of the DCY.

⑤ After the configured number of healing cycles (3 DCY in the example), the MIL indicator is deactivated, the permanent DTC is erased and aging starts.

⑥ Aging is not performed with driving cycles but with warm-up cycles (see 2.2.2.4). After at least 40 warm-up cycles without a detected failure, the confirmed bit is reset at the end of the DCY and any related stored data (snapshot, extended or freeze frame data) is erased.

Additional to the aging trigger after 40 warm-up cycles WWH-OBD ages the fault after 200h engine operating hours where the aging conditions have been met.

Note

A confirmed OBD DTC (with WIR Bit set) is healed first which results in resetting of the WIR bit, before it starts to age. Based on configuration, it typically takes 3 driving cycles to heal and followed by 40 warm-up cycles to age the event.

If a confirmed OBD DTC does not have its WIR bit set (for example: when the OBD DTC could not be stored in the memory and DemWarningIndicatorRequestedProcessing = STORED_ONLY, OR in case of BlackMIL DTCs) it must always be healed first before the aging starts. This also holds true, while restarting a previously interrupted aging.

2.2.4.2 Failure Confirmation

Figure 2-8 gives a more detailed view of the event confirmation and the impact of the driving cycle status (qualified / not qualified). The confirmed bit, as well as the warning indicator bit, is always set immediately if the trip condition is fulfilled. However, for each event it is possible to configure whether those status bit changes are externally visible immediately of if they are suppressed until the DCY is qualified. External visibility affects the reporting of OBD DTCs via Mode $03 as well as the reporting of UDS DTCs via Service $19. Note that related status changes (such as Mode $0A reporting (see chapter 2.2.6.1.2), MIL request and Mode $02 reporting of the legislative freeze frame (see chapter 2.2.5.1.2)) are also suppressed if the confirmed bit or the warning indicator bit are suppressed.

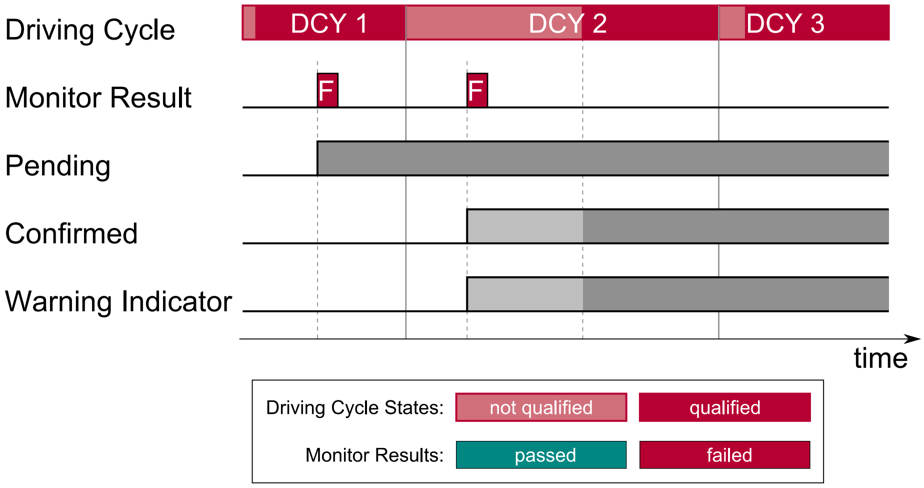

Figure 2-8 Confirmation for OBD relevant events (with delayed visibility)

2.2.4.3 Event Displacement

OBDII

OBD on UDS

This chapter is only relevant for **OBD II** and **OBDonUDS**.

An OBD relevant event results in an active OBD DTC if it:

- has the pending bit set Or
- currently requests the MIL indicator Or
- in case of OBD II legislation stores the mode $02 freeze frame.

In addition to the event displacement described in [1] an event which results in an active OBD DTC will only be displaced

- if its memory slot was reallocated for aging only Or
- if its priority is lower than the priority of an event, which needs to be stored.

If the fallback option to displace the oldest currently stored event is enabled, an event with an active OBD DTC will not be considered for displacement, even if it is the oldest currently stored event.

For a displaced OBD relevant event not only the event memory entry including snapshot and extended data records will be removed, but also the following can happen:

- If the event holds an OBD freeze frame, this OBD freeze frame will be cleared. For legislation OBDonUDS this is only done, if the event doesn't have the pending or confirmed bit set (see chapter 2.2.5.2.2).
- If event combination type 2 is disabled (see [1]) and if the event has stored a permanent DTC, this permanent DTC will be set to the same state as after a clear request (see chapter 2.2.6.1.4).

2.2.4.4 Similar Conditions

For an OBD related event similar conditions are met when the same conditions (e.g. engine speed) are given as when the event became pending.

If similar conditions are not met for an OBD related event, the Dem will not increment the trip count or reset the pending state for this event. As a result, the event will neither become confirmed nor heal without similar conditions. However de-bouncing and IUMPR processing are not affected by similar conditions. Also, an event with failure cycle counter threshold 0 becomes confirmed with the first failed result, independently of similar conditions.

Similar conditions can be reported by API `Dem_SetEventStatus` (see [1]). While similar conditions are not met, the following values for the monitor status shall be used:

- DEM_EVENT_STATUS_PASSED_CONDITIONS_NOT_FULFILLED
- DEM_EVENT_STATUS_FAILED_CONDITIONS_NOT_FULFILLED

- DEM_EVENT_STATUS_PREPASSED_CONDITIONS_NOT_FULFILLED
- DEM_EVENT_STATUS_PREFAILED_CONDITIONS_NOT_FULFILLED

As soon as similar conditions are met in the current driving cycle, the monitor status shall be reported with the following values:

- DEM_EVENT_STATUS_PASSED
- DEM_EVENT_STATUS_FAILED
- DEM_EVENT_STATUS_PREPASSED
- DEM_EVENT_STATUS_PREFAILED

Figure 2-9 gives a more detailed view of similar conditions and the impact on the trip count and pending state of an event.

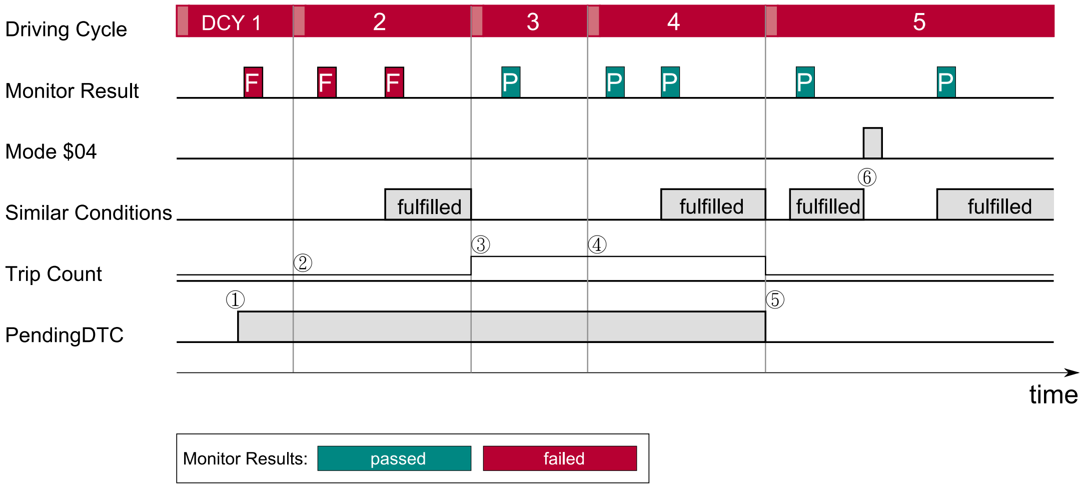

Figure 2-9 Similar conditions

① The event becomes pending if it is reported failed for the first time.

② The trip count is not incremented at the end of the driving cycle because similar conditions have not been met.

③ Since similar conditions have been met in the current driving cycle (reported with the second failed result) the trip count is incremented.

Similar conditions are reset on restart of the driving cycle.

④ The pending state of the event is not reset at the end of the driving cycle because similar conditions have not been met during the current cycle.

⑤ Since similar conditions have been met in the current driving cycle (reported with the second passed result) the pending state is reset and the trip count is reset.

Similar conditions are reset on restart of the driving cycle.

⑥ Similar conditions are reset on a clear request or on disconnection of the event.

They have to be reported again in the same driving cycle.

Caution
Similar conditions are only evaluated for OBD related events. Don't report the monitor status of other events with the values 'DEM_EVENT_STATUS_xxx_CONDITIONS_NOT_FULFILLED'.

Note
With regard to confirmation or healing, only the last reported similar condition state (fulfilled or not fulfilled) in the event's operation cycle is taken into account. If an event report with fulfilled conditions shall be taken into account, do not report unfulfilled conditions afterwards.

2.2.5 Legislative Freeze Frame

The legislative freeze frame includes different vehicle parameters like engine coolant temperature, engine speed or fuel pressure. The complete list is specified by the legislative regulation and depends on the vehicle type.

Note that the Data elements which are calculated by Dem (e.g. PID $01, PID $21 cannot be mapped to the freeze frame.
In the three supported legislations OBDII, OBDonUDS and WWH OBD, the legislative freeze frame is stored, reported, reset and potentially displaced in accordance to different rules. Therefore, this chapter provides subchapters for the different legislations.

2.2.5.1 OBD II

OBDII This chapter is only relevant for **OBD II**.

2.2.5.1.1 Storage

Dependent on the configuration either one or multiple events can store a freeze frame. Nevertheless, only one DTC and its corresponding freeze frame is available in Mode $02.

For all OBD events, only one global defined OBD freeze frame exists.

Since there are project specific requirements regarding the freeze frame behavior, the Dem supports multiple configuration options in terms of storage trigger, visibility in Mode $02, update behavior and number of internal stored freeze frames.

The storage of the legislative freeze frame is independent of event storage.

2.2.5.1.2 Reporting in Mode $02

Not only the storage, but also the reporting of freeze frames in Mode $02 can be configured. The following behaviors are possible:

- Freeze frame is stored with pending and is immediately reported in Mode $02
- Freeze frame is stored with pending, but Mode $02 reporting is suppressed until the confirmed status of the DTC is externally visible (see chapter 2.2.4.2). Using this configuration and depending on the number of configured OBD related events and freeze frame slots, it is possible that all freeze frame slots are occupied. In consequence, there might be no freeze frame visible in mode $02 although a confirmed DTC is visible in mode $03 in some scenarios. For example:
  - All freeze frame slots are occupied by pending freeze frames when another OBD related event confirms. The pending freeze frame cannot be displaced by the confirmed freeze frame since the reasons described in chapter 2.2.5.1.5 are not fulfilled.
  - All freeze frame slots are occupied by one confirmed and other pending freeze frames when another OBD related event becomes pending. The confirmed event is displaced by the pending event due to the reasons described in chapter 2.2.5.1.5.
- Freeze frame is stored when the event's trip condition is fulfilled (Refer to chapter 0). However, Mode $02 reporting is suppressed until the confirmed status of the DTC is externally visible (see chapter 2.2.4.2). In case the confirmed status of a different DTC gets externally visible before the confirmed status of the DTC holding the freeze frame, the already stored but not yet visible freeze frame will be displaced with the new, immediately visible freeze frame. In configurations with the feature *major monitors* enabled, prioritization of OBD freeze frames with respect to the event's configured readiness group is considered:
  - If the stored but not yet visible OBD freeze frame is related to a misfire or fuel system monitor, it is not displaced by a DTC whose confirmed status is immediately externally visible. Instead, the misfire or fuel system related DTC becomes visible without the need for a driving cycle qualification and is reported via mode $02 as well as mode $03.[1]
  - A stored freeze frame which is *not* related to a misfire or fuel system monitor is displaced by the freeze frame of a confirmed misfire or fuel system related monitor (cf. section 2.2.5.1.5). If the displaced freeze frame was already visible, the misfire or fuel system related freeze frame will immediately be reported via mode $02 and mode $03 without the need of driving cycle qualification.

[1] The DTC status change will be notified as if the driving cycle was qualified.

Caution
Correct reporting of the legislative freeze frame in Mode $02 can only be guaranteed if the priority of all OBD relevant events is configured as described in Table 2-46 Event configuration.

2.2.5.1.3 Reporting in Service $19 04

The legislative freeze frame becomes available in diagnostic service $19 04 as soon as it is stored in freeze frame memory.
The visibility in $19 04 is independent of the configured Mode $02 visibility. Therefore, the legislative freeze frame can be reported in $19 04 although it is not visible yet in Mode $02.

2.2.5.1.4 Removal

The legislative freeze frame is cleared together with the associated fault. There are the following reasons why the legislative freeze frame is erased:

- The fault is cleared by a diagnostic service request (e.g. Mode $04, see 2.2.10.1.4).
- The fault is displaced (see chapter 2.2.4.3).
- The associated fault has aged.
- The legislative freeze frame is displaced by the legislative freeze frame of another fault (see 2.2.5.1.5).

2.2.5.1.5 Displacement

In general, a stored legislative freeze frame is not displaced by another event's freeze frame. Freeze frames related to misfire and fuel system monitors are an exception: To comply with [8], the Dem will displace a currently stored legislative freeze frame related to a fault other than misfire or fuel system monitoring by a freeze frame related to misfire or fuel system monitoring. If multiple OBD freeze frame can be considered for displacement, the selection is governed by the following set of rules, in the order of mention:

- invisible before visible
- newer before older

In this case the freeze frame related to the misfire or fuel system fault is reported in Mode $02.

Note

As the OBD freeze frame is cleared at event displacement (cf. chapter 2.2.4.3), it has to be ensured that the event holding the OBD freeze frame according to [8], is not displaced by a fault other than misfire or fuel system or a newer fault related to misfire or fuel system monitoring. This can be achieved by assigning the same highest priority to all misfire and fuel system monitors and a lower priority to all other monitors.

Note

In case the Dem is configured to capture the freeze frame data when the event’s trip condition is fulfilled but visibility of the OBD freeze frame in mode $02 is delayed until driving cycle qualification, an invisible freeze frame can be displaced by an immediately visible freeze frame (i.e. corresponding to a DTC which is not configured to suppress the external visibility of the confirmed status bit until DCY qualification, see section 2.2.5.1.2).

2.2.5.2 OBDonUDS

This chapter is only relevant for **OBDonUDS**.

2.2.5.2.1 Storage

DEM stores OBDonUDS Freeze Frames, which can be accessed by diagnostic service $19-04. Each OBD relevant event tries to store two freeze frames which have identifiers 0x00 and 0xF0. Multiple such freeze frame pairs can be stored in different OBD Freeze Frame buffer slots. The number of slots is configurable.

OBDonUDS Freeze Frames with IDs 0x00 and 0xF0 are stored on the first transition of the TFTOC bit (Test Failed This Operation Cycle) from zero to one. After freeze frame 0xF0 has been stored, it gets updated each time, the TFTOC bit has another transition from zero to one.

OBDonUDS Freeze Frames are stored independently from event storage if event status bits PDTC and CDTC are set independently from event storage (see [1]).

2.2.5.2.2 Reporting, Removal and Displacement

The legislative freeze frame becomes available in diagnostic service $19 04 as soon as it is stored in freeze frame memory.
OBDonUDS Freeze Frames can’t be accessed by OBD Mode $02 nor UDS service $19 05.

The legislative freeze frame is cleared if event status bits PDTC and CDTC of the associated fault are not set. There are the following reasons why PDTC and/or CDTC are reset:

- The fault is cleared by a diagnostic service request (e.g., Service $14, see 2.2.10.2.1).
- The fault is displaced (see chapter 2.2.4.3) and this results in resetting PDTC[1] and/or CDTC[2] event status bits.
- The associated fault has aged.
- The associated fault is tested passed and not failed for a driving cycle.

2.2.5.3 WWH-OBD

This chapter is only relevant for **WWH-OBD**.

[1] This may be configured with: /MICROSAR/Dem/DemGeneral/DemPendingDtcProcessing=STOREDONLY

[2] This may be configured with: /MICROSAR/Dem/DemGeneral/DemResetConfirmedBitOnOverflow=TRUE

2.2.5.3.1 Storage

Each OBD related DTC that becomes a pending DTC stores an OBD freeze frame which can be accessed by diagnostic service $19. If the pending trigger reoccurs before the DTC is in confirmed state, the OBD freeze frame is updated.

Optionally the Dem can be configured to update the freeze frame content as soon as the DTC becomes a confirmed DTC.

The legislative freeze frame is only stored if the event could be successfully stored in memory.

2.2.5.3.2 Reporting, Removal and Displacement

The legislative freeze frame becomes available in diagnostic service $19 04 as soon as the corresponding fault is stored.
The legislative freeze frame is cleared together with the associated fault. There are the following reasons why the legislative freeze frame is erased:

- The fault is cleared by a diagnostic service request (e.g., Service $14, see 2.2.10.2.1).
- The fault is displaced (see chapter 2.2.4.3).
- The associated fault has aged.

2.2.6 Permanent DTCs

Dem supports permanent DTCs for all OBD legislations. As the behavior differs significantly there is a subchapter for OBDII and OBDonUDS as well as for WWH-OBD.

2.2.6.1 OBD II and OBDonUDS

OBDII
OBD on UDS

This chapter is relevant for **OBD II** and **OBDonUDS**.

2.2.6.1.1 Definition

An OBD DTC is designated as permanent if the related OBD event is confirmed and requests the MIL indicator. The event remains permanent while it commands the MIL indicator.

Permanent DTCs are reported separately from all other DTCs using the OBD specific service Mode $0A (see 2.2.10.1.7), UDS Service $19 15 (see 2.2.10.2.5) or the OBDonUDS Service $19 55 (see 2.2.10.2.7).

2.2.6.1.2 Storage

Internally, the permanent DTC is stored immediately as soon as the MIL indicator is commanded, but it is typically only reported by Mode $0A (see 2.2.10.1.7), UDS service $19 15 (see 2.2.10.2.5) or OBDonUDS Service $19 55 (see 2.2.10.2.7), after the driving cycle has been restarted. An event that is configured to confirm with the first detection of the fault and whose external visibility is not delayed until the DCY is qualified will be reported immediately. A further configuration option allows that all stored permanent DTCs whose WIR bit is externally visible will be reported.

If the permanent memory is full and a new failure occurs that may command the MIL indicator, the stored entries will not be replaced.

The storage of such a failure is delayed until one of the stored events is healed and an entry becomes free. This behavior is shown in Figure 2-10, where the permanent DTC of Event1 occupies the last free entry in permanent memory. The storage of Event2 is then delayed until Event1 is healed.

If an entry in permanent memory is freed, and several events are confirmed and commanding the MIL at the same time, the 'oldest', confirmed event is chosen for storage.

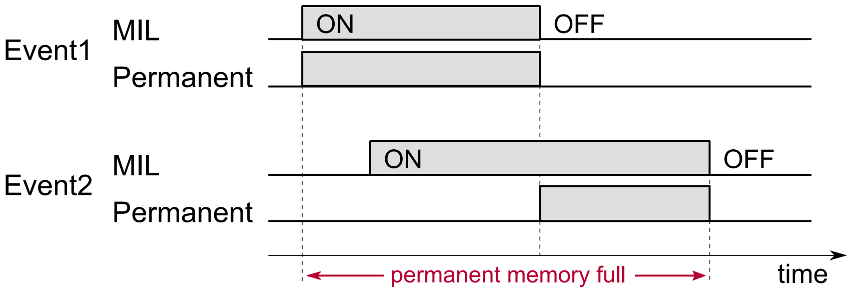

Figure 2-10 Storage of PDTC storage if permanent memory is full

Note
If the MIL indicator is not attached to an OBD related event aka black MIL DTC, the Dem does not store a permanent DTC.

2.2.6.1.3 PFC Cycle

The permanent fault code – cycle is a flag provided by the Master ECU that states if the vehicle has been operated under general conditions.

It is used to determine after a clear request from an external tester, if a permanent DTC can be erased in the current driving cycle. For this it is sufficient that the PFC cycle have been seen once on the bus during the current driving cycle.

The conditions for the PFC cycle are similar to those that lead to an increment of the general denominator for IUMPR, but do not contain requirements for temperature and altitude.

2.2.6.1.4 Erasing of Permanent DTCs

Generally, as long as a DTC has permanent status it cannot be erased by a tester or replaced by another confirmed fault.

Only when the event is healed and the MIL indicator is no longer commanded by this event, the permanent DTC is erased.

Because a clear request from an external tester (Mode $04 / Service $14) resets the MIL indicator, it would also erase all permanent DTCs. To prevent that permanent DTCs are removed although the underlying failure condition is still active, the clear request is only executed to the permanent state after the following conditions are fulfilled:

- For one driving cycle, the event is tested with status passed only (test result is never failed)
- If then the PFC cycle is started and the event is still not reported as failed, the permanent DTC is finally erased at the end of the driving cycle.
  If configured the Dem does not use this condition for DTCs that have a ratio attached.

The two conditions can, but do not have to occur in the same driving cycle. Single test results with status failed only delay the erasing of the permanent DTC. However, if the event is tested failed until the MIL indicator is requested again, the clear request is discarded.

Example
Erasing of permanent DTC if failure condition has disappeared

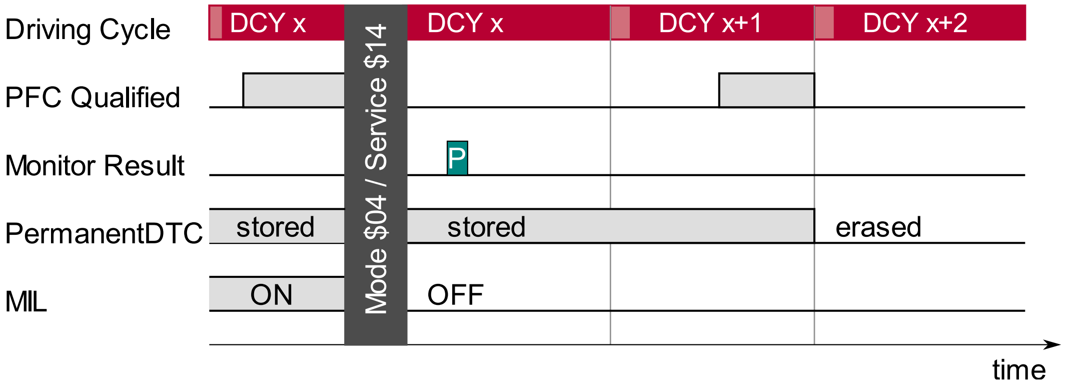

Figure 2-11 Erasing of permanent DTC – failure condition disappeared

Figure 2-11 shows the best-case behavior: the DTC is cleared and the failure no longer exists. Then the event is tested as passed and the conditions to clear the DTC are fulfilled (PFC cycle is provided by Master ECU). Therefore, the permanent DTC is erased at the end of the driving cycle, if both conditions have been fulfilled.

Example
Erasing of permanent DTC if failure toggles

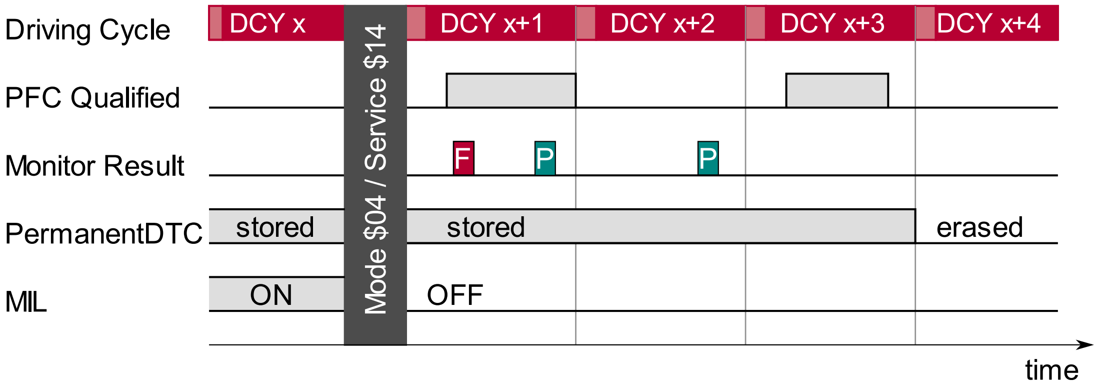

Figure 2-12 Erasing of permanent DTC – failure condition toggles

Figure 2-12 shows the behavior when the event is tested passed but also failed after the clear request. Although the conditions are given, the permanent DTC is then not erased. Only after the event is tested passed (DCY x+2) and a PFC cycle is detected (DCY x+3) without any failed monitor result, the permanent DTC is removed from memory.

Example
Erasing of permanent DTC if failure condition is still present

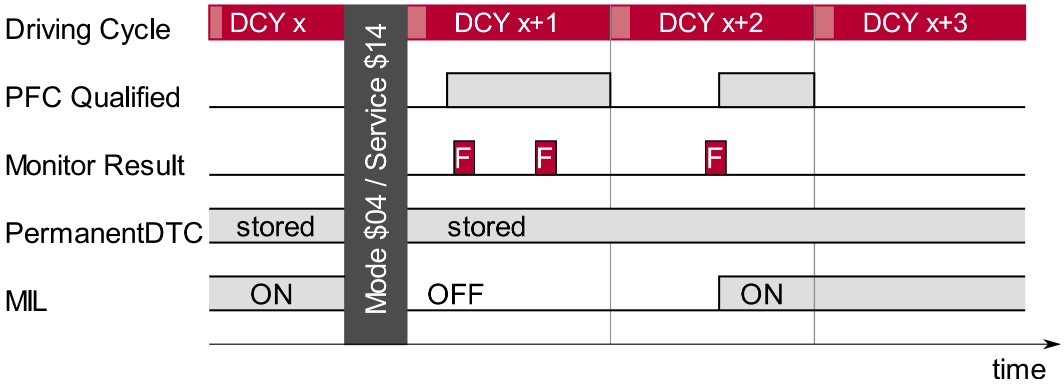

Figure 2-13 Erasing of permanent DTC – failure condition still present

In the last example, the underlying failure that caused the MIL indicator request is still present after the clear processing. When the event is tested often enough to re-activate the MIL indicator (in Figure 2-13, a two-trip configuration is assumed), the clear request is not executed and the permanent DTC will remain in memory.

To remove in this last example the permanent DTC from memory, one of the following behaviors can be configured in the Dem.

- **Option mandatory PFC cycle after clear is disabled:**
  When the event re-activates the MIL indicator after Mode $04, the PFC cycle is no longer needed to remove the permanent DTC. It is erased by regular healing (in case of OBD II: no failed results for 3 driving cycles)
- **Option mandatory PFC cycle after clear is enabled:**
  The PFC cycle is always mandatory after Mode $04 until the permanent DTC is finally erased. Even if the MIL indicator is re-activated by the event, the conditions to remove the permanent DTC remain the same as shown in Figure 2-11 and Figure 2-12.

2.2.6.1.5 Suppression of Permanent DTC Storage

During production or initial power-up, OBD relevant events may report an error, request the MIL indicator and store a permanent fault, although it is not relevant for regular operation.

As it is difficult to remove such unjustified permanent faults, the storage can be suppressed for initial power up. It is activated when the standard OBD odometer value (see 2.3.2) reaches a specific distance. This distance is calculated by adding the configured activation distance to the first valid odometer value provided by the application.

Additionally, the Dem provides the API `Dem_EnablePermanentStorage()` to activate permanent storage before the distance is reached (e.g. due to a diagnostic service request).

Once activated, permanent storage cannot be disabled. At the time of activation, all events that have the WIR status bit set will be stored as permanent. If the WIR status bit is externally visible, the permanent DTC will also be externally visible.

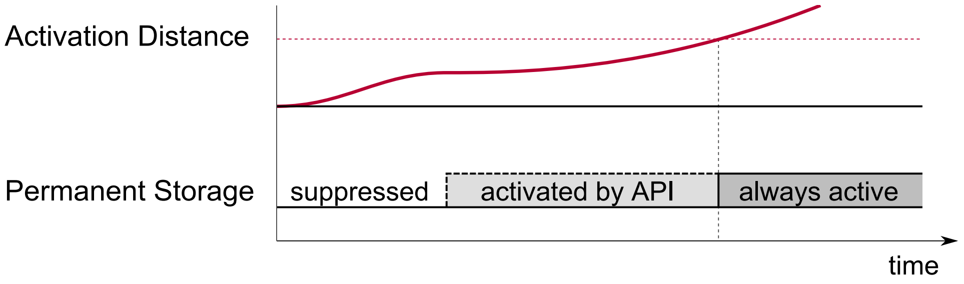

Figure 2-14 Permanent DTC Storage Suppression

Note
This feature is optional.
If the configuration parameter “Permanent DTC Activation Distance” is not provided, the API will not be available and permanent storage will never be suppressed.

2.2.6.2 WWH-OBD

This chapter is only relevant for **WWH-OBD**.

2.2.6.2.1 Definition and Storage

An OBD relevant DTC is designated as permanent if the related OBD event has a Class A or B1 DTC (see chapter 2.2.3.5 DTC Class), is confirmed and active (TestFailed Bit set). Additionally, for B1 DTCs the Single B1 Counter must exceed 200h (i.e., activation mode 4 / Continuous MI is triggered). If the DCY is not qualified while the event fulfills permanent memory storage preconditions, the storage is postponed until DCY is qualified. The event remains permanent as long as it is active.

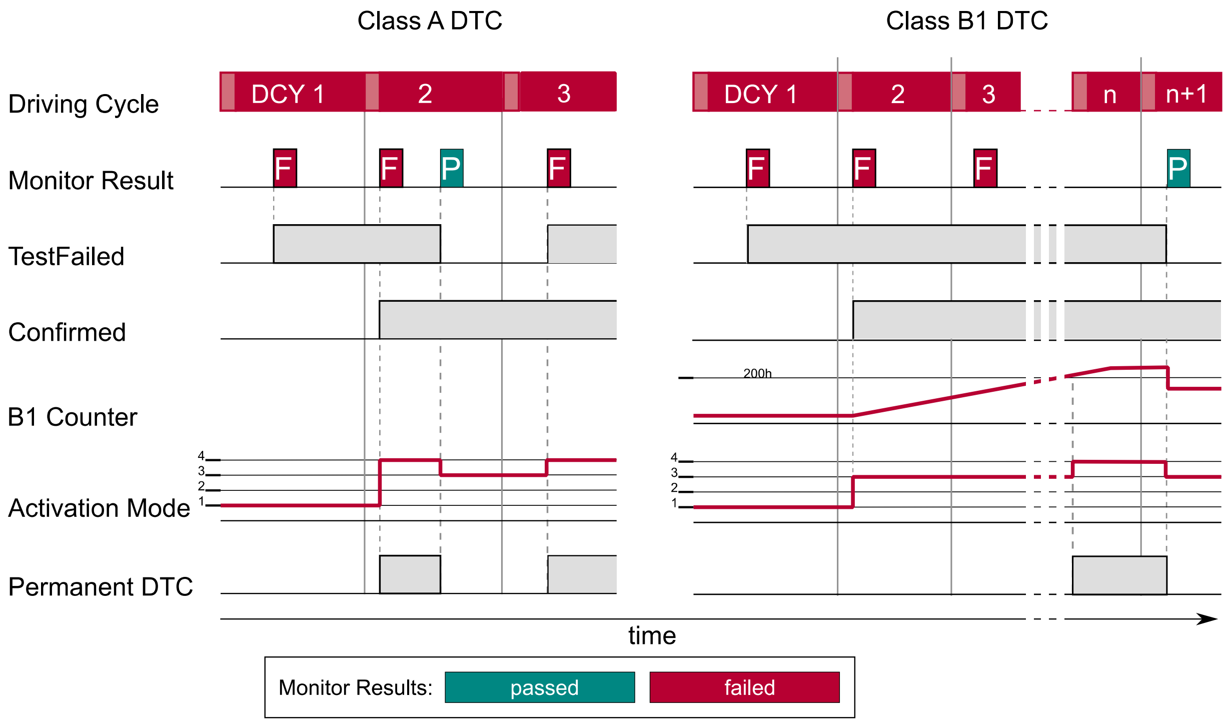

Figure 2-15 Permanent storage for a Class A DTC (left side) and a Class B1 DTC.

If a new failure confirms but could not be stored since permanent memory is full, the stored entries will not be replaced.

The storage of such a failure is delayed until one of the stored events is no longer active and an entry becomes free. This behavior is shown in Figure 2-16, where the permanent DTC of Event1 occupies the last free entry in permanent memory. The storage of Event2 is then delayed until Event1 is no longer active.

If an entry in permanent memory is freed, and several events are confirmed and active at the same time, the 'oldest', confirmed event is chosen for storage.

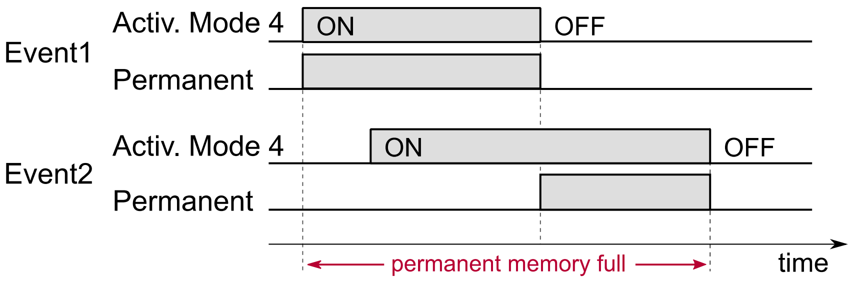

Figure 2-16 Storage of PDTC storage if permanent memory is full

Permanent DTCs are reported separately from all other DTCs using the UDS services $19 15 or $19 55.

2.2.6.2.2 Erasing Permanent DTCs

Generally, as long as a DTC has permanent status it cannot be erased by a tester or replaced by another confirmed fault.

After a clear request from an external tester (Service $14), the request is only executed to the permanent state after the event is tested with status passed only (test result is never failed) for one driving cycle.

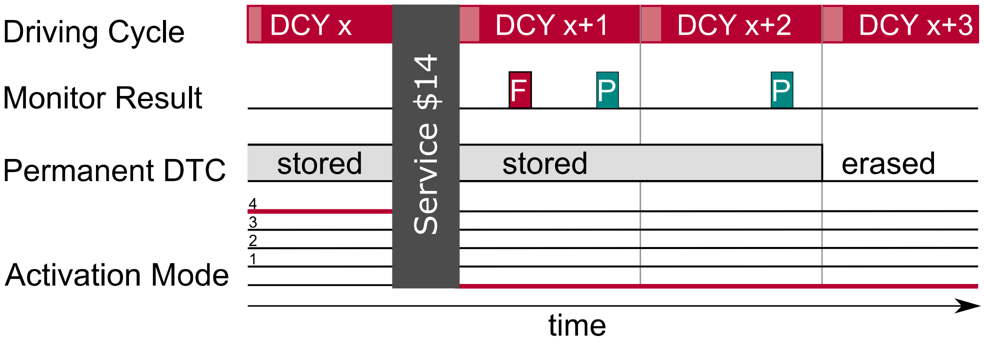

Figure 2-17 Erasing of permanent DTC – failure condition toggles

However, if the event is tested failed until it triggers activation mode 4 again, the clear request is discarded.

2.2.7 Activation Mode

WWH-OBD This chapter is only relevant for **WWH-OBD**.

The activation mode represents the severity of the detected malfunctions. Dependent on the current activation mode the application has to trigger the malfunction indicator according legislative requirements. The Dem provides the indicator status of the malfunction indicator lamp via the API `Dem_GetIndicatorStatus()` (refer to [1]) which can be mapped to the activation mode (see Table 2-11).

Activation Mode	Condition for Entering	Indicator Status
4	Class A malfunction detected or B1 counter >= 200h	DEM_INDICATOR_CONTINUOUS
3	Class B1 or B2 malfunction detected and B1 counter < 200h	DEM_INDICATOR_SHORT
2	Class C malfunction detected	DEM_INDICATOR_ON_DEMAND
1	No OBD related malfunction detected	DEM_INDICATOR_OFF

Table 2-11 Activation Mode Entering Conditions

Due to multiple faults with different DTC classes can be active at the same time the activation modes will overlap each other, in this case the activation mode with highest priority is active. Activation mode 1 is defined with the lowest priority and activation mode 4 with the highest priority. If no condition for activation mode 4 is fulfilled anymore, the activation mode is degraded to activation mode 3 instantly during the running driving cycle. The degradation of activation modes 3 or 2 is delayed until 3 driving cycles have passed without the conditions being fulfilled again (cf. examples in 2.2.7.2).

Caution
For the malfunction indicator unlike described in [1] the IndicatorStatus parameter value returned by API `Dem_GetIndicatorStatus()` doesn't depend on the configured indicator behavior values. Instead only the values listed in column 'Indicator Status' in Table 2-11 are returned for the IndicatorStatus parameter value depending on the entered activation mode.

2.2.7.1 Single B1 Counter

The global B1 counter traces the engine operation hours during which a Class B1 DTC is in state

- ConfirmedDTC == True **and**
- TestFailed == True

The counter is incremented once a Class B1 DTC fulfills the conditions mentioned above. If no Class B1 DTC fulfills these conditions, the counter will be latched at its current value or set to 190 hours when equal or greater than 200 hours. If all Class B1 DTCs are reported only passed for 3 driving cycles, the B1 counter will be reset to zero. If a Class B1 DTC again fulfills the conditions before the B1 counter is reset, then the counter continues incrementing.

Having a B1 counter of 200 hours and more will escalate the activation mode to mode 4.

Refer to Figure 2-19 for an example of the B1 counter.

Note
To reset the B1 counter after clearing all DTCs, it is also necessary for all Class B1 DTCs to be reported only passed for 3 driving cycles.

2.2.7.2 Activation Mode Examples

Example
Activation Mode for a **Class A** DTC.

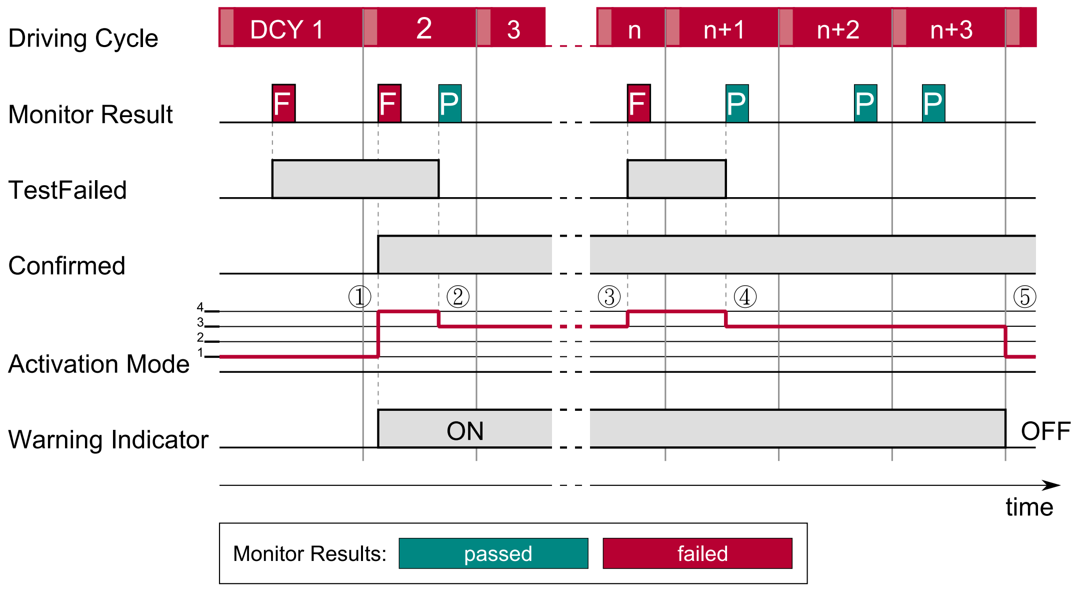

Figure 2-18 Activation Mode example for Class A DTC

① As soon as the class A DTC becomes confirmed, activation mode 4 is entered.

② After the confirmed DTC was tested as passed, the activation mode is degraded to activation mode 3.

③ Again the DTC is tested as failed, so the activation mode 4 is reentered.

④ Once more the DTC is tested as passed which leads to activation mode 3 degradation.

⑤ After 3 failure-free cycles the activation mode 1 is entered.

Example
Activation Mode for a **Class B1** DTC.

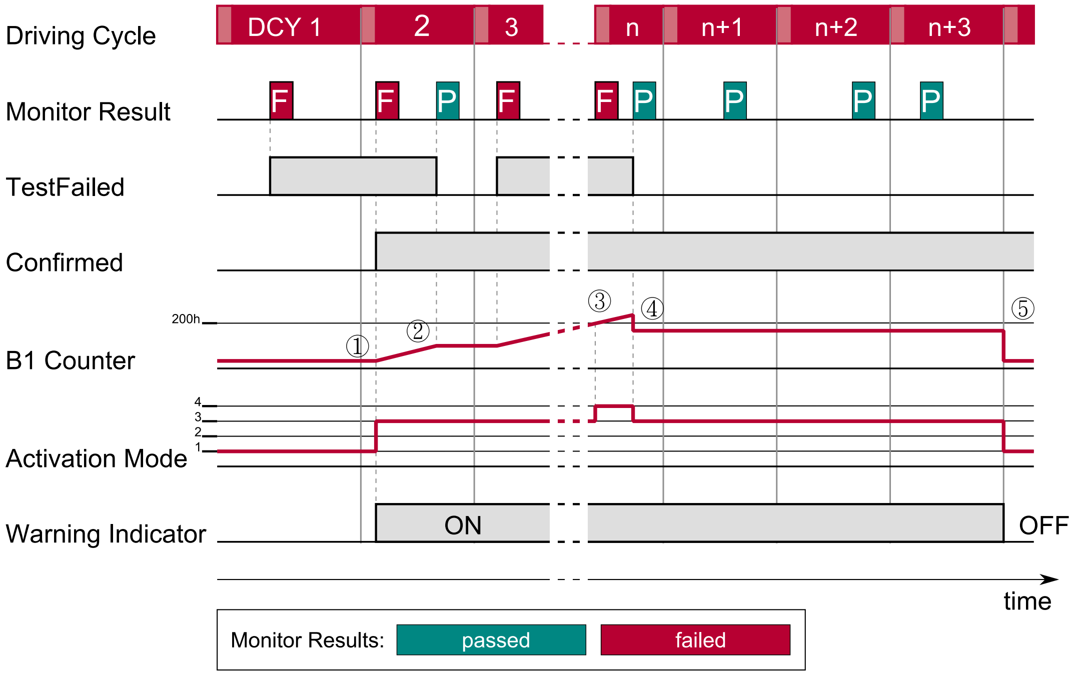

Figure 2-19 Activation Mode example for Class B1 DTC

① As soon as the class B1 DTC becomes confirmed, activation mode 3 is entered and the B1 counter is started.

② After the DTC was tested as passed the B1 counter is latched until the DTC is again tested as failed.

③ If the B1 counter reaches the 200h threshold the activation mode 4 is entered.

④ For the next passed result, the activation mode is degraded to activation mode 3 and the B1 counter is set to 190h.

⑤ After 3 tested and failure-free cycles the activation mode 1 is entered.

Example
Activation Mode for a **Class B2** DTC.

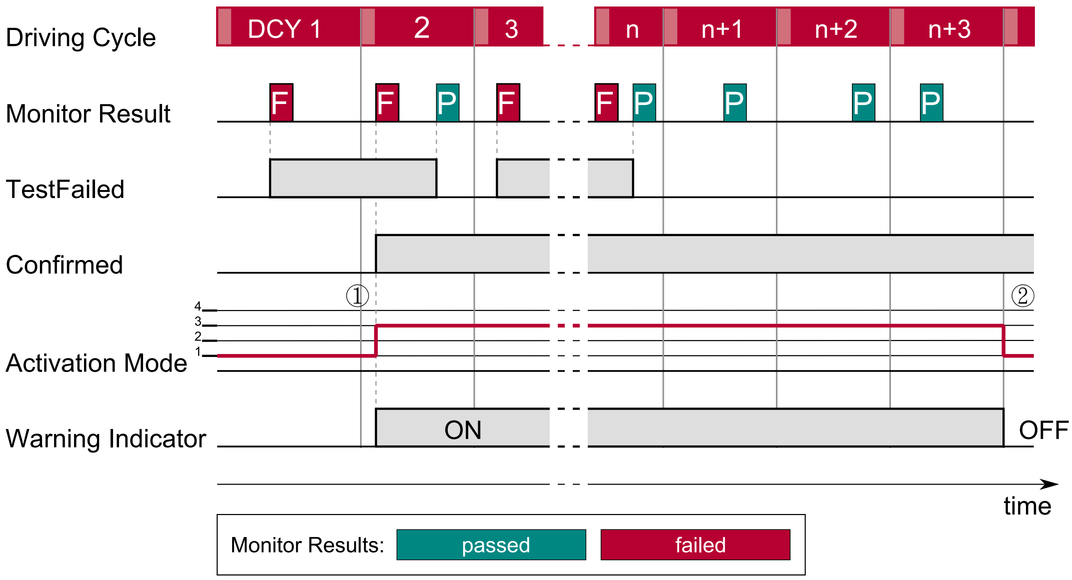

Figure 2-20 Activation Mode example for Class B2 DTC

① A class B2 DTC becomes confirmed so activation mode 3 is will be entered.

② Activation mode 1 is entered as soon as 3 failure-free cycles have been passed.

Example
Activation Mode for a **Class C** DTC.

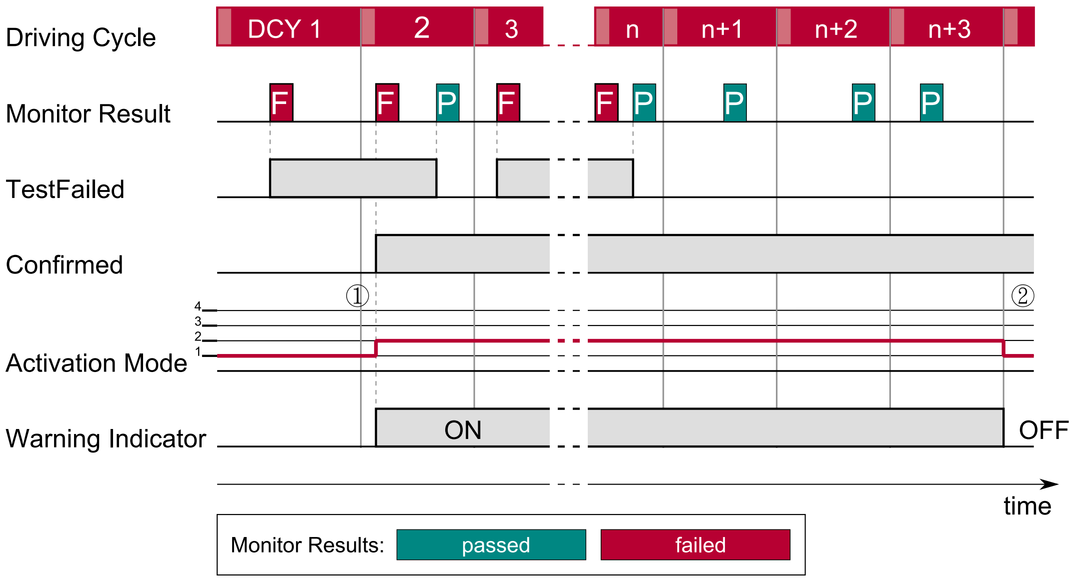

Figure 2-21 Activation Mode example for Class C DTC

① A class C DTC becomes confirmed so activation mode 2 is will be entered.

② Activation mode 1 is entered as soon as 3 failure-free cycles have been passed.

2.2.8 Continuous-MI Counter

This chapter is only relevant for **WWH-OBD**.

The continuous malfunction indicator counter records the number of engine runtime hours during which the continuous-MI (activation mode 4) was active.

The counter is

- incremented for every hour the activation mode 4 is active
- latched if activation mode 4 is degraded
- incremented if activation mode 4 is re-entered within 3 driving cycles
- reset to zero bevor it is incremented if activation mode 4 is re-entered after 3 driving cycles have passed without activation mode 4 was entered
- reset to zero if activation mode 4 was not re-entered for 40 warmup cycles or 200 engine operating hours

Figure 2-22 depicts the behavior described above.

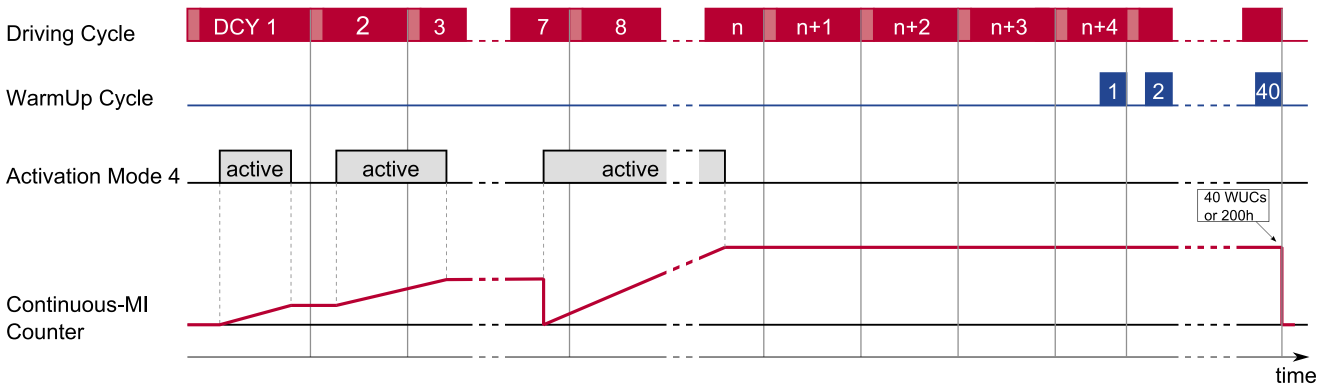

Figure 2-22 Continuous-MI counter

Note

Due to a DTC is first healed before it starts to aging the continuous-MI counter is linked to that behavior. Therefore 3 driving cycles plus 40 warmup cycles (or 200h) without activation mode 4 are necessary to reset the continuous-MI counter.

2.2.9 PIDs calculated by Dem

OBD PIDs are standardized codes to identify emission related data in a vehicle. Not all PIDs are provided by any ECU, and only a few PIDs are calculated by the Dem. For a complete description of all PIDs, please refer to [6].

2.2.9.1 PID $01 and PID $F501: Monitor status since DTCs cleared

The monitor status parameter contains the status of OBD relevant monitoring functions. If the feature *major monitors* is disabled, the Vector Dem only supports comprehensive components and most status bits are not applicable and statically set. Otherwise, all bits are calculated at runtime.

An emission related monitoring function will be reported as supported if at least one event is available and assigned to the related readiness group.

A supported monitoring function is completed or not applicable if

- no event is assigned **OR**
  no event is available in the related readiness group.
- at least one available event that is assigned to the related readiness group becomes confirmed.
- all available events that are assigned to the related readiness group are tested with status passed since DTCs cleared and none of these events is in state pending.
- event monitoring is disabled by Dem_DisableEventMonitoringInCurrentDcy().

Note
Please note that for an event that is used with event combination type 1 and type 3, its status ‘tested passed since DTC cleared’ is fulfilled if all subevents of the combined event are tested passed in the same operation cycle.

Caution
The Dem does not consider additional conditions for completeness (e.g. whether 4,000 engine revolutions have occurred).

A completed monitoring function will only be reset to not completed after DTCs are cleared, an unavailable event assigned to related readiness group is set to available or event monitoring of the readiness group is disabled through Dem_DisableEventMonitoringInCurrentDcy() and event reporting occurs or the driving cycle is restarted.

Note
Only OBD relevant events can be assigned to a readiness group.

Note
In PID $01, the readiness group DEM_OBD_RDY_MISF is always reported as completed for spark-ignition vehicles. For compression engines, setting DemGeneral/DemGeneralOBD/DemOBDCompressionMisfireAlwaysComplete to true results in readiness group DEM_OBD_RDY_MISF being always reported as completed in PID $01.

For PID $F501 the readiness group's completeness is always calculated during runtime.

Note
In PID $01, the readiness group DEM_OBD_RDY_CMPRCMPT is always reported as completed according to [6].

In PID $F501, setting DemGeneral/DemGeneralOBD/DemOBDPIDF501ComprehensiveComponentsAlwaysComplete to true results in readiness group DEM_OBD_RDY_CMPRCMPT being always reported as completed. Otherwise, the readiness group's completeness is calculated during runtime.

Note
The readiness groups DEM_OBD_RDY_FLSYS and DEM_OBD_RDY_FLSYS_NONCONT are assigned to the same bits in PID $01 and PID $F501.

For the 'supported bit' calculation both readiness groups are considered.

For the 'completeness bit' calculation only readiness group DEM_OBD_RDY_FLSYS_NONCONT is considered.

Also, setting DemGeneral/DemGeneralOBD/DemOBDFuelSystemAlwaysComplete to true results in readiness group DEM_OBD_RDY_FLSYS_NONCONT being always reported as completed.

Caution
Using clear single DTC (e.g. during production process) will reset the readiness state independent of the state of all other DTCs assigned to the same readiness group.

Using the clear event allowed feature will also reset the readiness state although one or more events could not be cleared.

2.2.9.1.1 Effects of Power Take-Off

For events affected by the Dem PTO handling, the completeness of a monitoring function is additionally recalculated based on the PTO activation timer value (see chapter 2.2.3.8.2):

- When an event's PTO activation timer reaches 750 minutes, the event is considered as not tested for the completeness calculation. At this time point, the completeness status is recalculated for a completed monitoring function. If no other available event in the readiness group has been confirmed since DTCs cleared, the monitoring function will be reset to not completed. Otherwise, the completeness status won't be changed.
- When an event's PTO activation timer is reset from 750 minutes to 0, the event can be considered tested again based on the current event status. The monitoring function will be set to completed if
  - the event is currently confirmed (CDTC bit is set), or
  - all available events that are assigned to the related readiness group are tested with status passed since DTCs cleared and none of these events is in state pending.

Caution

Events that have been confirmed but then aged or displaced, still cause the monitoring function to be complete, even if the PTO activation timer of any other event in this readiness group reaches 750 minutes.

Only if an event that has been confirmed but then aged or displaced, is affected by PTO and its PTO activation timer has reached 750 minutes, it will not cause the monitoring function to be complete.

Note

If any event's PTO activation timer has reached 750 minutes, even if all available events that are assigned to the related group are tested with status passed since DTCs cleared and none of these events is in state pending, the monitoring function is not completed.

2.2.9.1.2 PID $01 Structure

Byte	Bit	Major Monitors	Value	Description
0	0…6	don't care	Calculated	Number of OBD DTCs that are currently stored and that can be read by Mode $03. For WWH-OBD always set to 0.
	7	don't care	Calculated	Current local status of the MIL indicator. It is configurable (see configuration parameter "Dem Report Blinking Mil as Active" in chapter 2.4.1.9), if the Dem reports a blinking (and not continuous) MIL as activated. For WWH-OBD always set to 0.
1	0…3	Disabled	0x4	Support for different emission related monitoring functions. If feature *major monitors* is disabled only comprehensive component monitoring is enabled. (1 = supported, 0 = not supported)
		Enabled	Calculated	
	4…7	Disabled	0x0	Readiness status of these monitors (0 = monitor completed or N/A, 1 = monitor not completed)
		Enabled	Calculated	
2	0…7	Disabled	0x00	Support of additional emission related monitoring functions. If feature *major monitors* is disabled no additional monitors are supported. (1 = supported, 0 = not supported)
		Enabled	Calculated	
3	0…7	Disabled	0x00	Readiness status of the additional monitors (0 = monitor completed or N/A, 1 = monitor not completed)
		Enabled	Calculated	

Table 2-12 PID $01 implementation

- Related API function: Dem_DcmReadDataOfPID01()
- Detailed bitwise supportness and completion information can be found in the function declaration of Dem_DcmReadDataOfPID01() within Dem_Dcm.h.

2.2.9.1.3 PID $F501 Structure

This chapter is only relevant for **OBDonUDS**.

Byte	Bit	DemOBDPIDF501ComprehensiveComponentsAlwaysComplete	Value	Description
0	0…6	don't care	Calculated	Number of OBD DTCs that are currently stored and that are confirmed (CDTC bit is set).
	7	don't care	Calculated	Current local status of the MIL indicator. It is configurable (see configuration parameter "Dem Report Blinking Mil as Active" in chapter 2.4.1.9), if the Dem reports a blinking (and not continuous) MIL as activated.
1	0,1	don't care	0	Support for different emission related monitoring functions. If feature *major monitors* is disabled only comprehensive component monitoring is enabled. (1 = supported, 0 = not supported)
	2	Enabled	1	
		Disabled	Calculated	
	3	don't care	0	
	4,5	don't care	0	Readiness status of these monitors (0 = monitor completed or N/A, 1 = monitor not completed)
	6	Enabled	0	
		Disabled	Calculated	
	7	don't care	0	
2…5	0…3	don't care	0x00	Support of additional emission related monitoring functions. If feature *major monitors* is disabled no additional monitors are supported. (1 = supported, 0 = not supported)
	4…7	don't care	0x00	Readiness status of the additional monitors (0 = monitor completed or N/A, 1 = monitor not completed)

Table 2-13 PID $F501 implementation with disabled Major Monitors

Byte	Bit	DemOBDPIDF501ComprehensiveComponentsAlwaysComplete	Value	Description
0	0…6	don't care	Calculated	Number of OBD DTCs that are currently stored and that are confirmed (CDTC bit is set).
	7	don't care	Calculated	Current local status of the MIL indicator. It is configurable (see configuration parameter "Dem Report Blinking Mil as Active" in chapter 2.4.1.9), if the Dem reports a blinking (and not continuous) MIL as activated.

Byte	Bit	DemOBDPIDF501ComprehensiveComponentsAlwaysComplete	Value	Description
1	0…3	don't care	Calculated	Support for different emission related monitoring functions. (1 = supported, 0 = not supported)
	4,5	don't care	Calculated	Readiness status of these monitors (0 = monitor completed or N/A, 1 = monitor not completed)
	6	Enabled	0	
		Disabled	Calculated	
	7	don't care	Calculated	
2…5	0…3	don't care	Calculated	Support of additional emission related monitoring functions. (1 = supported, 0 = not supported)
	4…7	don't care	Calculated	Readiness status of the additional monitors (0 = monitor completed or N/A, 1 = monitor not completed)

Table 2-14 PID $F501 implementation with enabled Major Monitors

- Related API function: Dem_DcmReadDataOfPIDF501()
- Detailed bitwise supportness and completion information can be found in the function declaration of Dem_DcmReadDataOfPIDF501() within Dem_Dcm.h.

2.2.9.2 PID $02: DTC that caused required freeze frame data storage

OBDII

This chapter is only relevant for **OBD II**.

This PID contains the DTC number that triggered freeze frame storage.

- Related API function: Dem_DcmGetDTCOfOBDFreezeFrame()

2.2.9.3 PID $1C: OBD requirements to which vehicle is designed

The value of this PID is statically provided by configuration. It specifies the legislative regulation against which the ECU is certified (e.g. OBD I, OBD II, EOBD…).

- Related API function: Dem_DcmReadDataOfPID1C()

2.2.9.4 PID $21: Distance travelled while MIL is activated

PID $21 is a 2 byte value with a resolution of 1 km / Bit. It is accumulated while the MIL indicator is active.

If the counter reaches its maximum of 65535 km, this value is kept until the next reset, no wrap-around occurs.

The current counter value is frozen if the MIL indicator is deactivated (Figure 2-23).

The value of PID $21 is reset if the MIL is activated again, no DTC requests the MIL indicator for 40 warm-up cycles or if the memory is erased by Mode $04.

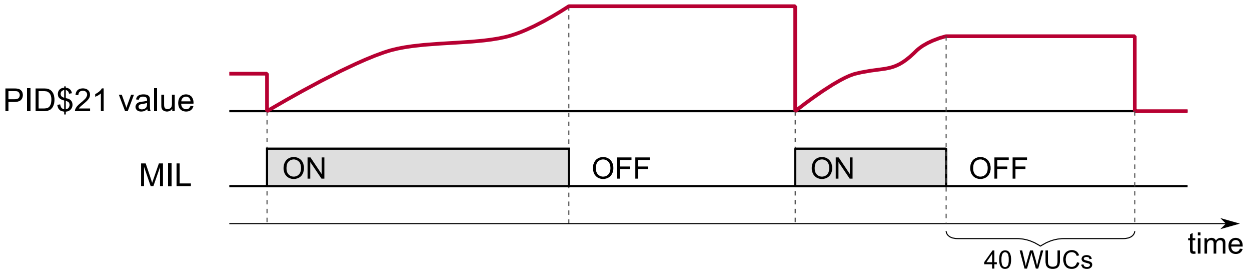

Figure 2-23 Calculation of distance with activated MIL indicator

- Related API function: Dem_DcmReadDataOfPID21()

Caution
As stated in see [12], the PID$21 value shall be frozen when source of vehicle speed is not available. DEM computes the PID$21 value based on the odometer value provided by the application. Thus, the application should freeze the reported odometer value when the source of vehicle speed is not available.

Note
It is configurable (see configuration parameter “Dem Report Blinking Mil As Active” in chapter 2.4.1.9), if the Dem considers a blinking (and not continuous) MIL as active for the PID $21 calculation.

Caution
It is not supported to have OBD DTCs without a MIL and a priority lower than other OBD DTCs with MIL support and the primary memory is not big enough to store all OBD DTCs at once.

2.2.9.5 PID $30: Number of warm-ups since DTCs cleared

This PID contains the number of warm-up cycles occurred since the last Mode $04 or Service 0x14 request. If the maximum value of 255 is reached, no wrap-around to zero occurs. The maximum value is kept until the next reset.

If the Dem is configured to delay the warm-up cycle (see chapter 2.2.2.4.1) until the end of a qualified driving cycle, PID $30 calculation will not be affected. PID $30 will be increased as soon as the application reports that the warm-up conditions are met.

- Related API function: Dem_DcmReadDataOfPID30()

2.2.9.6 PID $31: Distance since DTCs cleared

This PID is a 2 byte value that contains the travelled distance in km since the last Mode $04 or service 0x14 requests. If the maximum value of 65535 km is reached, no wrap-around to zero occurs. The maximum value is kept until the next reset.

- Related API function: Dem_DcmReadDataOfPID31()

Caution
As stated in see [12], the PID$31 value shall be frozen when source of vehicle speed is not available. DEM computes the PID$31 value based on the odometer value provided by the application. Thus, the application should freeze the reported odometer value when the source of vehicle speed is not available.

2.2.9.7 PID $41: Monitor status this driving cycle

Similar to PID $01, PID $41 is a 4 byte value that contains the current status of emission relevant monitoring functions.

If the feature *major monitors* is disabled the Vector Dem only supports comprehensive components. Depending on the configuration the value of PID $41 is statically set or calculated during runtime.

Byte	Bit	Comprehensive Component Always Complete	Value	Description
0	0…7	don't care	0x00	Reserved
1	0,1	don't care	0	Support for different emission related monitoring functions. (1 = supported, 0 = not supported) If feature *major monitors* is disabled only comprehensive component monitoring is enabled.
	2	Enabled	1	
		Disabled	Calculated	
	3	don't care	0	
	4,5	don't care	0	Readiness status of these monitors (0 = monitor completed or N/A, 1 = monitor not completed)
	6	Enabled	0	
		Disabled	Calculated	
	7	don't care	0	
2	0…7	don't care	0x00	Support of additional emission related monitoring functions (1 = supported, 0 = not supported) If feature *major monitors* is disabled no

Byte	Bit	Comprehensive Component Always Complete	Value	Description
				additional monitors are supported by Vector Dem
3	0…7	don’t care	0x00	Readiness status of the additional monitors (0 = monitor completed or N/A, 1 = monitor not completed)

Table 2-15 PID $41 implementation with disabled Major Monitors

Byte	Bit	Comprehensive Component Always Complete	Value	Description
0	0…7	don’t care	0x00	Reserved
1	0…3	don’t care	Calculated	Support for different emission related monitoring functions. (1 = supported, 0 = not supported)
	4…5	don’t care	Calculated	Readiness status of these monitors (0 = monitor completed or N/A, 1 = monitor not completed)
	6	Enabled	0	
		Disabled	Calculated	
	7	don’t care	0	
2	0…7	don’t care	Calculated	Support of additional emission related monitoring functions (1 = supported, 0 = not supported)
3	0…7	don’t care	Calculated	Readiness status of the additional monitors (0 = monitor completed or N/A, 1 = monitor not completed)

Table 2-16 PID $41 implementation with enabled Major Monitors

- Related API function: Dem_DcmReadDataOfPID41()
- Detailed bitwise supportness and completion information can be found in the function declaration of Dem_DcmReadDataOfPID41() within Dem_Dcm.h.

An emission related monitoring function is reported as not supported if

- an event that is available and assigned to the related readiness group is disabled.
- no event is assigned **OR**
  no event is available in the related readiness group.

Otherwise the monitoring function is reported as supported.

An emission related monitoring function is reported as completed or N/A if

- no event is assigned **OR**
  no event is available in the related readiness group.

- at least one available event that is assigned to the related readiness group is tested with status failed in the current driving cycle.
- all events that are available and assigned to the related readiness group are tested in the current cycle.

A supported monitoring function is set to not completed when the driving cycle is restarted.

If an event assigned to a readiness group is disabled, the monitoring function is reported as not supported during the current driving cycle. However, this will not affect the calculation of completeness.

Note
Only OBD relevant events can be assigned to a readiness group.

Note
For spark-ignition vehicles the readiness group DEM_OBD_RDY_MISF is always reported as completed.
For compression engines, setting DemGeneral/DemGeneralOBD/DemOBDCompressionMisfireAlwaysComplete to true results in readiness group DEM_OBD_RDY_MISF being always reported as completed.

Note
The readiness groups DEM_OBD_RDY_FLSYS and DEM_OBD_RDY_FLSYS_NONCONT are assigned to the same bits in PID $41.

For the ‘supported bit’ calculation both readiness groups are considered.

For the ‘completeness bit’ calculation only readiness group DEM_OBD_RDY_FLSYS_NONCONT is considered. However, disabling either one of the readiness groups DEM_OBD_RDY_FLSYS or DEM_OBD_RDY_FLSYS_NONCONT results in setting the completeness status to N/A.

Also, setting DemGeneral/DemGeneralOBD/DemOBDFuelSystemAlwaysComplete to true results in readiness group DEM_OBD_RDY_FLSYS_NONCONT being always reported as complete

2.2.9.8 PID $4D: Engine run time while MIL is activated

PID $4D is a 2 byte value with a resolution of 1 min / Bit and behaves analogous to PID $21. It is accumulated while the MIL indicator is active. If the counter reaches its maximum of 65535 minutes, this value is kept until the next reset, no wrap-around occurs.

The current counter value is frozen if the MIL indicator is deactivated.

The value of PID $4D is reset if the MIL is illuminated again, no DTC requests the MIL indicator for 40 warm-up cycles or if the memory is erased by Mode $04.

- Related API function: Dem_DcmReadDataOfPID4D()

Note
It is configurable (see configuration parameter “Dem Report Blinking Mil As Active” in chapter 2.4.1.9), if the Dem considers a blinking (and not continuous) MIL as active for the PID $4D calculation.

2.2.9.9 PID $4E: Engine run time since DTCs cleared

This PID is a 2 byte value that behaves analogous to PID $31. It contains the engine run time in minutes since the last Mode $04 or service 0x14 requests. If the maximum value of 65535 minutes is reached, no wrap-around to zero occurs. The maximum value is kept until the next reset.

- Related API function: Dem_DcmReadDataOfPID4E()

2.2.9.10 PID $91: ECU OBD System Information

WWH-OBD

This chapter is only relevant for **WWH-OBD**.

The OBD system information parameter is a 5 byte value containing ECU specific OBD system information.

Byte	Bit	Value	Description
0	0…3	Calculated	Current malfunction indicator activation mode. 0000: Activation Mode 1 0001: Activation Mode 2 0010: Activation Mode 3 0011: Activation Mode 4
	4…7	0x0	Reserved
1	0…7	Calculated	High byte of Continuous MI Counter (1h/bit)
2	0…7	Calculated	Low byte of Continuous MI Counter (1h/bit)
3	0…7	Calculated	High byte of highest ECU B1 counter (1h/bit)
4	0…7	Calculated	Low byte of highest ECU B1 counter (1h/bit)

Table 2-17 PID $91 implementation

- Related API function: Dem_DcmReadDataOfPID91()

2.2.10 OBD Related Diagnostic Services

This chapter describes those OBD related diagnostic services for which the functionality is partly implemented in the Dem. Note that additional services may be required in order to fulfill all legislative requirements (for details, refer to [6] and [8]).

For details how to access the Dem APIs, please refer to the sequence diagrams in the Dem specification [2] and to the description in the Dcm specification [3].

2.2.10.1 OBD II Services

OBDII This chapter is relevant for **OBD II**.

2.2.10.1.1 Mode $01: Request Current Powertrain Diagnostic Data

With Mode $01, the current value of any PID (except PID $02) can be read. For each PID calculated by the Dem, a dedicated API function is provided.

2.2.10.1.2 Mode $02: Request Powertrain Freeze Frame Data

To read the currently stored freeze frame, Mode $02 requires a specific sequence to read the data. It is implemented in the diagnostic handler (e.g. Dcm), but requires interaction with the Dem. According to this sequence, first the DTC number that stored the freeze frame (PID $02, see 2.2.9.1.2) is requested, then each PID that is part of the freeze frame can be read.

2.2.10.1.3 Mode $03: Request Emission-Related DTCs

This service is used to report all confirmed DTCs of an ECU.

To retrieve the confirmed OBD DTCs, a filter must be applied with DTCStatusMask = 0x08, DTCOrigin = “primary” and DTCKind = “emission related DTCs”.

For details how to read DTCs from the Dem, please refer to the sequence diagrams in the Dem specification [2] and to the description in the Dcm specification [3].

Note
Depending on the configuration, a confirmed DTC might not be externally visible until the driving cycle has been qualified. If a DTC was confirmed but is not yet externally visible when a Mode $03 request is started, the DTC will not be reported. (see also 2.2.4.2)

2.2.10.1.4 Mode $04: Clear/Reset Emission-Related DTCs

For the Dem, Mode $04 is equivalent to the UDS service $14 FFFFFF (clear all DTCs). If for the related event a “ClearEventAllowed” (refer to [1]) callback is configured, the application must also permit the clear request.

For DTCs with permanent status, additional conditions must be fulfilled before the DTC is removed from permanent memory (see 2.2.6.1.4).

2.2.10.1.5 Mode $06: Request Diagnostic Test Results

This service is used to report the supported MIDs and the latest related diagnostic test results.

Initially the supported MIDs are requested by the Dcm. Afterwards the number of TIDs for each supported MID is fetched. Using the supported MIDs and their related TIDs the respective DTR values can be retrieved.

2.2.10.1.6 Mode $07: Request Emission-Related DTCs Detected During Current or Last Completed Driving Cycle

This service is used to report all pending DTCs of an ECU.

To retrieve the pending OBD DTCs, a filter must be applied with DTCStatusMask = 0x04, DTCOrigin = “primary” and DTCKind = “emission related DTCs”.

2.2.10.1.7 Mode $0A: Request Emission-Related DTCs with Permanent Status

This service is used to report the DTCs that currently command the MIL indicator and therefore are stored in permanent memory.

To retrieve the permanent OBD DTCs, a filter must be applied with DTCStatusMask = 0x00, DTCOrigin = “permanent” and DTCKind = “emission related DTCs”

For details how to read DTCs from the Dem, please refer to the sequence diagrams in the Dem specification [2] and to the description in the Dcm specification [3].

Note
If Dem is configured to store a permanent DTC not before the end of the driving cycle, it may not be reported by Mode $0A although it already commands the MIL indicator. Refer to chapter 2.2.6.1.2 for more details regarding visibility of permanent entry.

After a clear request, permanent DTCs will still be reported (although the MIL indicator is inactive). This is an indication that the failure is still present or a performed repair action was not yet verified by the monitoring system of the ECU.

2.2.10.2 UDS, OBDonUDS and WWH-OBD Services

2.2.10.2.1 Service 14: Clear DTC

Providing a consistent readiness status and correct PID values it is defined by legislation that only clear all DTCs (for OBD II see 2.2.10.1.4) is allowed.

For enhanced diagnostic testers the UDS service to clear all DTCs ($14 0xFFFFFF) can be used to achieve this requirement.

Clearing all emission related DTCs ($14 0xFFFF33) has the same effect as clearing all DTCs ($14 0xFFFFFF).

Nevertheless, during production, it is necessary to clear a single DTC or a group of DTCs. Therefore, in addition to the UDS data the Dem will clear the following OBD related elements not only during clear all DTCs but also for a clear single DTC:

- The **Legislative Freeze Frame** is erased if the DTC to be cleared has caused the Freeze Frame storage.
- If the DTC to be cleared is a **Permanent DTC** the behavior described in chapter 2.2.6.1.4 will apply.

- **PID $21** will be reset to 0 if the DTC to be cleared is the last DTC that currently requests the MIL indicator. If the MIL indicator is still requested by a different DTC the PID is reset but continues to count as described in chapter 2.2.9.4.
- **PID $30**, **PID $31**, **PID $4D** and **PID $4E** will be always set to 0.
- **MIL Group trip count** is set to 0, if the DTC to be cleared is attached to a MIL Group.

WWH-OBD

For Clear DTC with GroupOfDTC set to **0xFFFFFF** or **0xFFFF33** the following items will also be cleared:

> activation mode
> Continuous-MI counter

Caution
As a clear single DTC must not be executed after production this functionality can be blocked by application using the manufacturer indication function of the Dcm (refer to [3]).

2.2.10.2.2 Service 19 04: Report DTC Snapshot Record by DTC Number

In general, the UDS service 19 04 reads the freeze frame of any DTC, but it can also be used to read the OBD freeze frame by specifying the DTC number of an OBD related event.

2.2.10.2.3 Service 19 12: Report Number of Emissions Related OBD DTCs by Status Mask

The number of OBD DTCs matching the requested DTC status mask can be retrieved from the Dem. Therefore, a filter must be applied with DTCKind = “emission related DTCs” and the status mask from the diagnostic request.

2.2.10.2.4 Service 19 13: Report Emissions Related OBD DTCs by Status Mask

To retrieve the emission related DTC numbers matching the requested DTC status mask, a filter must be applied with DTCKind = “emission related DTCs” and the status mask from the diagnostic request.

2.2.10.2.5 Service 19 15: Report DTCs with Permanent Status

For OBD II legislation the behavior of this service is the same as described for Mode $0A (see 2.2.10.1.7). For WWH-OBD and OBDonUDS the behavior of this service is the same as described for Service 19 55: Report Emission Related DTCs .

2.2.10.2.6 Service 19 42: Request Emission-Related DTCs with Status Mask

This chapter is only relevant for **WWH-OBD** and **OBDonUDS**.

To retrieve the emission related DTCs matching the DTC status mask, the DTC severity mask and the DTC status mask a filter must be applied with DTCKind = “emission related DTCs”.

2.2.10.2.7 Service 19 55: Report Emission Related DTCs with Permanent Status

This chapter is only relevant for **WWH-OBD** and **OBDonUDS**.

This service is used to report the DTCs that are stored in permanent memory.

To retrieve the permanent OBD DTCs, a filter must be applied with DTCStatusMask = 0x00 and DTCOrigin = “permanent”.

For details how to read DTCs from the Dem, please refer to the sequence diagrams in the Dem specification [2] and to the description in the Dcm specification [3].

2.2.10.2.8 Service 22: Read Data by Identifier

This chapter is only relevant for **WWH-OBD** and **OBDonUDS**.

Using the diagnostic service 22 the tester can retrieve the PIDs which are calculated by Dem (e.g. 0xF401 or 0xF491). The tester may retrieve the DTRs as calculated by Dem by using monitor IDs 0xF600 to 0xF6FF.

2.2.11 In-Use-Monitor Performance Ratio

The In-Use-Monitor Performance Ratio (IUMPR) is a measurement for how often an emission related monitoring function was active, compared to how often it should have been active. A ratio is calculated according to the following formula:

$$Ratio = \frac{Numerator}{Denominator}$$

- Numerator: number of DCY in which the monitoring function was active.
- Denominator: number of DCY in which the vehicle was operated under conditions that may allow the monitoring function to be active.

These counter values are calculated based on the presence or absence of different conditions which will be described in this chapter.

2.2.11.1 IUMPR Counters

2.2.11.1.1 Denominator

To increment a ratio specific denominator counter, at least a legislative set of general denominator conditions must be fulfilled. These conditions depend on the vehicle type, they are typically evaluated by the Master ECU and the result (conditions fulfilled / not fulfilled) is distributed over the bus system. This minimum set of conditions necessary to increment a ratio specific denominator are referred to as the general individual denominator conditions.

Depending on the monitored functionality, additional conditions may need to be fulfilled before a ratio-specific denominator can be incremented. These are referred to as special denominator conditions and must be reported separately to the Dem.

For use cases where the legislative conditions to increment the denominator are determined independently from the general individual denominator conditions, an independent denominator group is supported by the Dem.

For details how to report the conditions to the Dem, please refer to chapter 2.3.7.1.1.

2.2.11.1.2 Numerator

To increment a ratio specific numerator counter, the monitoring function must be able to detect a fault.

For symmetrical monitors, where the conditions to detect a failed and a passed result are the same, one test result (passed or failed) is enough to increment the numerator.

For asymmetrical monitors, where reporting a passed result does not imply that the monitor was able to detect a failed result, the numerator is only incremented if the conditions to detect a failure are given. This must be reported separately to the Dem. For details how to report the conditions, please refer to chapter 2.3.7.1.2.

2.2.11.1.3 General Denominator

The general denominator is a ratio independent counter that is incremented if the vehicle has been operated long enough under the legislative “general OBD conditions”. These conditions can be (but don’t have to be) the same as the general individual denominator conditions. General OBD conditions and general individual conditions are evaluated by the Master ECU and the result (conditions fulfilled / not fulfilled) is distributed over the bus system. For details how to report the conditions, please refer to chapter 2.3.7.1.3.

2.2.11.1.4 Ignition Cycle Counter

Basically, the ignition cycle counter represents the number of qualified driving cycles.

For hybrid vehicles, a second ignition cycle counter is provided, that counts the number of DCYs in which the combustion engine was active.

This second counter is only supported if an operation cycle of type *DEM_OPCYC_IGNITION_HYBRID* has been configured.

The ignition cycle counter and hybrid ignition cycle counter are not incremented if there is a pending fault relevant to detecting the DCY information. I.e. before incrementing those counters, the Dem checks that the FID used for calculating the ignition cycles is not disabled (see also chapters 2.3.3.3 and 2.3.7.2).

2.2.11.2 Counting Behavior

Calculation of the IUMPR numerator and denominator depends on the general individual denominator condition status, which is reported through the API `Dem_SetIUMPRDenCondition()`. This dependency is described in the following table:

General Individual Denominator Conditions	Denominator behavior	Numerator behavior
Inhibited	Locked until end of DCY	Locked until end of DCY
Not reached	All denominators depending on the general individual denominator conditions are locked until conditions are reached	Incremented when numerator conditions are fulfilled (see 2.3.7.1.2)
Reached	Incremented when additional denominator conditions are fulfilled (if required; see 2.3.7.1.1)	Incremented when numerator conditions are fulfilled (see 2.3.7.1.2)

Table 2-18 Dependency between ratio specific counters and general individual denominator conditions

The ratio specific denominator and numerator counters are also locked (i.e. not incremented) if a malfunction has been detected for the inputs of the related monitoring function (see also chapter 2.3.7.2). The counters remain locked as long as the fault is pending (Mode $07 fault), i.e. at least until the end of the next DCY (as shown in DCY2 and DCY3 of the following example).

Figure 2-24 shows which IUMPR counters are incremented based on the different conditions. The example uses a ratio belonging to a denominator group that requires additional denominator conditions to be incremented. Please note that the general individual denominator conditions are used for all ratios in the ECU at once, while all numerator signals are ratio specific. The (additional) denominator conditions are reported for the whole denominator group (cf. chapter 2.3.7.1.1), but each ratio has its own denominator.

The ratio specific denominator and numerator are incremented independently from each other as soon as the related conditions are reported as fulfilled and incrementing is not prevented by a pending fault (in the example: DCY0, DCY1 and DCY4).

For details regarding IUMPR input fault handling, see also chapter 2.3.7.2.

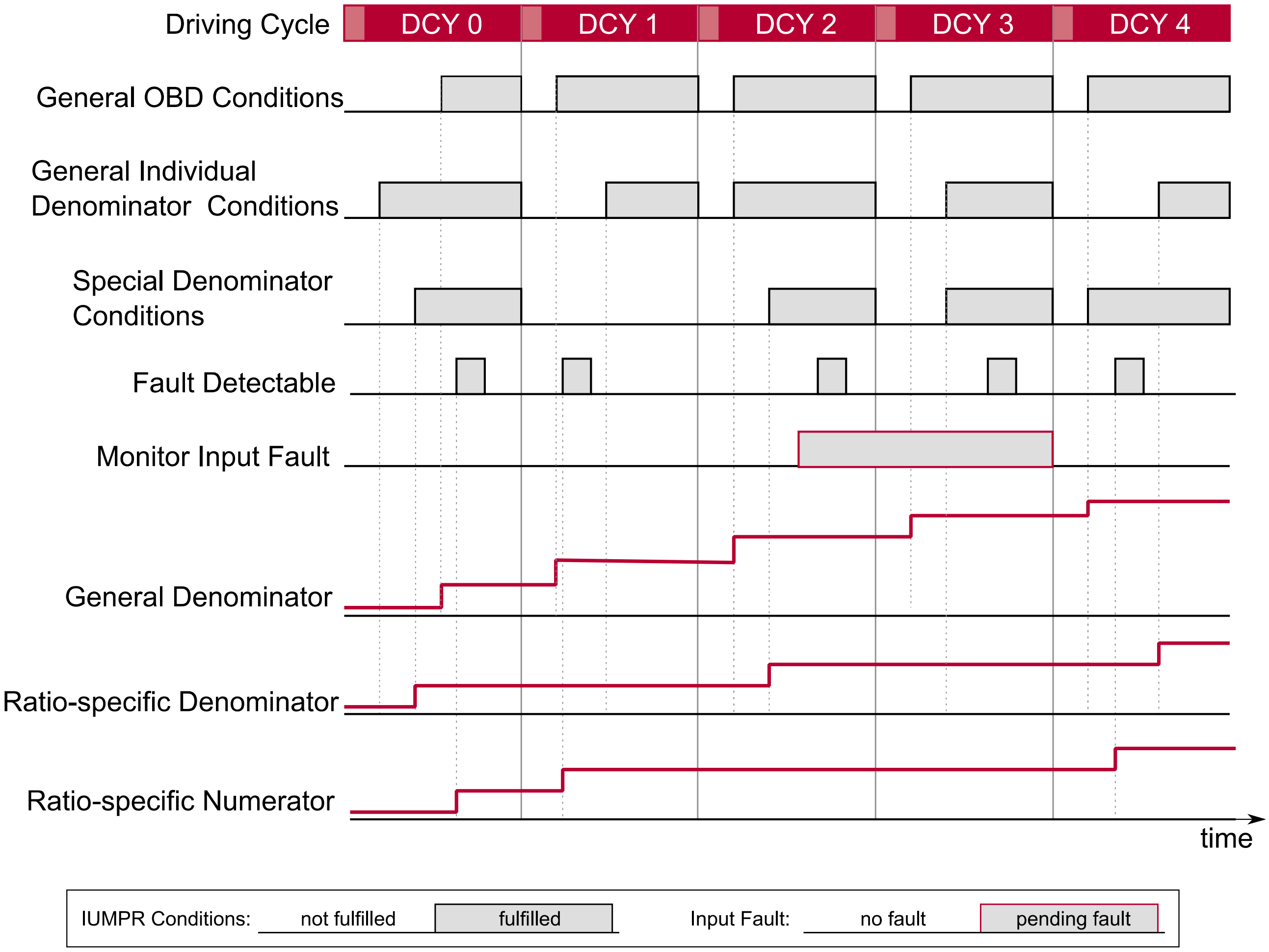

Figure 2-24 IUMPR Counting Behavior

2.2.11.3 Suppression Behavior

The Dem supports different mechanisms which are detailed described in [1] to “suppress” a DTC.

If the event or DTC that is related to a ratio is suppressed by using the API `Dem_SetEventSuppression()` the ratio will still be tracked in the Dem. Only the reporting of those ratios is skipped by the interfaces to retrieve the IUMPR data (refer to chapter 2.3.7.3). Further using suppression by API `Dem_SetEventAvailable()`, the ratio of the event is not tracked in the Dem and is not available for data reporting.

2.2.11.4 Effect of Power Take-Off

When PTO is activated, incrementing the ratio specific denominator and numerator counters is disabled for the events, which are affected by the Dem PTO handling. This means that:

> Numerator: incrementing the ratio specific numerator counter is disabled. For symmetrical monitors event status reports via `Dem_SetEventStatus()` are not accepted. For asymmetrical monitors reports that a fault could have been found (see `Dem_RepIUMPRFaultDetect()`) are not accepted.

- Denominator: the ratio specific denominator counter will not be incremented, if the conditions to increment the denominator are reported fulfilled while PTO is active. Furthermore, requests to lock and release the ratio specific denominator counters (refer to chapter 2.3.7.1.1) are rejected.

When PTO is deactivated, incrementing of all affected numerator and denominator counters will resume. Reports, which occurred while PTO was active, will not be processed retrospectively. Thus, to increment the affected counters all necessary conditions, which were reported fulfilled while PTO was active, need to be reported again.

2.2.12 Diagnostic Test Result

A Diagnostic Test Result (DTR) is defined by a Monitor Identifier (MID) and a Test Identifier (TID). It is handled by a DTR identifier generated during the Dem configuration process. A DTR consists of a data triple with test value, lower and upper limit representing the latest result of a qualified monitor test.

The current DTR values can be reported by the API `Dem_SetDTR()` using the appropriate DTR Id. The processing of the reported values depends on the DTR configuration, the used input parameters and the state of the assigned event.

The latest DTR values can be read

- for OBD II by using Mode $06 (see chapter 2.2.10.1.5).
- for WWH-OBD by using UDS service $22 (see chapter 2.2.10.2.8).
- for OBDonUDS:
  - by using UDS service $22 (see chapter 2.2.10.2.8).
  - by using UDS services (e.g., service $19 06) to read Extended Data Record 0x92 (see chapter 2.2.15.2), which contains all DTR values for a given DTC.

2.2.12.1 Processing of DTR values

If the control value *DEM_DTR_CTL_INVISIBLE* is used, the DTR Id is treated as not configured until new values are reported for it.

Using the control value *DEM_DTR_CTL_RESET* the test value, lower and upper limit are all reset to zero.

On each other control value the Dem either processes or ignores the reported values based on the configured update kind of the DTR Id and the status of the assigned event.

If the update kind is *DEM_DTR_UPDATE_ALWAYS* the DTR values are processed in any case.

On update kind *DEM_DTR_UPDATE_STEADY* for the processing at least the event's enable and storage conditions have to be fulfilled. In addition, it is required that the DTR values are consistent with the current event status. The values are processed according to Table 2-19.

Test value Event conditions	Processed when outside the limits	Processed when inside the limits
Enable or storage conditions not fulfilled		
TestNotCompletedThisOperationCycle == 1		
TestFailedThisOperationCycle == 1	■	■
TestNotCompletedThisOperationCycle == 0 && TestFailed == 0		■

Table 2-19 Processing of DTR values

2.2.12.2 Conversion of DTR values

If the DTR values are processed they are converted before storing them to non-volatile memory.

The conversion is done according to the configured conversion formula[1] for the particular DTR Id. After the linear mapping the values are corrected in case of rounding effects.

The Dem ensures that the relation between the test value and the two limits doesn't change due to the conversion. That is, if the limits are violated before the conversion, they shall still be violated after the conversion and vice versa. If rounding effects lead to a different relation between the three values, one or both limits are shifted by one.

If due to conversion or shifting the limits reach a value outside the valid range or if a shift by more than one would be necessary to correct the values, the DTR values are reset to zero. Also, an error is reported to the DET because this is the result of a wrong configuration of the conversion formula coefficients.

Using the control value DEM_DTR_CTL_NO_MIN a lower limit equal to the global lower limit is stored. Likewise, global upper limit is stored as upper limit if the control value DEM_DTR_CTL_NO_MAX is used.

[1] See /Dem/DemConfigSet/DemDtrs/DemDtr/DemDtrCompuDenominator0, DemDtrCompuNumerator0 and DemDtrCompuNumerator1 for more details.

Note
The valid range of DTR values after conversion is determined by the bit 7 of UaSID, which determines if the unsigned or signed 16bit integer range is used.

- DTR values with an UaSID < 0x80: Range from 0 up to +65535
  - Unsigned 16bit integer value in hexadecimal
    - Global lower limit: 0x0000
    - Global upper limit: 0xFFFF
- DTR values with an UaSID >= 0x80: Range from -32768 up to +32767
  - Signed 16bit integer value in hexadecimal
    - Global lower limit: 0x8000
    - Global upper limit: 0x7FFF

2.2.12.3 Resetting the DTR values

The Dem resets the DTR values corresponding to an event to zero, if

> the DTR Id is reported with control value *DEM_DTR_CTL_RESET* **or**

> the event assigned to the DTR Id is set unavailable (see *Event Availability* in [1]) **or**

> the event is affected by a clear request.

On a Mode $04 request the DTR values for all DTR Ids are reset to zero, independently of the assignment to an event.

2.2.12.4 Unavailable DTRs

A DTR Id is unavailable, if

> it is set invisible (see chapter 2.2.12.1) **or**

> if the assigned event is unavailable **or**

> if the DTC of the assigned event is suppressed (see *Suppress DTC* in [1]).

An unavailable DTR cannot be read by Dcm services. If all DTR Ids related to a MID are unavailable, the MID itself is unavailable (refer to chapter 2.3.8).

If the event assigned to a DTR Id is not available, the reporting of the DTR values by API `Dem_SetDTR()` is rejected independent of the control value or update kind.

2.2.13 Monitor Activity Ratio

This chapter is only relevant for **OBDonUDS**.

The Monitor Activity Ratio (MAR) is a measurement for how often an emission related monitoring function was active, compared to how often it could have been active. It is calculated by Dem for each OBD relevant event. It can be read out via Extended Data Record 0x93 (see chapter 2.2.15.3).

The MAR is defined as the ratio of an event-specific Monitor Activity Numerator (MAN) and a Monitor Activity Denominator (MAD) which is shared by all OBD relevant events. When the MAD reaches 255 the MAR is transferred to the Stored Monitor Activity Ratio (SMAR).

2.2.13.1 Monitor Activity Numerator

Each OBD relevant event has its own counter, the Monitor Activity Numerator (MAN). It is incremented at the end of the driving cycle, in which the corresponding monitoring function reported a passed result, unless any event that can illuminate the MIL was pending during this driving cycle. The MAN is incremented at most once per driving cycle.

The MAN is reset to 0 when the MAD reaches its maximum value of 255. It is possible that the MAN is bigger than the MAD and reaches 255 before the MAD. In this case the MAN latches at 255 until it is reset.

2.2.13.2 Monitor Activity Denominator

The Monitor Activity Denominator (MAD) is a counter which all OBD relevant events share. It counts how often the monitoring functions could have been active. It is incremented at the end of the driving cycle, in which the General Denominator (see chapter 2.2.11.1.3) was incremented, unless any event that can illuminate the MIL was pending during this driving cycle. The MAD freezes if any event that can illuminate the MIL is pending. It is immediately reset to 0 when it reaches its maximum value of 255.

The MAD can be read out via Info Type 0x79 (see chapter 4.1.1.19 Dem_DcmGetInfoTypeValue79()).

2.2.13.3 Stored Monitor Activity Ratio

If the MAD reaches its maximum value of 255 the current MAR is stored to the Stored Monitor Activity Ratio (SMAR) for each OBD relevant DTC. The SMAR is overwritten every time the MAD reaches its maximum value of 255. If multiple events are combined to the same DTC only the worst MAR, which is the smallest one, is stored as SMAR. Only available events are considered for the calculation of the worst MAR.

2.2.14 Non-Volatile Data Management

The Dem uses the standard AUTOSAR data management facilities provided by the NvM module.

2.2.14.1 NVRAM Write Frequency

Additional to the NVRAM blocks write frequency described in [1] the following NVRAM blocks are written with the mentioned OBD triggers.

NVRAM Item Write Frequency	OBD on UDS OBD II Freeze Frame Data	Permanent Data	IUMPR Data	DTR Data	Primary Entry
At shutdown – always			■	■	
At clear DTC	■	■	■		
OBD Freeze Frame storage/ update trigger occurs (see 2.2.5.1.1 for OBDII, 2.2.5.2.1 for OBDonUDS)	■				
OBD Freeze Frame gets reset (see 2.2.5.1.4 for OBDII, 2.2.5.2.2 for OBDonUDS).	■				
Permanent entry is created (see 2.2.6.1.2 and 2.2.6.2.1) or deleted (see 2.2.6.1.4 and 2.2.6.2.2) or state changes.		■			
Event status changes due to DCY qualification					■[1]
At call of API Dem_RequestNvSynchronization – if content has changed	■	■	■	■	

Table 2-20 NVRAM write frequency

[1] If the CDTC status bit is configured to be reset on overflow, otherwise the information is written into the Status Data block and is persisted at shutdown.

2.2.14.2 Data Recovery

Additional to the data recovery described in [1] the following corrections are done during initialization:

OBDII	> The legislative Freeze Frame is deleted if: > The Freeze Frame is configured to be stored on pending, and the event does not have confirmed nor pending status bit set. > The Freeze Frame is configured to be stored on confirmation, and the event does not have confirmed status bit set.
OBD on UDS	> The legislative Freeze Frame is deleted if > The event of the Freeze Frame does not have confirmed nor pending status bit set.
OBDII OBD on UDS	> The WarningIndicatorRequested bit (WIR bit) of an event is set if all below conditions are true: > An active permanent DTC is stored. > The event supports the MIL indicator but the WIR bit is not set. > The event is not a combined event. > The ConfirmedDTC bit (CDTC bit) of an event is also set when the WIR bit is recovered, but only if event combination type 2 is not supported in the configuration. > An active permanent DTC entry is deleted (set to the same state as after a clear request, refer to 2.2.6.1.4) if all conditions below are true: > The event supports the MIL indicator but the WIR bit is not set. > The event is a combined event. > A permanent DTC is created if: > An OBD relevant DTC triggers the MIL but has no permanent DTC stored.
WWH-OBD	> A permanent DTC is deleted (set to the same state as after Service $14, refer to 3.5.1.4) if: > The event does not fulfill all preconditions (see 2.2.6.2.1) to store a permanent DTC. > A permanent DTC is created if: > An event fulfills all preconditions (see 2.2.6.2.1) to store a permanent DTC but has none.

2.2.15 OBD Extended Data Records

OBD on UDS	This chapter is only relevant for **OBDonUDS**.

Extended data records are described in [1] in general. The extended data records with numbers 0x90 to 0x9F are specified to be OBD relevant in [11]. Details on these extended data records are provided below as far as they are not already defined in [1] or [11].

2.2.15.1 Extended Data Record 0x91

Extended data record 0x91 allows to read the IUMPR values for a specific DTC.

Byte	Entry
0	IUMPR Numerator High Byte
1	IUMPR Numerator Low Byte
2	IUMPR Denominator High Byte
3	IUMPR Denominator Low Byte

Table 2-21 Data part of extended data record 0x91

Because of event combination, there may be several IUMPR ratios available for a specific DTC. For these combined DTCs, the worst ratio of the combined group is reported which is usually the lowest ratio.

There are three edge cases for combined DTCs:

- If there are multiple events with the same lowest ratio value, then the ratio with the biggest denominator is reported.
- If there are multiple events with a numerator equal to zero and a denominator greater than zero, the ratio with the biggest denominator is reported.
- A ratio with numerator and denominator of zero is always considered the worst ratio.

The internal data element DEM_IUMPR can be used to configure extended data record 0x91.

2.2.15.2 Extended Data Record 0x92

Extended data record 0x92 allows to read all diagnostic test result values for a specific DTC. See chapter 2.2.12 for more details on diagnostic test results (DTR).

The data reported in extended data record 0x92 can be seen in Table 2-22. The 2 byte Unit and Scaling Identifier (UASID) reported in extended data record 0x92 only uses the low byte. UASIDs can only be configured up to 255 (1 byte of length), since Mode $06 limits the UASID to 1 byte.

Byte	Entry	Remark
0	Number of Test Results	
1	Unit And Scaling ID High Byte	Bytes 1 to 8 are reported once per test result
2	Unit And Scaling ID Low Byte	
3	Test Value High Byte	
4	Test Value Low Byte	
5	Minimum Test Limit High Byte	
6	Minimum Test Limit Low Byte	
7	Maximum Test Limit High Byte	
8	Maximum Test Limit Low Byte	

Table 2-22 Data part of extended data record 0x92

Because of event combination or because an event can have multiple DTRs configured, there may be several DTR values available for a specific DTC. Within the extended data record 0x92, the different DTR values are sorted by ascending configured Monitor Identifier (MID) Value. DTR values with identical MID Value are sorted by ascending TID-Value.

DTR values of unavailable events are reported with default values, all bytes are set to zero.

The internal data element DEM_DTR can be used to configure extended data record 0x92.

2.2.15.3 Extended Data Record 0x93

Extended data record 0x93 allows to read the Monitor Activity Data (see chapter 2.2.13) of an OBD relevant DTC. The internal data element DEM_MONITOR_ACTIVITY_DATA can be used to configure the extended data record 0x93.

Byte	Entry
0	Monitor Activity Numerator
1	Stored Monitor Activity Ratio

Table 2-23 Data part of extended data record 0x93

If multiple events are combined to the same DTC there are several MANs available per DTC. In this case the worst MAN of the DTC, which is the smallest one, is reported. Only the MANs of available events are considered for the calculation of the worst MAN.

2.2.16 J1939

Note
Dependent on the licensed components of your delivery the feature J1939 may not be available in DEM.

2.2.16.1 OBD II

OBDII This chapter is only relevant for **OBD II**.

If both, J1939 and OBD II, are licensed, the Diagnostic Messages listed in Table 2-24 are supported. Other diagnostic messages may be rejected, or the response message may be incorrect.

Diagnostic Message
DM1 – Active Diagnostic Trouble Codes
DM2 – Previously Active Diagnostic Trouble Codes
DM5 -- Diagnostic Readiness 1

Diagnostic Message
DM6 – Emission-Related Pending Diagnostic Trouble Codes
DM11 - Diagnostic Data Clear/Reset for Active DTCs
DM12 – Emission-Related MIL-On Diagnostic Trouble Codes
DM20 – Monitor Performance Ratio
DM21 – Diagnostic Readiness 2
DM22 - Individual Clear/Reset of Active and Previously Active DTC
DM23 – Emission-Related Previously MIL-On Diagnostic Trouble Codes
DM24 – SPN Support
DM25 – Expanded Freeze Frame
DM26 – Diagnostic Readiness 3
DM27 – All Pending DTCs
DM28 – Emission-Related Permanent Diagnostic Trouble Codes
DM29 – DTC Counts

Table 2-24 Diagnostic messages supported by Dem for OBD II

The Dem interfaces related to these Diagnostic Messages are described in chapter 4.1.3 J1939Dcm and in [1]. The order in which the interfaces must be called for the different Diagnostic Messages and the respective input parameters are described in [4].

2.2.16.1.1 Diagnostic Message 5: Diagnostic Readiness 1

For OBDII a request to read diagnostic readiness information returns the readiness information in the format described in [13].

The readiness information is calculated similar to PID $01 and PID $F501 (see 2.2.9.1). The readiness calculation is also affected by Power Take-Off as described in 2.2.9.1.1.

2.2.16.1.2 Diagnostic Message 11: Diagnostic Data Clear/Reset for Active DTCs

For OBD II a request to clear active J1939 DTCs clears all diagnostic information (see [13]). The request clears not only J1939 DTCs, but all DTCs stored in the primary memory. If for the related event a “ClearEventAllowed” (see [1]) callback is configured, the application must also permit the clear request.

2.2.16.1.3 Diagnostic Message 20: Monitor Performance Ratio

For OBD II it is supported to read the SPN, numerator and denominator of In-Use-Monitor Performance (IUMPR) Ratios. Multiple ratios can be assigned to the same SPN value because their events are combined to the same J1939 DTC or because multiple J1939 DTCs with ratios have the same SPN value.

If multiple ratios belong to the same SPN the worst of these ratios is reported. The worst ratio is usually the ratio with the lowest numerical value. There are two edge cases:

- If there are multiple ratios with the same lowest value, the ratio with the biggest denominator is reported.
- A ratio with numerator and denominator of zero is always considered the worst ratio.

2.2.16.1.4 Diagnostic Message 21: Diagnostic Readiness 2

For OBDII a request to read diagnostic readiness 2 information returns the readiness 2 information in the format described in [13].

2.2.16.1.5 Diagnostic Message 22: Individual Clear/Reset of Active and Previously Active DTC

OBD relevant DTCs cannot be cleared individually using DM22.

2.2.16.1.6 Diagnostic Message 24: SPN Support

For OBDII, a request to read supported SPNs returns the SPN Ids assigned to the PIDs of the Mode $02 OBD Freeze Frame in the format described in [13].

2.2.16.1.7 Diagnostic Message 25: Expanded Freeze Frame

For OBDII, a request to read J1939 Expanded Freeze frame shall return the Mode $02 OBD freeze frame in the format described in [13].

2.2.16.1.8 Diagnostic Message 26: Diagnostic Readiness 3

For OBDII a request to read diagnostic readiness 3 information returns the readiness 3 information in the format described in [13].

The (Non-)Continuously Monitored Systems Enabled/Completed Status reported in diagnostic readiness 3 is calculated similar to PID $41 (see 2.2.9.7). The same configuration options (DemGeneral/DemGeneralOBD/DemOBDCompressionMisfireAlwaysComplete, DemGeneral/DemGeneralOBD/DemOBDFuelSystemAlwaysComplete) and conditions to report a monitoring function as supported or not supported apply. The readiness groups DEM_OBD_RDY_FLSYS and DEM_OBD_RDY_FLSYS_NONCONT are considered in the same way (see info box in chapter 2.2.9.7).

In contrast to PID $41 an emission related monitoring function is reported as completed or N/A if

- no event is assigned **OR**
  no event is available in the related readiness group
- all events that are available and assigned to the related readiness group are tested in the current cycle.

2.2.16.1.9 Runtime Limitation for Diagnostic Messages

For OBD II the runtime limitation is also applied to the Diagnostic Messages DM6, DM12, DM23 and DM28. For more information on the runtime limitation refer to [1].

2.2.16.2 OBDonUDS

OBD on UDS This chapter is only relevant for **OBDonUDS**.

If both, J1939 and OBDonUDS, are licensed, the Diagnostic Messages listed in Table 2-25 are supported. Other diagnostic messages may be rejected, or the response message may be incorrect.

Diagnostic Message
DM1 – Active Diagnostic Trouble Codes
DM2 – Previously Active Diagnostic Trouble Codes

Table 2-25 Diagnostic messages supported by Dem for OBDonUDS

The Dem interfaces related to these Diagnostic Messages are described in [1]. The order in which the interfaces must be called for the different Diagnostic Messages and the respective input parameters are described in [4].

2.3 Integration

2.3.1 Operation Cycle Handling

2.3.1.1 Driving Cycle

To qualify and restart the DCY, the API `Dem_SetOperationCycleState()` is used. The application must call this API if the according operation cycle information (normal driving conditions, ignition cycle) has been received from the OBD Master ECU.

The mapping between the operation cycle state needed by the API and the effect on the driving cycle is described in Table 2-26 and Figure 2-25.

OpCycleState	Effect on DCY	Precondition	Trigger
START	qualify DCY	DCY must not be qualified	normal driving conditions are fulfilled (indicated by OBD Master)
END	restart DCY	DCY must be qualified	ignition state changes from “OFF” to “ON” Or ECU is initialized

Table 2-26 OBD Driving Cycle Handling

Accordingly, the API `Dem_GetOperationCycleState()` will return the current qualification state when called for the driving cycle.

DCY status	Result of `Dem_GetOperationCycleState()`
Qualified	START
Not qualified	END

Table 2-27 OBD Driving Cycle Qualification state

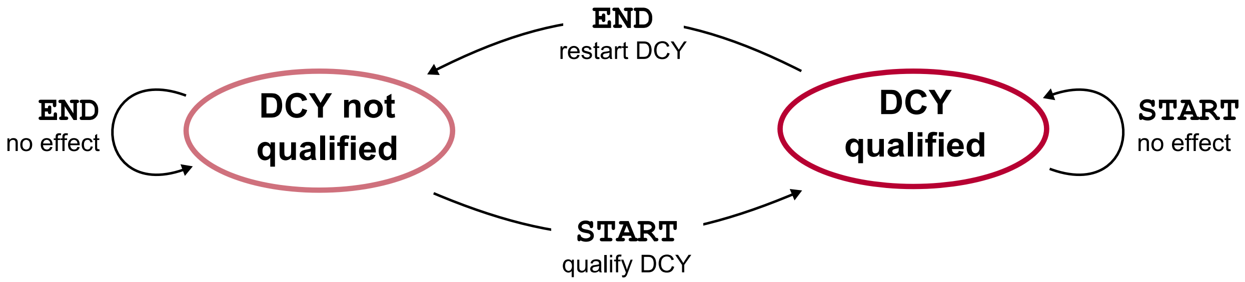

Figure 2-25 OBD Driving Cycle State Transitions

2.3.1.2 Warm-Up Cycle

If the Dem is configured not to delay the warm-up cycle (see chapter 2.2.2.4.1) until the end of a qualified driving cycle, the aging of PIDs (PID $4D, PID $21) is done as soon as the DEM receives warm-up cycle start trigger (or restart trigger, if it is already started) from the application, i.e., when the warm-up cycle is started the 40th time without the MIL being active, both PIDs are reset. The aging of DTCs using the warm-up cycle as aging cycle (i.e., at least all OBD related DTCs) is done as soon as the DEM receives warm-up cycle stop trigger (or restart trigger, if it is already started) from the application. To ensure legislative conform healing and aging of these events, the application must ensure that the warm-up cycle is started only once per driving cycle and that the end of the warm-up cycle is processed before the end of the driving cycle (time between reporting operation cycle states > DEM task time).

Enabling the feature to delay the warm-up cycle the Dem will use the warm-up cycle start trigger from application to mark the current driving cycle as having met the warm-up conditions and therefore the aging of PIDs and DTCs are done at the end of a qualified driving cycle. It is not necessary to explicitly end the warm-up cycle, an according API call will be ignored. Therefore, it is not recommended to use warm-up cycle as a normal operation cycle for DTCs (see chapter 2.2.1.3.4).

2.3.2 Required Data

The following data is needed by the Dem for OBD functionality and must be provided to the Dem during integration.

Data	Legislation	Source	Description
DCY	OBDII OBD on UDS WWH-OBD	Master ECU	The signal is provided via the bus system to indicate that normal driving conditions are fulfilled. If DCY occurs on the bus, the driving cycle must be qualified using `Dem_SetOperationCycleState()`.
WUC	OBDII OBD on UDS WWH-OBD	Master ECU	The signal is provided via the bus system to indicate that the vehicle is currently in the warm-up phase. When the WUC occurs on the bus, the warm-up cycle must be started by using `Dem_SetOperationCycleState()`.

Data	Legislation	Source	Description
PFC	OBDII OBD on UDS	Master ECU	The signal is provided via the bus system to indicate that the vehicle has been operated under general conditions. The occurrence of PFC on the bus is reported to the Dem using `Dem_SetPfcCycleQualified()`. It is used to remove permanent DTCs after a Mode $04 request (see 2.2.6.1.4).
Ignition Cycle Hybrid	OBDII OBD on UDS WWH-OBD	Master ECU	This signal is only available in hybrid vehicles. If it is supported, an according operation cycle of type DEM_OPCYC_IGNITION_HYBRID must be configured. When indicated by the Master ECU, this operation cycle has to be started by using `Dem_SetOperationCycleState()`. Starting this operation cycle will increment the second ignition cycle counter, which is required for hybrid vehicles.
Legislative General OBD Conditions	OBDII OBD on UDS WWH-OBD	Master ECU	The conditions to increment the general denominator (sometimes called “Normed Trip”) are typically calculated by the Master ECU and distributed over the bus system. The application has to report those conditions to the Dem using the API `Dem_SetIUMPRDenCondition()` (see 2.3.7.1.3).
Denominator Conditions	OBDII OBD on UDS WWH-OBD	Master ECU	The general individual denominator conditions as well as the special denominator conditions to increment the ratio specific denominators are reported to the Dem using the API `Dem_SetIUMPRDenCondition()` (see 2.3.7.1.1).

Data	Legislation	Source	Description
Distance Information	OBDII OBD on UDS WWH-OBD	Application	This 4 byte value with the travelled distance is needed to calculate PID $21 (2.2.9.4) and PID $31 (2.2.9.6) and to determine Permanent Storage Activation (2.2.6.1.5). The callback used by the Dem to request the distance information is configured with the parameter “OBD Input Distance Information”. If no valid value can be provided, the callback implementation must return E_NOT_OK. The Dem will continue to trigger this callback until E_OK will be returned. The returned value must have a resolution of 1 kilometer or 1 mile per bit.
Engine Run Time Minutes	OBDII OBD on UDS WWH-OBD	Application	This 4 byte value with the time since engine start in minutes is needed to calculate PID $4D (see 2.2.9.8), PID $4E (see 2.2.9.9) and PTO activation timer (see 2.2.3.8.2). The callback used by the Dem to request the engine run time in minutes is configured with the parameter “Time Since Engine Start Minutes”. If no valid value can be provided, the callback implementation must return E_NOT_OK. The Dem will continue to trigger this callback until E_OK will be returned. The returned value must have a resolution of 1 minute per bit and it must be provided in big endian byte order.
Engine Run Time Seconds	OBDII	Application	This 4 byte value with the time since engine start in seconds is needed to support J1939 diagnostic readiness 3 (see 2.2.16). The callback used by the Dem to request the engine run time in seconds is configured with the parameter “Time Since Engine Start Seconds”. If no valid value can be provided, the callback implementation must return E_NOT_OK. The Dem will continue to trigger this callback until E_OK will be returned. The returned value must have a resolution of 1 second per bit and it must be provided in big endian byte order.

Table 2-28 Required data for OBD

Caution
For hybrid vehicles or for vehicles that employ engine shutoff strategies (e.g., engine shutoff at idle) additional conditions shall increment

- PID $4D and PID $4E (see [12])
- Minutes run by engine while MIL is activated and time since diagnostic trouble codes cleared reported in J1939 diagnostic readiness 2 (see [13])
- Time since engine start reported in J1939 diagnostic readiness 3 (see[13]).

The application must adapt the engine runtime value provided via callback so that the values are incremented when the conditions are fulfilled.

2.3.3 Configuration Aspects

2.3.3.1 NvM Storage

The feature “ImmediateNvRamStorage” must be enabled globally in the Dem/General section.

2.3.3.2 MIL Indicator

As OBD events which are currently commanding the MIL indicator need special treatment, it is important for the Dem to know if an indicator referenced by the event is the MIL indicator or if it is a standard AUTOSAR indicator.

Thus the Indicator representing the MIL must be explicitly specified through the configuration parameter “MIL Indicator Ref”.

2.3.3.3 FiM References

For IUMPR, the Function Inhibition Manager (FiM; see [5]) is required. FiM FIDs are used to lock IUMPR counters if an input fault is detected (see also 2.3.7.2).

The following FiM FIDs must be configured:

Configuration Parameter	FID Description
DemGeneral/DemRatioId/DemFunctionIdRef	At least one Function Id must be referenced by each ratio. It is used to determine if the inputs of an OBD related monitoring function are valid or not. If the related function is disabled, the increment of the ratio specific numerator and denominator is inhibited.
DemGeneral/DemIgnitionCycleFunctionIdRef	Function Id that contains the validity of the ignition cycle signal (e.g. if the according CAN message have been received or not). If the related function is disabled, only the increment of the ignition cycle counter and hybrid ignition cycle counter is inhibited. This FID doesn’t affect the incrementation of other IUMPR counters. Use DemGeneral/DemRatioId/DemFunctionIdRef instead for locking ratio specific counters.

Table 2-29 FIDs required by IUMPR

2.3.3.4 Event Behavior

Basically, OBD events use the same mechanisms as defined by AUTOSAR and UDS. However, to fulfill the legislative regulations, some parameters require fixed settings. The main distinctive criterion for the Dem to handle an event as OBD related (e.g. store a freeze frame or a permanent DTC) is the existence of the MIL indicator.

Please refer to chapter 2.4 for a recommended Dem configuration.

2.3.3.5 Permanent DTCs

Emission related systems must reserve non-volatile memory for failure codes of at least four permanent DTCs.

A chronology of confirmed DTCs is maintained to fill-up the permanent memory if a previously occupied entry is freed. The size of the permanent chronology must be configured to the maximum number of DTCs that can store a permanent DTC (OBD II/OBDonUDS: DTCs that command the MIL, WWH-OBD: the sum of Class A DTCs and B1 DTCs) at the same time in a (realistic) worst case scenario. The chronology has no effect if it’s size is less or equal the size of the memory for permanent DTCs.

2.3.4 Development Error Codes

If the Dem is used for OBD, additional service IDs are defined for development error reporting to DET.

The following list extends the service IDs defined in [1]:

Service ID	Service
0x51	Dem_SetEventDisabled
0x52	Dem_DcmReadDataOfOBDFreezeFrame
0x53	Dem_DcmGetDTCOfOBDFreezeFrame
0x55	Dem_MainFunction
0x61	Dem_DcmReadDataOfPID01
0x63	Dem_DcmReadDataOfPID1C
0x64	Dem_DcmReadDataOfPID21
0x65	Dem_DcmReadDataOfPID30
0x66	Dem_DcmReadDataOfPID31
0x67	Dem_DcmReadDataOfPID41
0x68	Dem_DcmReadDataOfPID4D
0x69	Dem_DcmReadDataOfPID4E
0x6A	Dem_DcmReadDataOfPID91
0x6B	Dem_DcmReadDataOfPIDF501
0x71	Dem_RepIUMPRDenLock
0x72	Dem_RepIUMPRDenRelease
0x73	Dem_RepIUMPRFaultDetect
0x99	Dem_J1939DcmSetRatioFilter
0x9A	Dem_J1939DcmGetNextFilteredRatio
0x9C	Dem_J1939DcmReadDiagnosticReadiness2

Service ID	Service
0x9D	Dem_J1939DcmReadDiagnosticReadiness3
0xA2	Dem_SetDTR
0xA3	Dem_DcmGetAvailableOBDMIDs
0xA4	Dem_DcmGetNumTIDsOfOBDMID
0xA5	Dem_DcmGetDTRData
0xAA	Dem_SetPfcCycleQualified
0xAE	Dem_SetIUMPRDenCondition
0xB2	Dem_DcmGetDTCSeverityAvailabilityMask
0xB3	Dem_ReadDataOfPID01
0xB4	Dem_GetB1Counter
0xD4	Dem_EnablePermanentStorage
0xD5	Dem_GetIUMPRGeneralData
0xD7	Dem_GetNextIUMPRRatioDataAndDTC
0xD8	Dem_GetCurrentIUMPRRatioDataAndDTC
0xD9	Dem_GetPermanentStorageState
0xDA	Dem_IUMPRLockNumerators
0xDD	Dem_SetIUMPRFilter
0xDE	Dem_GetNumberOfFilteredIUMPR
0xDF	Dem_UpdateAvailableOBDMIDs
0xE3	Dem_SetDTCFilterByReadinessGroup()
0xF3	Dem_SetHideObdOccurrences
0xF4	Dem_GetHideObdOccurrences

Table 2-30 OBD specific DET service IDs

2.3.5 Critical Sections

For OBD, the same exclusive areas as for the standard Dem are used (see [1]). However, additional OBD APIs are assigned to these areas. These are:

Exclusive Area 0

> `Dem_RepIUMPRFaultDetect()`

> `Dem_RepIUMPRDenLock()`

> `Dem_RepIUMPRDenRelease()`

> `Dem_SetIUMPRDenCondition()`

> `Dem_IUMPRLockNumerators()`

> `Dem_GetNextFilteredExtendedDataRecord()`
> (if used to access DTR values via ERec 0x92)

- `Dem_GetNextExtendedDataRecord()`
  (if used to access DTR values via ERec 0x92)

Exclusive Area 1

- `Dem_SetEventDisabled()`

Exclusive Area 2

- `Dem_EnablePermanentStorage()`

2.3.6 NVM Integration

Basically, the NVM handling for the OBD Dem is the same as already described in the regular Dem Technical Reference (see [1]).

This chapter only lists additional NVM blocks, initialization functions and the write frequency that are needed for OBD.

2.3.6.1 NVRAM Demand

NVRAM Item	Legislation	RAM buffer symbol	Type
Freeze Frame Data	OBDII OBD on UDS	Dem_Cfg_FreezeFrameData	Dem_Cfg_FreezeFrameDataType
Permanent Data	OBDII OBD on UDS WWH-OBD	Dem_Cfg_PermanentData	Dem_Cfg_PermanentDataType
IUMPR Data	OBDII OBD on UDS WWH-OBD	Dem_Cfg_ObdIumprData	Dem_Cfg_ObdIumprDataType
DTR Data	OBDII OBD on UDS WWH-OBD	Dem_Cfg_DtrData	Dem_Cfg_DtrDataType

Table 2-31 OBD NVRAM blocks

2.3.6.2 NVRAM Initialization

NVRAM Item	Legislation	Initialization
Freeze Frame Data	OBDII OBD on UDS	Call `Dem_NvM_InitObdFreezeFrameData()`
Permanent Data	OBDII OBD on UDS WWH-OBD	Call `Dem_NvM_InitObdPermanentData()`
IUMPR Data	OBDII OBD on UDS WWH-OBD	Call `Dem_NvM_InitObdIumprData()`
DTR Data	OBDII OBD on UDS WWH-OBD	Call `Dem_NvM_InitDtrData()`

Table 2-32 OBD NVRAM initialization

Caution
The Dem reinitializes its NVRAM structures automatically if the configuration id, the major or minor version of the module has changed or if the NvM initialization Callback for Admin Data is called.

This does **not** apply to the permanent DTC NVRAM block. Hence the application needs to decide by itself if the legislative preconditions are fulfilled to reinitialize the permanent data, given that the EventId/DTC mapping was not changed.

If the configuration was changed (through reprogramming or post-build) and has a modified EventId/DTC mapping, the permanent DTC NVRAM block must be reinitialized since it contains the EventIds related to the permanent DTCs. Otherwise, the reported permanent DTCs might be wrong and those DTCs might not be removable from the permanent memory through the regular mechanisms anymore.

If the permanent DTC NVRAM block is reinitialized ensure that all other NVRAM structures are also reinitialized.

Caution
The NvM initialization callbacks listed in section 2.3.6.2 must only be invoked after Dem's pre-initialization, but before Dem is initialized.

2.3.7 IUMPR Integration

2.3.7.1 IUMPR Conditions

To trigger or lock the increment of IUMPR counters, AUTOSAR defines several APIs and configuration parameters in order to fulfill the legislative requirements. When to use which API is described in this chapter.

Caution
The processing of IUMPR (numerators, denominators and general denominator) is deferred to the Dem_MainFunction. Do not call Dem_Shutdown before ratios have been processed, since this would prevent the ratios from updating.

2.3.7.1.1 Denominator

The general individual denominator conditions can be reported using the API `Dem_SetIUMPRDenCondition()` with ConditionId = DEM_IUMPR_GENERAL_INDIVIDUAL_DENOMINATOR (see 4.1.2.9).

Additional conditions for the ratio specific denominator are reported depending on the configuration parameter DemIUMPRDenGroup.

Denominator Group	Description
DEM_IUMPR_DEN_NONE	No additional conditions are required for the monitored functionality. The ratio-specific denominator is incremented when the general individual denominator conditions are fulfilled.
DEM_IUMPR_DEN_COLDSTART DEM_IUMPR_DEN_EVAP	The monitor is part of a system where additional conditions need to be fulfilled before the denominator is incremented. These conditions are reported similar to the general conditions by using `Dem_SetIUMPRDenCondition()` with the corresponding ConditionId.

Denominator Group	Description
DEM_IUMPR_DEN_500MI	Similar to DEM_IUMPR_DEN_COLDSTART and DEM_IUMPR_DEN_EVAP, monitor is part of a system where additional conditions need to be fulfilled before the denominator is incremented. These conditions are reported similar to the general conditions by using `Dem_SetIUMPRDenCondition()` with the corresponding ConditionId. In contrast to the aforementioned denominator groups, incrementing the 500 miles denominator does not require general individual denominator conditions to be fulfilled in the same driving cycle. The ratio-specific 500 miles denominator is incremented when the additional conditions are reported as fulfilled if the general individual denominator conditions were fulfilled at least once since the previous 500 miles denominator increment of that ratio.
DEM_IUMPR_DEN_PHYS_API	As for DEM_IUMPR_DEN_NONE, the denominator is incremented when the general individual denominator conditions are fulfilled. Additionally, an API (`Dem_RepIUMPRDenLock()`) is provided to lock the denominator if other, non-standardized conditions need to be fulfilled before the denominator can be incremented. As soon as the locked state is released again (via API `Dem_RepIUMPRDenRelease()`) while general individual denominator conditions are fulfilled, the denominator is incremented. The locking state is not persisted when restarting the next DCY. `Dem_RepIUMPRDenLock()` must be called at the beginning of each DCY before the general conditions are fulfilled.
DEM_IUMPR_DEN_INDEPENDENT	As an exception to all other denominator groups, the legislative conditions required to increment the denominators in this denominator group are determined independently from the general individual denominator conditions. Note that inhibiting the general individual denominator condition still locks the independent denominator ratios just as all the other denominator groups. As with denominator group DEN_PHYS_API, an API (`Dem_RepIUMPRDenLock()`) is provided to lock the denominator if other, non-standardized conditions need to be fulfilled before the denominator can be incremented. As soon as the locked state is released again (via API `Dem_RepIUMPRDenRelease()`) while independent denominator conditions are fulfilled, the denominator is incremented. The locking state is not persisted when restarting the next DCY. `Dem_RepIUMPRDenLock()` must be called at the beginning of each DCY before the independent conditions are fulfilled.

Table 2-33 Triggers to increment the Denominator

The special denominator conditions can be reported with ConditionStatus REACHED or NOT_REACHED (using the API `Dem_SetIUMPRDenCondition()`).

The general individual denominator conditions can additionally be reported with status INHIBITED (using the API `Dem_SetIUMPRDenCondition()`) to indicate that the general individual denominator conditions cannot be evaluated. In this case, all ratios (numerators and denominators) are locked until the restart of the driving cycle.

Note
The return code of `Dem_SetIUMPRDenCondition()` only indicates that the denominator conditions were successfully reported to the Dem. It does not indicate if (or how many) ratios were incremented. That is: For ratios that are locked (due to monitor input faults, cf. chapter 2.3.7.2) when processing the reported denominator conditions, the ratio-specific denominator is not incremented.

2.3.7.1.2 Numerator

For the ratio specific numerator, the conditions are reported depending on the configuration parameter DemRatioIdType.

RatioId Type	Description
DEM_RATIO_API	The numerator is incremented when the ability of the monitoring function to detect a fault is reported via API `Dem_RepIUMPRFaultDetect()`. This setting is used for asymmetrical monitors.
DEM_RATIO_OBSERVER	The numerator is incremented when a test result (passed or failed) is reported via `Dem_SetEventStatus()`. This setting is used for symmetrical monitors.

Table 2-34 Triggers to increment the Numerator

To prevent the numerators from an erroneous increment, e.g. during an actuator-test the Dem offers the interface `Dem_IUMPRLockNumerators()` which will lock the numerator of each ratio for the current driving cycle.

Caution
The API `Dem_RepIUMPRFaultDetect()` is only callable from the master partition. To avoid overhead due to cross partition communication asymmetrical monitors (RatioId Type set to DEM_RATIO_API) should be mapped to the same partition as the Dem Master (refer to [1]).

2.3.7.1.3 General Denominator

In primary ECUs, the information if the legislative general OBD conditions are fulfilled is received from the Master ECU and reported to the Dem using the API `Dem_SetIUMPRDenCondition()` with ConditionId = `DEM_IUMPR_GENERAL_OBDCOND` (see 4.1.2.9).

If the condition signal cannot be read by the application, it must be reported to the Dem with ConditionStatus `DEM_IUMPR_DEN_STATUS_INHIBITED`. The inhibition status cannot be changed until the driving cycle is restarted.

Changes

Since implementation version 18.03.00, the `DEM_IUMPR_GENERAL_DENOMINATOR` condition was replaced with two separate conditions:
`DEM_IUMPR_GENERAL_INDIVIDUAL_DENOMINATOR` and
`DEM_IUMPR_GENERAL_OBDCOND`.

For backward compatibility, it is possible to report the legislative general OBD conditions together with the individual general denominator conditions using ConditionId = `DEM_IUMPR_GENERAL_DENOMINATOR`. This has the same effect as reporting `DEM_IUMPR_GENERAL_OBDCOND` and `DEM_IUMPR_GENERAL_INDIVIDUAL_DENOMINATOR` one after another.

Do not mix the backwards compatibility option together with separate reporting of the conditions.

2.3.7.2 Input Faults

According to the legislative regulations, incrementing the IUMPR counters shall be disabled if a malfunction has been detected (i.e. a pending fault code has been stored) that disables the related monitoring function.

In AUTOSAR, disabling a software function due to input faults can be achieved using the Function Inhibition Manger (FiM; see [5]). This BSW module uses Function Identifiers (FIDs) that are associated to Dem events to determine, if software functionalities need to be disabled.

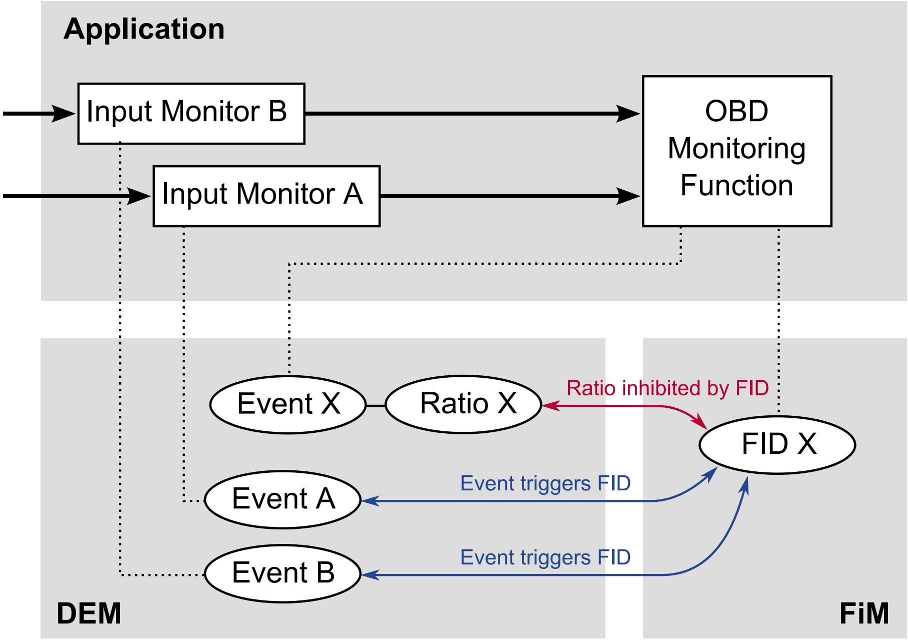

Figure 2-26 Relation between Dem and FiM for IUMPR

To use this existing mechanism for IUMPR, each ratio references at least one FiM FID that is checked by the Dem before incrementing the ratio specific numerator and denominator (see also 2.3.3.3).

Figure 2-26 shows an example, how a FID may be used to lock an IUMPR ratio. If Input Monitor A or Input Monitor B report a fault which disables the OBD monitoring function, the counters of the related Ratio X must be locked. Thereto FID X will be checked by the Dem, as it contains this information.

Locking of IUMPR counters affects not only the ratio specific counters, but also the Ignition Cycle Counter and Hybrid Ignition Cycle Counter. To indicate that a pending fault exists for the DCY signal (and thus for the Ignition Cycle Counters), a FID representing the status of the respective bus signal must be defined and configured (see also 2.3.3.3).

2.3.7.3 Retrieving IUMPR Data

Since AUTOSAR does not define any application interface to read the IUMPR counters, for this purpose the Vector Dem provides proprietary APIs, which are intended to be used in a sequence to read out all available ratios en bloc.

The first API `Dem_SetIUMPRFilter()` (see 4.1.2.10) starts this sequence, while `Dem_GetNumberOfFilteredIUMPR()` (see 4.1.2.11) will provide the number of ratios matching the given filter criteria. Each subsequent call to `Dem_GetNextIUMPRRatioDataAndDTC()` (see 4.1.2.13) returns the available ratios one after another. If no more ratios are available, E_NOT_OK is returned.

To restart reading the ratios, `Dem_SetIUMPRFilter()` must be called again.

If during the read sequence, the same ratio should be read multiple times, the API `Dem_GetCurrentIUMPRRatioDataAndDTC()` (see 4.1.2.14) can be used.

Figure 2-27 provides an overview of the behavior described above.

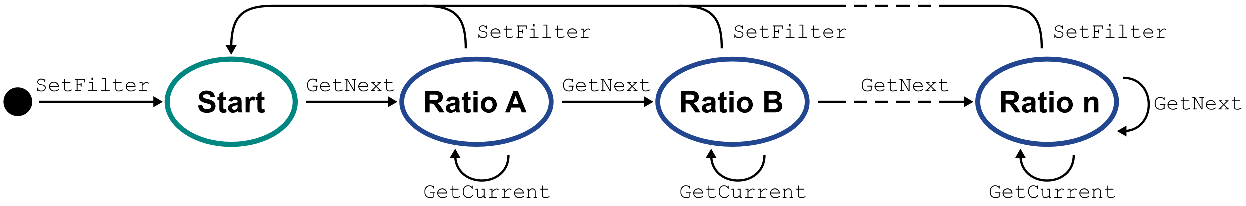

Figure 2-27 Retrieving IUMPR data

Using feature "Event Combination" (for a description, refer to [1]) each event that is configured to support a ratio will track its own numerator and denominator. Therefore, the APIs described above will return the ratio specific data but because of event combination the UDS and OBD DTC will be reported multiple times.

Practical Procedure

- Call `Dem_GetIUMPRGeneralData()`
  This will return the general denominator and the ignition cycle counter(s).
- Call `Dem_SetIUMPRFilter()`
  This will initialize the IUMPR filter criteria to the given memory location and IUMPR readiness group.
- Call `Dem_GetNumberOfFilteredIUMPR()`
  This will return the number of filtered ratios matching the filter criteria.
- Loop until the number of ratios is reached or `Dem_GetNextIUMPRRatioDataAndDTC()` returns E_NOT_OK
  - For each ratio, call `Dem_GetNextIUMPRRatioDataAndDTC()` to get the Numerator and the Denominator values. To identify the ratio, the call also returns the OBD DTC and the UDS DTC.

Note

The returned ratios are neither sorted nor can a specific ratio be accessed directly.

If this is needed to fulfill OEM specific requirements, the implementation of the diagnostic service, which is used to read IUMPR data, must provide an according buffering or sorting mechanism.

2.3.8 DTR Integration

If all DTR Ids related to an OBD MID are unavailable, the OBD MID itself can become unavailable during runtime (refer to chapter 2.2.12.4). To save runtime on the retrieval of the supported OBD MIDs the API `Dem_UpdateAvailableOBDMIDs()` can be called in advance, after

- a DTR Id was set invisible **or**
- an event assigned to a DTR Id was set unavailable **or**
- the DTC of an event assigned to a DTR Id was suppressed.

Otherwise availability of the supported OBD MIDs will automatically be updated when the Dcm requests them.

2.4 Configuration

This chapter shall give an overview of the most important configuration parameters and their values according to OBD. It makes no claim to be complete, and some settings may differ if the Dem shall be compliant to other OBD standards (e.g. EOBD).

In any case, the customer has to take the related OEM and OBD requirements into account.

2.4.1 DemGeneral

Dem/DemGeneral		
Configuration Parameter	Legislation	Setting
DemIgnitionCycleFunctionIdRef	OBDII OBD on UDS WWH-OBD	the FID that is disabled when the input signals to calculate the Ignition Cycle and Hybrid Ignition Cycle are defective (only relevant if IUMPR needs to be supported)
DemImmediateNvStorageSupport	OBDII OBD on UDS WWH-OBD	**True**
DemMILIndicatorRef	OBDII OBD on UDS WWH-OBD	the respective indicator (refer to chapter 2.4.1.1) that shall represent the MIL
DemMaxNumberEventEntryOBDFreezeFrame	OBDII OBD on UDS	at least **1**, should be at least **5** in case of OBDonUDS
DemMaxNumberEventEntryPermanent	OBDII OBD on UDS WWH-OBD	**4**
DemMultipleTripSupport	OBDII OBD on UDS WWH-OBD	**True**
DemOBDSupport	OBDII OBD on UDS WWH-OBD	**PRIMARY_ECU**
DemOperationCycleStatusStorage	OBDII OBD on UDS WWH-OBD	**True**
DemUseNvm	OBDII OBD on UDS WWH-OBD	**True**
DemResetTestFailedOnOperationCycleStart	WWH-OBD	**False**

Table 2-35 DemGeneral configuration

2.4.1.1 Event Memory Storage Trigger

Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory		
Configuration Parameter	Legislation	Setting
DemEventMemoryEntryStorageTrigger	OBDII OBD on UDS WWH-OBD	**TEST_FAILED**

Table 2-36 Event Memory Storage Trigger configuration

2.4.1.2 Malfunction Indicator Lamp

Dem/DemGeneral/DemIndicator		
Configuration Parameter	Legislation	Setting
-	OBDII OBD on UDS WWH-OBD	create an indicator element which can be marked as MIL indicator by parameter, refer to Table 2-35

Table 2-37 MIL indicator configuration

2.4.1.3 PID Configuration

Dem/DemGeneral/DemPidConfiguration		
Configuration Parameter	Legislation	Setting
DemSupportPid01	OBDII OBD on UDS WWH-OBD	**True**
DemSupportPid1C	OBDII OBD on UDS WWH-OBD	Project specific
DemSupportPid21	OBDII OBD on UDS WWH-OBD	Project specific
DemSupportPid30	OBDII OBD on UDS WWH-OBD	Project specific
DemSupportPid31	OBDII OBD on UDS WWH-OBD	Project specific
DemSupportPid41	OBDII OBD on UDS WWH-OBD	**True**
DemSupportPid4D	OBDII OBD on UDS WWH-OBD	Project specific
DemSupportPid4E	OBDII OBD on UDS WWH-OBD	Project specific
DemSupportPid91	WWH-OBD	**True**

Table 2-38 DemPidConfiguration configuration

2.4.1.4 Driving Cycle

Dem/DemGeneral/DemOperationCycle		
Configuration Parameter	Legislation	Setting
DemOperationCycleType	OBDII OBD on UDS WWH-OBD	**DEM_OPCYC_OBD_DCY**

Table 2-39 Driving Cycle configuration

2.4.1.5 Warmup Cycle

	Dem/DemGeneral/DemOperationCycle	
Configuration Parameter	Legislation	Setting
DemOperationCycleType	OBDII OBD on UDS WWH-OBD	**DEM_OPCYC_WARMUP**

Table 2-40 Warmup Cycle configuration

2.4.1.6 Additional Ignition Cycle for Hybrids

Only needed if your ECU is used in a hybrid car and if IUMPR is supported.

Dem/DemGeneral/DemOperationCycle		
Configuration Parameter	Legislation	Setting
DemOperationCycleType	OBDII OBD on UDS WWH-OBD	**DEM_OPCYC_IGNITION_HYBRID**

Table 2-41 Hybrid Ignition Cycle configuration

2.4.1.7 Data Element for Distance Information

Dem/DemGeneral/DemDataClass		
Configuration Parameter	Legislation	Setting
DemDataElementDataSize	OBDII OBD on UDS WWH-OBD	**4**
DemDataElementUsePort		**USE_DATA_CLIENT_SERVER_PORT or USE_DATA_FUNCTION_CALL**
DemDataElementDataType		**UINT8_N**

Table 2-42 Distance information data element configuration

2.4.1.8 Data Elements for Time Since Engine Start Minutes or Seconds

Dem/DemGeneral/DemDataClass		
Configuration Parameter	Legislation	Setting
DemDataElementDataSize	OBDII OBD on UDS WWH-OBD	**4**
DemDataElementUsePort		**USE_DATA_CLIENT_SERVER_PORT or USE_DATA_FUNCTION_CALL**
DemDataElementDataType		**UINT8_N**

Table 2-43 Time since engine start minutes or seconds data element configuration

2.4.1.9 General OBD

Dem/DemGeneral/DemGeneralOBD		
Configuration Parameter	Legislation	Setting
DemClearPermanentDtcBehavior	OBDII OBD on UDS	Project specific
DemDelayWarmUpCycleToDcyEnd	OBDII OBD on UDS WWH-OBD	Recommended setting to TRUE to satisfy requirements in [8]
DemEngineRuntimePolling	OBDII OBD on UDS WWH-OBD	Project specific
DemEngineRuntimeSecondsPolling	OBDII	Project specific
DemOBDFreezeFrameBehavior	OBDII WWH-OBD	Project specific

Dem/DemGeneral/DemGeneralOBD		
Configuration Parameter	Legislation	Setting
DemOBDFreezeFrameInService19	OBDII	Project specific
DemOBDHideOccurrences	OBDII OBD on UDS	Project specific
DemOBDInputDistanceInformation	OBDII OBD on UDS WWH-OBD	the data element created for the distance information, refer to chapter 2.4.1.7
DemOBDLegislation	OBDII OBD on UDS WWH-OBD	Project specific
DemOBDPermanentDtcActivationDistance	OBDII OBD on UDS	Project specific
DemOBDPermanentMandatoryPfcAfterClear	OBDII OBD on UDS	Project specific
DemOBDPermanentDtcImmediateReport	OBDII OBD on UDS	Project specific
DemOBDPermanentEraseConditionDcy	OBDII OBD on UDS	Project specific
DemOBDPermanentForWWHOBD	WWH-OBD	Project specific
DemOBDRestartDcyOnClearDTC	OBDII OBD on UDS WWH-OBD	Project specific
DemOBDTimeSinceEngineStart	OBDII OBD on UDS WWH-OBD	The data element created for the time since engine start minutes, refer to chapter 2.4.1.8.
DemOBDTimeSinceEngineStartSeconds	OBDII	The data element created for the time since engine start seconds, refer to chapter 2.4.1.8.
DemOBDSupportMajorMonitors	OBDII OBD on UDS WWH-OBD	Project specific
DemOBDPID41ComprehensiveComponentsAlwaysComplete	OBDII OBD on UDS WWH-OBD	Project specific
DemOBDPIDF501ComprehensiveComponentsAlwaysComplete	OBD on UDS	Project specific
DemOBDSimilarConditionsSupport	OBDII OBD on UDS WWH-OBD	Project specific
DemReportBlinkingMilAsActive	OBDII OBD on UDS WWH-OBD	Project specific
DemOBDPermanentChronologyMaxNumberEvents	OBDII OBD on UDS WWH-OBD	Project specific
DemSupportedObdUdsDtcSeparation	OBD on UDS	Project specific

Dem/DemGeneral/DemGeneralOBD		
Configuration Parameter	Legislation	Setting
DemDisableEventMonitoringInCurrentDcy	OBDII OBD on UDS WWH-OBD	Project specific

Table 2-44 DemGeneralOBD configuration

2.4.2 DemConfigSet

2.4.2.1 DTC

Dem/DemConfigSet/DemDTCClass		
Configuration Parameter	Legislation	Setting
DemFunctionalGroupIdentifier	OBDII OBD on UDS WWH-OBD	**0x33** (for emissions-system group). Setting this value determines if a DTC is OBD relevant. Manual setting is required for OBDonUDS only. See also parameters DemObdDTC and DemOBDFreezeFrameClassRef.
DemDTCAttributesRef	OBDII OBD on UDS WWH-OBD	If DTC is OBD Relevant (DemFunctionalGroupIdentifier = 0x33), this must refer to a DemDTCAttributes container having Primary Memory as its Memory Destination Reference.
DemImmediateNvStorage	OBD on UDS WWH-OBD	**True**
DemObdDTC	OBDII	The respective DTC number must be added here. Setting a DTC number here automatically sets the parameter DemFunctionalGroupIdentifier to 0x33.
DemObdDTC3Byte	OBD on UDS	If DemSupportedObdUdsDtcSeparation is enabled, this DTC number must be configured for all OBD relevant DTCs.
DemWWHOBDDTCClass	WWH-OBD	Depends on the monitor functionality.
DemOBDONUDSDTCClass	OBD on UDS	Depends on the monitor functionality.

Table 2-45 DTC configuration

2.4.2.2 Event

Dem/DemConfigSet/DemEventParameter/DemEventClass		
Configuration Parameter	Legislation	Setting
DemAgingCycleCounterThreshold	OBDII OBD on UDS WWH-OBD	**40**

Dem/DemConfigSet/DemEventParameter/DemEventClass		
Configuration Parameter	Legislation	Setting
DemAgingCycleRef	OBDII OBD on UDS WWH-OBD	The respective operation cycle that is configured as warmup cycle (refer to chapter 2.4.1.5)
DemEventFailureCycleRef	OBDII OBD on UDS WWH-OBD	The respective operation cycle that is configured as DCY (refer to chapter 2.4.1.4)
DemEventFailureCycleCounterThreshold	OBDII OBD on UDS WWH-OBD	**1** for US and **2** for EU
DemOBDVisibilityDelayedUntilDcyQualification	OBDII OBD on UDS WWH-OBD	Defines when the CDTC and the WIR status bits of an OBD relevant event become externally visible.
DemEventPriority	OBDII OBD on UDS WWH-OBD	greater (smaller number) than any UDS only DTCs For WWH-OBD refer to Table 2-9
DemOperationCycleRef	OBDII OBD on UDS WWH-OBD	The respective operation cycle that is configured as DCY (refer to chapter 2.4.1.4)
DemOBDFreezeFrameClassRef	OBD on UDS WWH-OBD	Reference to the legislative freeze frame that shall be stored for the event. For OBD legislation WWH-OBD referencing a legislative freeze frame automatically sets the DemFunctionalGroupIdentifier of the DTC referenced by the event to 0x33.
DemEventOBDReadinessGroup	OBDII OBD on UDS WWH-OBD	Project specific

Table 2-46 Event configuration

Dem/DemConfigSet/DemEventParameter/DemEventClass/DemIndicatorAttribute		
Configuration Parameter	Legislation	Setting
DemIndicatorHealingCycleCounterThreshold	OBDII OBD on UDS WWH-OBD	**3**
DemIndicatorHealingCycleRef	OBDII OBD on UDS WWH-OBD	The respective operation cycle that is configured as DCY (refer to chapter 2.4.1.4)
DemIndicatorRef	OBDII OBD on UDS WWH-OBD	the respective indicator (refer to chapter 2.4.1.1) that shall represent the MIL

Table 2-47 Event indicator configuration

2.4.2.3 Legislative Freeze Frame

Dem/DemConfigSet/DemPidClass		
Configuration Parameter	Legislation	Setting
	OBDII	OEM specific freeze frame content. Reference the required data elements here.

Table 2-48 Legislative freeze frame configuration OBDII

WWH-OBD OBD on UDS

Mind that the legislative Freeze frame is defined per event in OBDonUDS and WWH-OBD legislation, as defined in Table 2-46**.**

2.4.2.4 DTR

Dem/DemConfigSet/DemDtrs/DemDtr		
Configuration Parameter	Legislation	Setting
DemDtrCompuDenominator0	OBDII OBD on UDS WWH-OBD	Project specific
DemDtrCompuNumerator0	OBDII OBD on UDS WWH-OBD	Project specific
DemDtrCompuNumerator1	OBDII OBD on UDS WWH-OBD	Project specific
DemDtrEventRef	OBDII OBD on UDS WWH-OBD	Event associated to Dtr
DemDtrMid	OBDII OBD on UDS WWH-OBD	Project specific (values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0 and 0xE0 are reserved)
DemDtrTid	OBDII OBD on UDS WWH-OBD	Project specific
DemDtrUasid	OBDII OBD on UDS WWH-OBD	Project specific
DemDtrUpdateKind	OBDII OBD on UDS WWH-OBD	Project specific

Table 2-49 DTR configuration

2.4.3 Post-Build Loadable

Using the feature Post-Build Loadable as described in [1] the following use cases (among others) typically used in OBD projects can be covered:

Legislation	Use Case
OBDII OBD on UDS WWH-OBD	**OBD Fault Relevance** The OBD relevance of a fault can be controlled by attaching or detaching the MIL indicator to/ from the event. **OBD II**: Additionally, the 2 byte OBD II DTC must be added or removed. If no MIL

Legislation	Use Case
	indicator and OBD II DTC is available the event cannot be stored in permanent memory, does not store an OBD Freeze Frame and is not available in the OBD Modes (e.g. Mode $03 or Mode $07). **OBDonUDS**: Additionally, the DemFunctionalGroupIdentifier must be set to 0x33. **WWH-OBD**: Additionally, the DemWWHOBDDTCClass must be set to a valid DTC Class and DemOBDFreezeFrameClassRef must be set.
OBDII OBD on UDS WWH-OBD	**MIL Group Affiliation** (refer to 2.2.3.3) An event can be attached to a MIL Group from a pool of pre-compile configured groups.
OBDII	**Combining OBD DTCs** (refer to 2.2.3.2) Multiple fault paths can be combined to one single OBD DTC by providing the same DTC number.
OBDII OBD on UDS WWH-OBD	**1-Trip/ 2-Trip/ 3-Trip Fault** The DTC specific trip count can be configured according the regulations of the related market, e.g. 2-trip in the US and 3-trip in the EU. To provide immediate attention to a fault the failure cycle counter threshold can be deleted.

Table 2-50 Post-Build Loadable use cases

2.4.4 Post-Build Selectable

It is possible to use different configuration sets with legislation OBD-II, OBDonUDS, WWH-OBD or without OBD support in parallel. This allows for an ECU to adapt to the applying OBD legislation on start-up by choosing the corresponding configuration variant.

In a Post-Build Selectable variant where no OBD functionality is needed, ‘OBD support’ must be disabled.

In a Post-Build Selectable variant where OBD functionality is needed, ‘OBS support’ must be enabled and the required legislation must be configured.

In a Post-Build Selectable variant with legislation WWH-OBD the customer can decide, if permanent storage shall be enabled for DTCs. It is possible to have one Post-Build Selectable variant with permanent storage enabled and one variant with permanent storage disabled in parallel.

If J1939 is licensed, Post-Build Selectable variants with legislation OBD-II and OBDonUDS or without OBD support can be used in parallel.

For information which additional OBD relevant configuration parameters support Post-Build Selectable, have a look at the basic software module description (bswmd) file accompanying your delivery.

3 Zero Emission Vehicle OBD

3.1 Introduction

Zero Emission Vehicle OBD (ZEVonUDS) is the regulated Advanced Clean Cars II that demands a certain set of diagnostic services implemented for all zero emission vehicles. The legislation is demanded by CARB see [15] and the diagnostic communication is standardized by SAE J1979-3 see [14] which defines the diagnostic services for electric, fuel cell, and other zero emission propulsion systems. ZEVonUDS only supports OBD functionality described in this chapter.

3.2 Functional Description

ZEVonUDS contains a reduced function set of OBDonUDS (SAE J1979-2). The following Dem relevant OBD functionality is supported for ZEVonUDS:

3.2.1 Functional Group Identifier (FGID) 0x33

With ZEVonUDS supported DTCs can be configured as ZEV propulsion related by setting their FGID to 0x33.

3.2.2 Service 0x19 42 Request ZEV-Related DTCs with Status Mask

To retrieve the ZEV related DTCs matching the DTC status mask, the DTC severity mask and the DTC status mask a filter must be applied with DTCClass_1 (ZEV Propulsion related DTC) according to SAE J1979-3.

3.2.3 Service 0x14 Clear DTC

Providing correct PID values it is defined by legislation that only clear all DTCs is allowed.

For enhanced diagnostic testers the UDS service to clear all DTCs ($14 0xFFFFFF) can be used to achieve this requirement.

Clearing all ZEV related DTCs ($14 0xFFFF33) has the same effect as clearing all DTCs ($14 0xFFFFFF).

Nevertheless, during production, it is necessary to clear a single DTC or a group of DTCs. Therefore, in addition to the UDS data the Dem will clear the following OBD related elements not only during clear all DTCs but also for a clear single DTC:

- **PID $21** and **PID $4D** will be reset to 0 if the DTC to be cleared is the last confirmed DTC.

3.2.4 PIDs calculated by Dem

ZEV OBD PIDs are supported for readout via service 0x22 ReadDataByIdentifier.

Not all PIDs are provided by any ECU, and only a few PIDs are calculated by the Dem. For a complete description of all PIDs, please refer to [14].

3.2.4.1 PID $1C: OBD requirements to which vehicle is designed

The value of this PID is statically provided by configuration. It specifies the regulated ZEV market (e.g. Heavy Duty ZEV…).

- Related API function: Dem_DcmReadDataOfPID1C()

3.2.4.2 PID $21: Distance travelled while any ZEV propulsion related DTC is confirmed

PID $21 is a 2 byte value with a resolution of 1 km / Bit. It is accumulated if at least one ZEV propulsion related DTC is confirmed.

If the counter reaches its maximum of 65535 km, this value is kept until the next reset, no wrap-around occurs.

The value of PID $21 is reset on clear request or when all ZEV propulsion related DTCs are no longer confirmed.

- Related API function: Dem_DcmReadDataOfPID21()

Caution
As stated in see [16] , the PID$21 value shall be frozen when source of vehicle speed is not available. DEM computes the PID$21 value based on the odometer value provided by the application. Thus, the application should freeze the reported odometer value when the source of vehicle speed is not available.

3.2.4.3 PID $31: Distance since DTCs cleared

This PID is a 2 byte value that contains the travelled distance in km since the service 0x14 request. If the maximum value of 65535 km is reached, no wrap-around to zero occurs. The maximum value is kept until the next reset.

- Related API function: Dem_DcmReadDataOfPID31()

Caution
As stated in see [16] , the PID$31 value shall be frozen when source of vehicle speed is not available. DEM computes the PID$31 value based on the odometer value provided by the application. Thus, the application should freeze the reported odometer value when the source of vehicle speed is not available.

3.2.4.4 PID $4D: Engine Runtime while any ZEV propulsion related DTC is confirmed

PID $4D is a 2 byte value with a resolution of 1 min / Bit and behaves analogous to PID $21. It is accumulated while any ZEV propulsion related DTC is confirmed. If the counter reaches its maximum of 65535 minutes, this value is kept until the next reset, no wrap-around occurs.

The value of PID $4D is reset on clear request or when all ZEV propulsion related DTCs are no longer confirmed.

- Related API function: Dem_DcmReadDataOfPID4D()

3.2.4.5 PID $4E: Engine run time since DTCs cleared

This PID is a 2 byte value that behaves analogous to PID $31. It contains the engine run time in minutes since the last service 0x14 request. If the maximum value of 65535 minutes is reached, no wrap-around to zero occurs. The maximum value is kept until the next reset.

- Related API function: Dem_DcmReadDataOfPID4E()

3.3 Integration

3.3.1 Operation Cycle Handling

For ZEV propulsion relevant events the operation cycle handling is the same as the behavior specified by UDS.

3.3.2 Required Data

The following data is needed by the Dem for ZEV OBD functionality and must be provided to the Dem during integration.

Data	Source	Description
Distance Information	Application	This 4 byte value with the travelled distance is needed to calculate PID $21 (3.2.4.2) and PID $31 (3.2.4.3). The callback used by the Dem to request the distance information is configured with the parameter "OBD Input Distance Information". If no valid value can be provided, the callback implementation must return E_NOT_OK. The Dem will continue to trigger this callback until E_OK will be returned. The returned value must have a resolution of 1 kilometer or 1 mile per bit.
Engine Run Time Minutes	Application	This 4 byte value with the time since engine start in minutes is needed to calculate PID $4D (3.2.4.4) and PID $4E (3.2.4.5). The callback used by the Dem to request the engine run time in minutes is configured with the parameter "Time Since Engine Start Minutes". If no valid value can be provided, the callback implementation must return E_NOT_OK. The Dem will continue to trigger this callback until E_OK will be returned. The returned value must have a resolution of 1 minute per bit and it must be provided in big endian byte order.

Table 3-1 Required data for ZEVonUDS

3.3.3 Development Error Codes

If the Dem is used for any OBD legislation, additional service IDs are defined for development error reporting to DET, see 2.3.4

3.4 Configuration

This chapter shall give an overview of the most important configuration parameters and their values according to ZEV OBD. It makes no claim to be complete, and some settings may differ if the Dem shall be compliant to other OBD standards.

In any case, the customer must take the related OEM and OBD requirements into account.

3.4.1 DemGeneral

Dem/DemGeneral	
Configuration Parameter	Setting
DemImmediateNvStorageSupport	**True**

Dem/DemGeneral	
Configuration Parameter	Setting
DemOBDSupport	**PRIMARY_ECU**
DemUseNvm	**True**

Table 3-2 DemGeneral configuration

3.4.2 PID Configuration

Dem/DemGeneral/DemPidConfiguration	
Configuration Parameter	Setting
DemSupportPid1C	**Project specific**
DemSupportPid21	**Project specific**
DemSupportPid31	**Project specific**
DemSupportPid4D	**Project specific**
DemSupportPid4E	**Project specific**

Table 3-2 DemPidConfiguration configuration

3.4.3 Data Element for Distance Information

Dem/DemGeneral/DemDataClass	
Configuration Parameter	Setting
DemDataElementDataSize	**4**
DemDataElementUsePort	**USE_DATA_CLIENT_SERVER_PORT or USE_DATA_FUNCTION_CALL**
DemDataElementDataType	**UINT8_N**

Table 3-3 Distance information data element configuration

3.4.4 Data Elements for Time Since Engine Start Minutes

Dem/DemGeneral/DemDataClass	
Configuration Parameter	Setting
DemDataElementDataSize	**4**
DemDataElementUsePort	**USE_DATA_CLIENT_SERVER_PORT or USE_DATA_FUNCTION_CALL**
DemDataElementDataType	**UINT8_N**

Table 3-4 Time since engine start minutes or seconds data element configuration

3.4.5 General OBD

Dem/DemGeneral/DemGeneralOBD	
Configuration Parameter	Setting
DemEngineRuntimePolling	**Project specific**
DemOBDInputDistanceInformation	the data element created for the distance information, refer to chapter 3.4.3.
DemOBDLegislation	**ZEVonUDS**
DemOBDTimeSinceEngineStart	The data element created for the time since engine start minutes, refer to chapter 3.4.4.

Table 3-5 DemGeneralOBD configuration

3.4.6 DemConfigSet

3.4.6.1 DTC

Dem/DemConfigSet/DemDTCClass	
Configuration Parameter	Setting
DemFunctionalGroupIdentifier	**0x33 (for ZEV-system group). Setting this value determines if a DTC is OBD relevant. Manual setting is required for ZEVonUDS.**
DemDTCAttributesRef	**If DTC is OBD Relevant (DemFunctionalGroupIdentifier = 0x33), this must refer to a DemDTCAttributes container having Primary Memory as its Memory Destination Reference**.
DemImmediateNvStorage	**True**
DemOBDONUDSDTCClass	**DTC Class 1**

Table 3-6 DTC configuration

3.5 Incompatible Features

3.5.1 ZEVonUDS and J1939

Features ZEVonUDS and J1939 are not allowed to be enabled at the same time.

4 Interfaces

The Dem provides and requires additional APIs to meet the requirements of the OBD use case. These interfaces are listed in [1] in the chapter “Not supported APIs” because they are not available if the Dem is used without OBD.

4.1 Provided Interfaces

4.1.1 DCM

4.1.1.1 Dem_DcmReadDataOfPID01()

Prototype	
`Std_ReturnType` **`Dem_DcmReadDataOfPID01`** `( uint8* PID01value )`	
Parameter	
`PID01value`	Buffer with size of four bytes containing the contents of PID $01 computed by the Dem.
Return code	
`Std_ReturnType`	E_OK: is always returned with disabled DET E_NOT_OK: is returned with enabled DET when an error is detected
Functional Description	
Provides the monitoring status since DTCs have been cleared. For a detailed description of the status bits, refer to [6].	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-1 Dem_DcmReadDataOfPID01()

4.1.1.2 Dem_DcmReadDataOfPID1C()

Prototype	
`Std_ReturnType` **`Dem_DcmReadDataOfPID1C`** `( uint8* PID1Cvalue )`	
Parameter	
`PID1Cvalue`	Buffer containing the contents of PID $1C computed by the Dem.
Return code	
`Std_ReturnType`	E_OK: is always returned with disabled DET E_NOT_OK: is returned with enabled DET when an error is detected
Functional Description	
Provides the OBD requirements to which the vehicle or engine is certified. 0x12…0xFA reserved 0xFB…0xFF reserved (SAE J1939)	

Particularities and Limitations
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition

Table 4-2 Dem_DcmReadDataOfPID1C()

4.1.1.3 Dem_DcmReadDataOfPID21()

Prototype	
Std_ReturnType **Dem_DcmReadDataOfPID21** (uint8* PID21value)	
Parameter	
PID21value	Buffer with size of two bytes containing the contents of PID $21 computed by the Dem.
Return code	
Std_ReturnType	E_OK: is always returned with disabled DET E_NOT_OK: is returned with enabled DET when an error is detected
Functional Description	
Provides the distance traveled while the MIL indicator is active	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-3 Dem_DcmReadDataOfPID21()

4.1.1.4 Dem_DcmReadDataOfPID30()

Prototype	
Std_ReturnType **Dem_DcmReadDataOfPID30** (uint8* PID30value)	
Parameter	
PID30value	Buffer containing the contents of PID $30 computed by the Dem.
Return code	
Std_ReturnType	E_OK: is always returned with disabled DET E_NOT_OK: is returned with enabled DET when an error is detected
Functional Description	
Provides the number of warm up cycles since DTCs cleared.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-4 Dem_DcmReadDataOfPID30()

4.1.1.5 Dem_DcmReadDataOfPID31()

Prototype	
Std_ReturnType **Dem_DcmReadDataOfPID31** (uint8* PID31value)	
Parameter	
PID31value	Buffer with size of two bytes containing the contents of PID $31 computed by the Dem.
Return code	
Std_ReturnType	E_OK: is always returned with disabled DET E_NOT_OK: is returned with enabled DET when an error is detected
Functional Description	
Provides the distance traveled since DTCs cleared.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-5 Dem_DcmReadDataOfPID31()

4.1.1.6 Dem_DcmReadDataOfPID41()

Prototype	
Std_ReturnType **Dem_DcmReadDataOfPID41** (uint8* PID41value)	
Parameter	
PID41value	Buffer with size of four bytes containing the contents of PID $41 computed by the Dem.
Return code	
Std_ReturnType	E_OK: is always returned with disabled DET E_NOT_OK: is returned with enabled DET when an error is detected
Functional Description	
Provides the monitoring status for this driving cycle. For a detailed description of the status bits, refer to [6].	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-6 Dem_DcmReadDataOfPID41()

4.1.1.7 Dem_DcmReadDataOfPID4D()

Prototype	
Std_ReturnType **Dem_DcmReadDataOfPID4D** (uint8* PID4Dvalue)	

Parameter	
`PID4Dvalue`	Buffer with size of two bytes containing the contents of PID $4D computed by the Dem.
Return code	
`Std_ReturnType`	E_OK: is always returned with disabled DET E_NOT_OK: is returned with enabled DET when an error is detected
Functional Description	
Provides the engine run time while the MIL indicator is activated.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-7 Dem_DcmReadDataOfPID4D()

4.1.1.8 Dem_DcmReadDataOfPID4E()

Prototype	
`Std_ReturnType` **`Dem_DcmReadDataOfPID4E`** `( uint8* PID4Evalue )`	
Parameter	
`PID4Evalue`	Buffer with size of two bytes containing the contents of PID $4E computed by the Dem.
Return code	
`Std_ReturnType`	E_OK: is always returned with disabled DET E_NOT_OK: is returned with enabled DET when an error is detected
Functional Description	
Provides the engine run time since DTCs cleared.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-8 Dem_DcmReadDataOfPID4E()

4.1.1.9 Dem_DcmReadDataOfPID91()

WWH-OBD

This chapter is only relevant for **WWH-OBD**.

Prototype	
`Std_ReturnType` **`Dem_DcmReadDataOfPID91`** `( uint8* PID91value )`	

Parameter	
`PID91value`	Buffer with size of five bytes containing the contents of PID $91 computed by the Dem.
Return code	
`Std_ReturnType`	E_OK: is always returned with disabled DET E_NOT_OK: is returned with enabled DET when an error is detected
Functional Description	
Provides the ECU OBD system information.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-9 Dem_DcmReadDataOfPID91()

4.1.1.10 Dem_DcmReadDataOfPIDF501()

This chapter is only relevant for **OBDonUDS**.

Prototype	
`Std_ReturnType` **`Dem_DcmReadDataOfPIDF501`** `( uint8* PIDF501value )`	
Parameter	
`PIDF501value`	Buffer with size of six bytes containing the contents of PID $F501 computed by the Dem.
Return code	
`Std_ReturnType`	E_OK: is always returned with disabled DET E_NOT_OK: is returned with enabled DET when an error is detected
Functional Description	
Provides the monitoring status since DTCs have been cleared. For a detailed description of the status bits, refer to [6].	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-10 Dem_DcmReadDataOfPIDF501()

4.1.1.11 Dem_GetDTCSeverityAvailabilityMask()

This chapter is only relevant for **WWH-OBD** and **OBDonUDS**.

Prototype	
`Std_ReturnType` **`Dem_GetDTCSeverityAvailabilityMask`** `( uint8 ClientId, Dem_DTCSeverityType* DTCSeverityMask )`	
Parameter	
`ClientId`	Unique client id, assigned to the instance of the calling module
`DTCSeverityMask`	Receives the supported severity bits from the Dem. All supported information is indicated by setting the corresponding severity bit to 1.
Return code	
`Std_ReturnType`	E_OK: is always returned with disabled DET E_NOT_OK: is returned with enabled DET when an error is detected
Functional Description	
Gets the DTC severity availability mask.	
Particularities and Limitations	
> This function is reentrant for different ClientIds. > This function is synchronous. > This function is only callable from the master partition	

Table 4-11 Dem_GetDTCSeverityAvailabilityMask()

4.1.1.12 Dem_DcmReadDataOfOBDFreezeFrame()

This chapter is only relevant for **OBD II**.

Prototype	
`Std_ReturnType` **`Dem_DcmReadDataOfOBDFreezeFrame`** `( uint8 PID, uint8 DataElementIndexOfPID, uint8* DestBuffer, uint16* BufSize )`	
Parameter	
`PID`	This parameter is an identifier for a PID as defined in ISO15031-5.
`DataElementIndexOfPID`	Data element index of this PID according to the Dcm configuration of service $02.
`DestBuffer`	Points to the buffer to which the data element of the PID shall be written to.

`BufSize`	When the function is called this parameter contains the maximum number of data bytes that can be written to the buffer. The function returns the actual number of written data bytes in this parameter.
Return code	
`Std_ReturnType`	E_OK: freeze frame data was successfully reported E_NOT_OK: freeze frame data was not successfully reported
Functional Description	
Gets a 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. When the requested data is currently not accessible due to an ongoing data update of the OBD Freeze frame memory block, the function returns E_NOT_OK.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-12 Dem_DcmReadDataOfOBDFreezeFrame()

4.1.1.13 Dem_DcmGetDTCOfOBDFreezeFrame()

OBDII This chapter is only relevant for **OBD II**.

Prototype	
`Std_ReturnType` **`Dem_DcmGetDTCOfOBDFreezeFrame`** `( uint8 FrameNumber, uint32* DTC, Dem_DTCFormatType DTCFormat )`	
Parameter	
`FrameNumber`	Unique identifier for a freeze frame record as defined in ISO 15031-5. The value 0x00 indicates the complete OBD freeze frame. Other values are reserved for future functionality.
`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.
`DTCFormat`	Defines the output-format of the requested DTC value. DEM_DTC_FORMAT_UDS: output format shall be UDS DEM_DTC_FORMAT_OBD: output format shall be OBD DEM_DTC_FORMAT_J1939: output format shall be J1939 DEM_DTC_FORMAT_OBD_3BYTE: not allowed
Return code	
`Std_ReturnType`	E_OK: operation was successful E_NOT_OK: no DTC available
Functional Description	
Provides the DTC by freeze frame record number.	

Particularities and Limitations
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition

Table 4-13 Dem_DcmGetDTCOfOBDFreezeFrame()

4.1.1.14 Dem_DcmGetOBDFreezeFrameData()

OBDII This chapter is only relevant for **OBD II**.

Prototype	
`Std_ReturnType` **`Dem_DcmGetOBDFreezeFrameData`** `( uint32* DTC, uint8* DestBuffer, uint16* BufSize )`	
Parameter	
`DTC`	Receives the DTC value in UDS format returned by this function. If the return value of the function is other than DEM_GET_FFBYRECORD_OK this parameter does not contain valid data.
`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: {NumOfPIDs, 0xF4PID[1], data[1], …, 0xF4PID[N], data[N]}.
`BufSize`	When the function is called this parameter contains the maximum number of data bytes that can be written to the buffer. The function returns the actual number of written data bytes in this parameter.
Return code	
`Std_ReturnType`	E_OK: DTC and OBD freeze frame data successfully reported E_NOT_OK: No DTC and OBD freeze frame data available
Functional Description	
Provides the DTC and its associated OBD freeze frame record via UDS protocol. The function stores the data in the provided DestBuffer. When the requested data is currently not accessible due to an ongoing data update of the OBD Freeze frame memory block, the function returns E_NOT_OK.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is a deprecated AUTOSAR API. Due to the restrictions of the Autosar design, it only works correctly if the Dem provides the full data of each PID. > This function is only callable from the master partition	

Table 4-14 Dem_DcmGetOBDFreezeFrameData()

4.1.1.15 Dem_DcmGetAvailableOBDMIDs()

Prototype	
Std_ReturnType **Dem_DcmGetAvailableOBDMIDs** (uint8 Obdmid, uint32* Obdmidvalue)	
Parameter	
Obdmid	Availability OBDMID ($00, $20, $40, …).
Obdmidvalue	Bit coded information on the support of OBDMIDs in the respective availability OBDMID range.
Return code	
Std_ReturnType	E_OK: The bit mask was reported successfully. E_NOT_OK: The requested availability OBDMID is not supported.
Functional Description	
Provides the supported MIDs in the requested range.	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-15 Dem_DcmGetAvailableOBDMIDs()

4.1.1.16 Dem_DcmGetNumTIDsOfOBDMID()

Prototype	
Std_ReturnType **Dem_DcmGetNumTIDsOfOBDMID** (uint8 Obdmid, uint8* numberOfTIDs)	
Parameter	
Obdmid	OBDMID for which the number of assigned TIDs is requested
numberOfTIDs	Number of TIDs for the requested OBDMID.
Return code	
Std_ReturnType	E_OK: The number of TIDs was reported successfully. E_NOT_OK: The requested OBDMID is not supported.
Functional Description	
Provides the number of TIDs per OBDMID.	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-16 Dem_DcmGetNumTIDsOfOBDMID()

4.1.1.17 Dem_DcmGetDTRData()

Prototype	
Std_ReturnType **Dem_DcmGetDTRData** (uint8 Obdmid, uint8 TIDindex, uint8* TIDvalue, uint8* UaSID, uint16* Testvalue, uint16* Lowlimvalue, uint16* Upplimvalue)	
Parameter	
Obdmid	OBDMID by which the DTR Id can be identified in combination with the TIDindex.
TIDindex	Index of the TID within the DEM. Runs from 0 to “numberOfTIDs” obtained in the call to Dem_DcmGetNumTIDsOfOBDMID().
TIDvalue	Value of the TID corresponding to the TIDindex.
UaSID	UnitandScalingID for the external representation of the data.
Testvalue	Latest test result for the requested OBDMID / TID combination.
Lowlimvalue	Lower limit value associated to the latest test result.
Upplimvalue	Upper limit value associated to the latest test result.
Return code	
Std_ReturnType	E_OK: Requested DTR data was reported successfully. E_NOT_OK: The requested OBDMID or the TIDindex for the OBDMID is not supported.
Functional Description	
Provides the DTR data for an OBDMID / TIDindex combination.	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-17 Dem_DcmGetDTRData()

4.1.1.18 Dem_SetDTCFilterByReadinessGroup()

Prototype	
Std_ReturnType **Dem_SetDTCFilterByReadinessGroup** (uint8 ClientId, Dem_EventOBDReadinessGroupType ReadinessGroupNumber, Dem_DTCFormatType DTCFormat)	
Parameter	
ClientId	Unique client id, assigned to the instance of the calling module.

`ReadinessGroupNumber`	Unique readiness group identifier > 0x00: No Readiness Group > 0x01: Catalyst Monitoring Group > 0x02: Heated Catalyst Monitoring Group > 0x03: Misfire Monitoring Group > 0x04: Evaporative Monitoring Group > 0x05: Secondary Air Monitoring Group > 0x06: Fuel Systems Monitoring Group > 0x07: Exhaust Gas Sensor Monitoring Group > 0x08: EGR Systems Monitoring Group > 0x09: Positive/Crankcase Ventilation (PCV) System Monitoring Group > 0x0A: Engine Cooling System Monitoring Group > 0x0B: Cold Start Emission Reduction Strategy(CSER) System Monitoring Group > 0x0C: Variable Valve Timing, Lift, and/or Control (VVT) System Monitoring Group > 0x0D: Direct Ozone Reduction (DOR) System Monitoring Group > 0x0E: Comprehensive Components Monitoring Group > 0x0F: Other Emission Control or Source System Monitoring Group > 0x10: NMHC Catalyst Monitoring Group > 0x11: NOx Aftertreatment Systems Monitoring Group > 0x12: Boost Pressure System Monitoring Group > 0x13: NOx Adsorber Monitoring Group > 0x14: PM Filter Monitoring Group > 0x15: Oxygen Sensor Monitoring Group > 0x17: Non Continuous Fuel System Group > 0x18: Oxygen Sensor Heater Group > 0x19: AC System Component Monitoring Group > 0x1A: Exhaust Gas Heating Sensor Monitoring Group > 0x1B: Cold Start Aid System Monitoring Group Any not listed Readiness Groups are currently not supported.
`DTCFormat`	Defines the output-format of the requested DTC values for the sub-sequent API calls. Currently limited to: DEM_DTC_FORMAT_UDS: report DTC in UDS format DEM_DTC_FORMAT_OBD_3BYTE: report DTC in OBD 3 byte format
Return code	
`Std_ReturnType`	E_OK: Selection processed successfully. E_NOT_OK: Invalid parameters passed to the function or the feature for service 0x19 subfunction 0x56 support is disabled.

Functional Description
Sets the filter to be used by Dem_GetNumberOfFilteredDTC() (see [1]) and Dem_GetNextFilteredDTC() (see [1]). Sets the filter to OBD relevant DTCs, which are part of the provided readiness group. Sets the filter to return DTCs in the requested format. Note: This function will overwrite the filter set by Dem_SetDTCFilter() (see [1]).
Particularities and Limitations
> This function is reentrant for different ClientIds. > This function is synchronous. > Only functional if 'DemSupportDcm' and 'DemSupportService19x56' is set to enabled. > This function is only callable from the master partition.
Expected Caller Context
> This function can be called from any context.

Table 4-18 Dem_SetDTCFilterByReadinessGroup()

4.1.1.19 Dem_DcmGetInfoTypeValue79()

Prototype	
`Std_ReturnType` **`Dem_DcmGetInfoTypeValue79`**`( Dcm_OpStatusType OpStatus, uint8* DataValueBuffer, uint8* DataValueBufferSize )`	
Parameter	
`OpStatus`	This parameter is not used.
`DataValueBuffer`	The pointer to the buffer where the Monitor Activity Denominator (MAD) computed by the Dem shall be written to.
`DataValueBufferSize`	When the function is called this parameter contains the maximum number of data bytes that can be written to the buffer. The function returns the actual number of written data bytes in this parameter.
Return code	
`Std_ReturnType`	E_OK: Denominator value was successfully written into the buffer. E_NOT_OK: is returned with enabled DET when an error is detected. DEM_BUFFER_TOO_SMALL: The provided destination buffer is too small.
Functional Description	
Provides the Monitor Activity Denominator (MAD) of the Monitor Activity Ratio (MAR).	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition.	
Call context	
> This function can be called from any context.	

Table 4-19 Dem_DcmGetInfoTypeValue79()

4.1.2 SWC and CDD

4.1.2.1 Dem_SetEventDisabled()

Prototype	
Std_ReturnType **Dem_SetEventDisabled** (Dem_EventIdType EventId)	
Parameter	
EventId	Identification of an event by assigned EventId.
Return code	
Std_ReturnType	E_OK: set of event to disabled was successful. E_NOT_OK: set of event disabled failed
Functional Description	
Service for reporting the event as disabled. A disabled event will set referenced readiness group to not supported.	
Particularities and Limitations	
> This function is reentrant for different EventIds. > This function is synchronous. > This function is only callable from the master partition	

Table 4-20 Dem_SetEventDisabled()

4.1.2.2 Dem_RepIUMPRFaultDetect()

Prototype	
Std_ReturnType **Dem_RepIUMPRFaultDetect** (Dem_RatioIdType RatioID)	
Parameter	
RatioID	Ratio Identifier reporting that a respective monitor could have found a fault. If no RTE is used pass the EventId instead.
Return code	
Std_ReturnType	E_OK: report of IUMPR result was successfully reported E_NOT_OK: the monitor status was not accepted. This can be caused by undefined ratio ids, referenced ratio is not of type “API”, the RatioId/ EventId is not available or the related event is affected by active PTO
Functional Description	
Service for reporting that faults could have been found because of all conditions are fulfilled.	
Particularities and Limitations	
> This function is reentrant for different RatioIDs. > This function is only callable from the master partition	

Table 4-21 Dem_RepIUMPRFaultDetect()

4.1.2.3 Dem_RepIUMPRDenLock()

Prototype	
`Std_ReturnType` **`Dem_RepIUMPRDenLock`** `( Dem_RatioIdType RatioID )`	
Parameter	
`RatioID`	Ratio identifier to select the specific denominator which shall be locked. If no RTE is used pass the EventId instead.
Return code	
`Std_ReturnType`	E_OK: report of IUMPR denominator status was successfully reported E_NOT_OK: report of IUMPR denominator status was not successfully reported
Functional Description	
Service is used to lock the denominator of a specific monitor.	
Particularities and Limitations	
> This function is reentrant for different RatioIDs. > This function is synchronous. > This function is only callable from the master partition	

Table 4-22 Dem_RepIUMPRDenLock()

4.1.2.4 Dem_RepIUMPRDenRelease()

Prototype	
`Std_ReturnType` **`Dem_RepIUMPRDenRelease`** `( Dem_RatioIdType RatioID )`	
Parameter	
`RatioID`	Ratio identifier to select the specific denominator which shall be released. If no RTE is used pass the EventId instead.
Return code	
`Std_ReturnType`	E_OK: report of IUMPR denominator status was successfully reported E_NOT_OK: report of IUMPR denominator status was not successfully reported
Functional Description	
Service is used to release the denominator of a specific monitor.	
Particularities and Limitations	
> This function is reentrant for different RatioIDs. > This function is synchronous. > This function is only callable from the master partition	

Table 4-23 Dem_RepIUMPRDenRelease()

4.1.2.5 Dem_SetPfcCycleQualified()

This chapter is only relevant for **OBD II** and **OBDonUDS**.

Prototype	
Std_ReturnType **Dem_SetPfcCycleQualified** (void) Std_ReturnType **Dem_SetPfcCycle** (void)	
Parameter	
void	
Return code	
Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
Functional Description	
Marks the current OBD driving cycle as having met the criteria for the PFC cycle. API Dem_SetPfcCycle is available for compatibility reasons. Please use Dem_SetPfcCycleQualified instead.	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-24 Dem_SetPfcCycleQualified()

4.1.2.6 Dem_IUMPRLockNumerators()

Prototype	
Std_ReturnType **Dem_IUMPRLockNumerators** (void)	
Parameter	
void	
Return code	
Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
Functional Description	
Block numerator increments for this driving cycle. Denominators are still incremented if the respective criteria are met.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-25 Dem_IUMPRLockNumerators()

4.1.2.7 Dem_EnablePermanentStorage()

This chapter is only relevant for **OBD II** and **OBDonUDS**.

Prototype	
`Std_ReturnType` **`Dem_EnablePermanentStorage`** `()`	
Parameter	
`void`	
Return code	
`Std_ReturnType`	Always E_OK is returned, as E_NOT_OK will never appear.
Functional Description	
Activates permanent storage after initial power-up.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition > This function has to be called from same task where Dem_MainFunction() is scheduled to avoid mutual interruptions.	

Table 4-26 Dem_EnablePermanentStorage ()

4.1.2.8 Dem_GetPermanentStorageState()

OBDII

OBD on UDS

This chapter is only relevant for **OBD II** and **OBDonUDS**.

Prototype	
Std_ReturnType **Dem_GetPermanentStorageState** (boolean* isEnabled)	
Parameter	
isEnabled	True: permanent memory storage is enabled False: permanent memory storage is not enabled
Return code	
Std_ReturnType	Always E_OK is returned, as E_NOT_OK will never appear.
Functional Description	
Returns the permanent storage state.	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-27 Dem_GetPermanentStorageState ()

4.1.2.9 Dem_SetIUMPRDenCondition()

Prototype	
Std_ReturnType **Dem_SetIUMPRDenCondition** (Dem_IumprDenomCondIdType ConditionId, Dem_IumprDenomCondStatusType ConditionStatus)	
Parameter	
ConditionId	Identification of an IUMPR denominator condition ID: DEM_IUMPR_GENERAL_DENOMINATOR: general individual denominator condition and increment of the general denominator DEM_IUMPR_DEN_COND_COLDSTART: additional IUMPR condition “cold start” DEM_IUMPR_DEN_COND_EVAP: additional IUMPR condition “EVAP” DEM_IUMPR_DEN_COND_500MI: additional IUMPR condition „500 miles“ DEM_IUMPR_GENERAL_INDIVIDUAL_DENOMINATOR: general individual denominator condition DEM_IUMPR_GENERAL_OBDCOND: increment of the general denominator DEM_IUMPR_DEN_COND_INDEPENDENT: IUMPR condition independent of the general individual denominator
ConditionStatus	Status of the IUMPR denominator condition: DEM_IUMPR_DEN_STATUS_REACHED: Conditions are fulfilled. DEM_IUMPR_DEN_STATUS_NOT_REACHED: Conditions are not (yet) fulfilled. DEM_IUMPR_DEN_STATUS_INHIBITED: Conditions are inhibited for the rest of the current driving cycle due to an error. This status can only be reported for ConditionIDs DEM_IUMPR_GENERAL_DENOMINATOR, DEM_IUMPR_GENERAL_INDIVIDUAL_DENOMINATOR or DEM_IUMPR_GENERAL_OBDCOND. An inhibited general individual denominator condition leads to locking of all ratios (numerators and denominators).
Return code	
Std_ReturnType	E_OK: IUMPR denominator conditions have been set successfully E_NOT_OK: IUMPR denominator conditions have not been set
Functional Description	
In order to communicate the status of general and additional denominator conditions among the OBD relevant ECUs, the API is used to forward the condition status to the Dem.	
Particularities and Limitations	
> This function is reentrant for different condition IDs with following exception: > The three condition IDs DEM_IUMPR_DEN_COND_INDEPENDENT, DEM_IUMPR_GENERAL_OBDCOND, DEM_IUMPR_GENERAL_DENOMINATOR cannot be reported reentrant of each other. But reported with other condition IDs they are considered as reentrant. > This function is partly synchronous. > Synchronous: Denominator status and General Denominator is processed > Asynchronous: Ratio specific denominators are processed > This function is only callable from the master partition	

Table 4-28 Dem_SetIUMPRDenCondition()

4.1.2.10 Dem_SetIUMPRFilter()

Prototype	
`Std_ReturnType Dem_SetIUMPRFilter ( Dem_IumprReadinessGroupType IumprReadinessGroup, Dem_DTCOriginType DTCOrigin )`	
Parameter	
`IumprReadinessGroup`	The readiness group that shall be used for the filter
`DTCOrigin`	The memory origin which shall be used for the filter. Supported origin is either primary memory or user defined memory.
Return code	
`Std_ReturnType`	E_OK: Filter was successfully set E_NOT_OK: Filter was not set
Functional Description	
Sets the criteria for which the ratios shall be filtered for.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-29 Dem_SetIUMPRFilter()

4.1.2.11 Dem_GetNumberOfFilteredIUMPR()

Prototype	
`Std_ReturnType Dem_GetNumberOfFilteredIUMPR ( uint16* NumberOfFilteredRatios )`	
Parameter	
`NumberOfFilteredRatios`	The number of ratios matching the filter criteria.
Return code	
`Std_ReturnType`	E_OK: Number of filtered ratios retrieved successfully E_NOT_OK: Number of filtered ratios not retrieved successfully
Functional Description	
Returns the number of ratios matching the filter criteria. Before using this API, Dem_SetIUMPRFilter() must have been called (see also 2.3.7.3).	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-30 Dem_GetNumberOfFilteredIUMPR()

4.1.2.12 Dem_GetIUMPRGeneralData()

Prototype	

`Std_ReturnType` **`Dem_GetIUMPRGeneralData`** `(`
`uint16* GeneralDenominator,`
`uint16* IgnitionsCycles,`
`uint16* IgnitionCyclesHybrid )`	

Parameter	
`GeneralDenominator`	current value of the General Denominator
`IgnitionsCycles`	current value of the IgnitionCycleCounter
`IgnitionCyclesHybrid`	current value of the second IgnitionCycleCounter that is required by law for hybrid vehicles if it is not supported, the value is set to 0
Return code	
`Std_ReturnType`	E_OK: IUMPR data has been retrieved successfully E_NOT_OK: IUMPR data could not be retrieved
Functional Description	
Returns the ratio independent counters for IUMPR.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-31 Dem_GetIUMPRGeneralData()

4.1.2.13 Dem_GetNextIUMPRRatioDataAndDTC()

Prototype	

`Std_ReturnType` **`Dem_GetNextIUMPRRatioDataAndDTC`** `(`
`uint32* UdsDtcNumber,`
`uint16* ObdDtcNumber,`
`uint16* Denominator,`
`uint16* Numerator )`	

Parameter	
`UdsDtcNumber`	UDS DTC that is related to the IUMPR ratio
`ObdDtcNumber`	OBD II DTC that is related to the IUMPR ratio For WWH-OBD this parameter is set to 0
`Denominator`	current value of the ratio specific Denominator
`Numerator`	current value of the ratio specific Numerator
Return code	
`Std_ReturnType`	E_OK: IUMPR data has been retrieved successfully E_NOT_OK: IUMPR data could not be retrieved (e.g. no more ratios available)

Functional Description
Returns the ratio specific counters for an IUMPR ratio. Before using this API, Dem_SetIUMPRFilter() must have been called. Afterwards for each available ratio Dem_GetNextIUMPRRatioDataAndDTC() have to be called. If all ratios have been read, the API will return E_NOT_OK. To restart reading the ratios, Dem_SetIUMPRFilter() must be called again. (see also 2.3.7.3)
Particularities and Limitations
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition

Table 4-32 Dem_GetNextIUMPRRatioDataAndDTC()

4.1.2.14 Dem_GetCurrentIUMPRRatioDataAndDTC()

Prototype	

`Std_ReturnType` **`Dem_CurrentIUMPRRatioDataAndDTC`** `(`
`  uint32* UdsDtcNumber,`
`  uint16* ObdDtcNumber,`
`  uint16* Denominator,`
`  uint16* Numerator )`	

Parameter	
`UdsDtcNumber`	UDS DTC that is related to the IUMPR ratio
`ObdDtcNumber`	OBD II DTC that is related to the IUMPR ratio For WWH-OBD this parameter is set to 0
`Denominator`	current value of the ratio specific Denominator
`Numerator`	current value of the ratio specific Numerator
Return code	
`Std_ReturnType`	E_OK: IUMPR data has been retrieved successfully E_NOT_OK: IUMPR data could not be retrieved (e.g. read sequence not yet started)
Functional Description	
Returns the ratio specific counters for an IUMPR ratio. Before using this API, Dem_SetIUMPRFilter() and Dem_GetNextIUMPRRatioDataAndDTC() must have been called (see also 2.3.7.3).	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-33 Dem_GetCurrentIUMPRRatioDataAndDTC()

4.1.2.15 Dem_SetHideObdOccurrences()

Prototype	
Std_ReturnType **Dem_SetHideObdOccurrences** (boolean DoHide)	
Parameter	
DoHide	True: hide occurrences False: do not hide occurrences
Return code	
Std_ReturnType	E_OK: operation was successful E_NOT_OK: operation failed
Functional Description	
Select whether the Dem reports Dependent Secondary ECU DTCs in Dcm responses.	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-34 Dem_SetHideObdOccurrences()

4.1.2.16 Dem_GetHideObdOccurrences()

Prototype	
Std_ReturnType **Dem_GetHideObdOccurrences** (boolean* IsHidden)	
Parameter	
IsHidden	True: occurrences hidden False: occurrences not hidden
Return code	
Std_ReturnType	E_OK: operation was successful E_NOT_OK: operation failed
Functional Description	
Test whether the Dem reports Dependent Secondary ECU DTCs in Dcm responses.	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-35 Dem_GetHideObdOccurrences()

4.1.2.17 Dem_ReadDataOfPID01()

Prototype	
`Std_ReturnType` **`Dem_ReadDataOfPID01`** `( uint8* PID01value )`	
Parameter	
`PID01value`	Buffer containing the contents of PID $01 computed by the Dem.
Return code	
`Std_ReturnType`	E_OK: is always returned with disabled DET E_NOT_OK: is returned with enabled DET when an error is detected
Functional Description	
Provides the monitoring status since DTCs have been cleared. For a detailed description of the status bits, refer to [6].	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-36 Dem_ReadDataOfPID01()

4.1.2.18 Dem_GetB1Counter()

This chapter is only relevant for **WWH-OBD**.

Prototype	
`Std_ReturnType` **`Dem_GetB1Counter`** `( uint16* B1Counter )`	
Parameter	
`B1Counter`	Buffer containing the current value of the B1 counter.
Return code	
`Std_ReturnType`	E_OK: is always returned with disabled DET E_NOT_OK: is returned with enabled DET when an error is detected
Functional Description	
Provides the current value of the B1 counter.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-37 Dem_GetB1Counter()

4.1.2.19 Dem_SetDTR()

<table>
<tr><th colspan="2">Prototype</th></tr>
<tr><td colspan="2">Std_ReturnType <b>Dem_SetDTR</b> (uint16 DTRId, sint32 TestResult, sint32 LowerLimit, sint32 UpperLimit, Dem_DTRControlType Ctrlval)</td></tr>
<tr><th colspan="2">Parameter</th></tr>
<tr><td>DTRId</td><td>Identification of a DTR element by assigned DTR Id.</td></tr>
<tr><td>TestResult</td><td>Test result to be stored for the DTR Id.</td></tr>
<tr><td>LowerLimit</td><td>Lower limit to be stored for the DTR Id.</td></tr>
<tr><td>UpperLimit</td><td>Upper limit to be stored for the DTR Id.</td></tr>
<tr><td>Ctrlval</td><td>Control value for the interpretation of the reported test result:
DEM_DTR_CTL_NORMAL: Values are reported and regarded as valid test result.
DEM_DTR_CTL_NO_MAX: Values are reported, but upper limit is not valid. Thus upper limit value is ignored.
DEM_DTR_CTL_NO_MIN: Values are reported, but lower limit is not valid. Thus lower limit value is ignored.
DEM_DTR_CTL_RESET: All values for the DTR Id are reset to zero.
DEM_DTR_CTL_INVISIBLE: All values are ignored. The DTR Id is treated as if not integrated until new values for the DTR Id are reported.</td></tr>
<tr><th colspan="2">Return code</th></tr>
<tr><td>Std_ReturnType</td><td>E_OK: Report of DTR result was successful.
E_NOT_OK: Report of DTR result failed.</td></tr>
<tr><th colspan="2">Functional Description</th></tr>
<tr><td colspan="2">Sets a test result with lower and upper limit for a DTR Id.
Depending on the configured DTR update kind, the de-bouncing state of the assigned event and the control value, the DTR values are either processed, reset or ignored.
Reporting of DTRs doesn’t lead to an increment of the IUMPR numerator of a related ratio.</td></tr>
<tr><th colspan="2">Particularities and Limitations</th></tr>
<tr><td colspan="2">&gt; This function is reentrant.
> This function is synchronous.
> This function is only callable from the master partition</td></tr>
</table>

Table 4-38 Dem_SetDTR()

4.1.2.20 Dem_UpdateAvailableOBDMIDs()

Prototype	
`Std_ReturnType` **`Dem_UpdateAvailableOBDMIDs`** `( void )`	
Parameter	
`void`	
Return code	
`Std_ReturnType`	Always E_OK is returned, as E_NOT_OK will never appear.
Functional Description	
Requests the update of the available OBDMIDs.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-39 Dem_UpdateAvailableOBDMIDs()

4.1.2.21 Dem_SetPtoStatus()

Prototype	
`Std_ReturnType` **`Dem_SetPtoStatus`** `( boolean )`	
Parameter	
`PtoStatus`	Sets the status of the PTO True: PTO is active False: PTO is inactive
Return code	
`Std_ReturnType`	E_OK: PTO status has been adopted by DEM E_NOT_OK: otherwise
Functional Description	
Notifies the change in PTO status.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 4-40 Dem_SetPtoStatus()

4.1.2.22 Dem_DisableEventMonitoringInCurrentDcy()

This chapter is only relevant for **OBD II** and **OBDonUDS**.

Prototype	
`Std_ReturnType` **`Dem_DisableEventMonitoringInCurrentDcy`** `( Dem_EventIdType EventId )`	
Parameter	
`EventId`	Identification of an event by assigned EventId
Return code	
`Std_ReturnType`	E_OK: set event monitor to disabled was successful E_NOT_OK: set event monitor disabled failed
Functional Description	
Indicates to DEM that event monitoring is disabled in the current driving cycle (subject to requirements in [8]). The disabled event monitor will be considered as tested completed during the readiness calculation of the following readiness data: PID $01, PID $F501 and DM05.	
Particularities and Limitations	
> This function is reentrant for different EventIds. > This function is synchronous. > This function is only callable from the master partition > Event reporting will override the disabled event monitor and readiness group will be re-calculated as described in 2.2.9.1. > This function may only be called to indicate to DEM that monitoring has been disabled for multiple driving cycles due to the continued presence of extreme operating conditions (e.g., cold ambient temperatures, high altitudes) (for details, refer to [8]).	

Table 4-41 Dem_DisableEventMonitoringInCurrentDcy ()

4.1.3 J1939Dcm

Note
Dependent on the licensed components of your delivery the feature J1939 may not be available in DEM.

This chapter is only relevant for **OBD II**.

4.1.3.1 Dem_J1939DcmSetRatioFilter()

<table>
<tr><th colspan="2">Prototype</th></tr>
<tr><td colspan="2"><code>Std_ReturnType <b>Dem_J1939DcmSetRatioFilter</b> (uint16* IgnitionCycleCounter, uint16* OBDMonitoringConditionsEncountered, uint8 ClientId)</code></td></tr>
<tr><th colspan="2">Parameter</th></tr>
<tr><td><code>IgnitionCycleCounter</code></td><td>Receives the Ignition Cycle Counter. If the return value of the function is E_NOT_OK this parameter does not contain valid data.</td></tr>
<tr><td><code>OBDMonitoring Conditions Encountered</code></td><td>Receives the General Denominator. If the return value of the function is E_NOT_OK this parameter does not contain valid data.</td></tr>
<tr><td><code>ClientId</code></td><td>Unique client id, assigned to the instance of the calling module.</td></tr>
<tr><th colspan="2">Return code</th></tr>
<tr><td><code>Std_ReturnType</code></td><td>E_OK: operation was successful
E_NOT_OK: Filter could not be set</td></tr>
<tr><th colspan="2">Functional Description</th></tr>
<tr><td colspan="2">Sets the Ratio filter for a specific node and returns the Ignition Cycle Counter and General Denominator.</td></tr>
<tr><th colspan="2">Particularities and Limitations</th></tr>
<tr><td colspan="2">&gt; This function is reentrant for different ClientIds.
> This function is synchronous.
> This function is only callable from the master partition.
> Only available if ‘DemJ1939RatioSupport’ is set to enabled.</td></tr>
</table>

Table 4-42 Dem_J1939DcmSetRatioFilter()

4.1.3.2 Dem_J1939DcmGetNextFilteredRatio()

<table>
<tr><th colspan="2">Prototype</th></tr>
<tr><td colspan="2"><code>Std_ReturnType <b>Dem_J1939DcmGetNextFilteredRatio</b> (uint32* SPN, uint16* Numerator, uint16* Denominator, uint8 ClientId)</code></td></tr>
<tr><th colspan="2">Parameter</th></tr>
<tr><td><code>SPN</code></td><td>Receives the SPN of the applicable system monitor. If the return value of the function is other than E_OK this parameter does not contain valid data.</td></tr>
<tr><td><code>Numerator</code></td><td>Receives the Numerator of the applicable system monitor. If the return value of the function is other than E_OK this parameter does not contain valid data</td></tr>
<tr><td><code>Denominator</code></td><td>Receives the Denominator of the applicable system monitor. If the return value of the function is other than E_OK this parameter does not contain valid data.</td></tr>
<tr><td><code>ClientId</code></td><td>Unique client id, assigned to the instance of the calling module.</td></tr>
</table>

Return code	
Std_ReturnType	E_OK: operation was successful DEM_NO_SUCH_ELEMENT: The requested element is not available DEM_PENDING: Operation successful and result pending DEM_BUFFER_TOO_SMALL: The provided buffer is too small
Functional Description	
Gets the next filtered Ratio.	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition. > Only available if 'DemJ1939RatioSupport' is set to enabled.	

Table 4-43 Dem_J1939DcmGetNextFilteredRatio()

4.1.3.3 Dem_J1939DcmReadDiagnosticReadiness2()

Prototype	
Std_ReturnType **Dem_J1939DcmReadDiagnosticReadiness2** (Dem_J1939DcmDiagnosticReadiness2Type* DataValue, uint8 ClientId)	
Parameter	
DataValue	Buffer of 8 bytes containing the contents of Diagnostic Readiness 2 (DM21) computed by the Dem.
ClientId	Unique client id, assigned to the instance of the calling module.
Return code	
Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Diagnostic readiness couldn't be read, or DM21 is disabled, or J1939 is disabled
Functional Description	
Service to report the value of Diagnostic Readiness 2 (DM21) computed by the Dem.	
Particularities and Limitations	
> This function is reentrant for different ClientIds. > This function is synchronous. > This function is only callable from the master partition. > Only available if 'DemJ1939Readiness2Support' is set to enabled.	

Table 4-44 Dem_J1939DcmReadDiagnosticReadiness2()

4.1.3.4 Dem_J1939DcmReadDiagnosticReadiness3()

Prototype	

`Std_ReturnType` **`Dem_J1939DcmReadDiagnosticReadiness3`** `(`
`Dem_J1939DcmDiagnosticReadiness3Type* DataValue, uint8 ClientId)`	

Parameter	
`DataValue`	Buffer of 8 bytes containing the contents of Diagnostic Readiness 3 (DM26) computed by the Dem.
`ClientId`	Unique client id, assigned to the instance of the calling module.
Return code	
`Std_ReturnType`	E_OK: Operation was successful E_NOT_OK: Diagnostic readiness couldn't be read, or DM26 is disabled, or J1939 is disabled
Functional Description	
Service to report the value of Diagnostic Readiness 3 (DM26) computed by the Dem.	
Particularities and Limitations	
> This function is reentrant for different ClientIds. > This function is synchronous. > This function is only callable from the master partition. > Only available if ‘DemJ1939Readiness3Support’ is set to enabled.	

Table 4-45 Dem_J1939DcmReadDiagnosticReadiness3()

4.2 Service Ports

4.2.1 Client Server Interface

4.2.1.1 Provided Ports

The following ports are only available at the DemMaster SWC (cf. [1]).

4.2.1.1.1 OBDServices

Operation	Legislation	API	Arguments
SetPfcCycle	OBDII OBD on UDS	Dem_SetPfcCycleQualified	ERR{E_NOT_OK}
EnablePermanentStorage	OBDII OBD on UDS	Dem_EnablePermanentStorage	ERR{E_NOT_OK}
GetPermanentStorageState	OBDII OBD on UDS	Dem_GetPermanentStorageState	OUT boolean isEnabled, ERR{E_NOT_OK}
IUMPRLockNumerators	OBDII OBD on UDS WWH-OBD	Dem_IUMPRLockNumerators	ERR{E_NOT_OK}
SetHideOccurrences	OBDII OBD on UDS WWH-OBD	Dem_SetHideObdOccurrences	IN boolean DoHide, ERR{E_NOT_OK}
GetHideOccurrences	OBDII OBD on UDS WWH-OBD	Dem_GetHideObdOccurrences	OUT boolean isHidden, ERR{E_NOT_OK}

Operation	Legislation	API	Arguments
ReadDataOfPID01	OBDII OBD on UDS WWH-OBD	Dem_ReadDataOfPID01	OUT uint8 PID01Value, ERR{E_NOT_OK}
GetB1Counter	WWH-OBD	Dem_GetB1Counter	OUT uint16 B1Counter, ERR{E_NOT_OK}
DisableEventMonitoringCurrentDcy	OBDII OBD on UDS WWH-OBD	Dem_DisableEventMonitoringInCurrentDcy	IN Dem_EventIdType EventId, ERR{E_NOT_OK}

Table 4-46 Port OBDServices

4.2.1.1.2 IUMPRDenominator

Operation	API	Arguments
RepIUMPRDenLock	Dem_RepIUMPRDenLock	ERR{E_NOT_OK}
RepIUMPRDenRelease	Dem_RepIUMPRDenRelease	ERR{E_NOT_OK}

Table 4-47 Port IUMPRDenominator

4.2.1.1.3 IUMPRDenominatorCondition

Operation	API	Arguments
SetIUMPRDenCondition	Dem_SetIUMPRDenCondition	IN Dem_IumprDenomCondStatusType ConditionStatus, ERR{E_NOT_OK}

Table 4-48 Port IUMPRDenominatorCondition

4.2.1.1.4 IUMPRNumerator

Operation	API	Arguments
RepIUMPRFaultDetect	Dem_RepIUMPRFaultDetect	ERR{E_NOT_OK}

Table 4-49 Port IUMPRNumerator

4.2.1.1.5 IUMPRData

Operation	API	Arguments
SetIUMPRFilter	Dem_ SetIUMPRFilter	IN Dem_IumprReadinessGroupType IumprReadinessGroup, IN Dem_DTCOriginType DTCOrigin, ERR{E_NOT_OK}
GetNumberOfFilteredIUMPR	Dem_ GetNumberOfFilteredIUMPR	OUT uint16 NumberOfFilteredRatios, ERR{E_NOT_OK}
GetIUMPRGeneralData	Dem_GetIUMPRGeneralData	OUT uint16 GeneralDenominator, OUT uint16 IgnitionCycles, OUT uint16 IgnitionCycleCounterHybrid, ERR{E_NOT_OK}
GetCurrentIUMPRRatioDataAndDTC	Dem_GetCurrentIUMPRRatioDataAndDTC	OUT uint32 UdsDtcNumber, OUT uint16 ObdDtcNumber, OUT uint16 Denominator,

Operation	API	Arguments
		OUT uint16 Numerator, ERR{E_NOT_OK}
GetNextIUMPRRatioDataAndDTC	Dem_GetNextIUMPRRatioDataAndDTC	OUT uint32 UdsDtcNumber, OUT uint16 ObdDtcNumber, OUT uint16 Denominator, OUT uint16 Numerator, ERR{E_NOT_OK}

Table 4-50 Port IUMPRData

4.2.1.1.6 DTRCentralReport

Operation	API	Arguments
SetDTR	Dem_SetDTR	IN sint32 TestResult, IN sint32 LowerLimit, IN sint32 UpperLimit, IN Dem_DTRControlType CtrlVal, ERR{E_NOT_OK}

Table 4-51 Port DTRCentralReport

4.2.1.1.7 PowerTakeOff

Operation	API	Arguments
SetPtoStatus	Dem_SetPtoStatus	IN boolean PtoStatus, ERR{E_NOT_OK}

Table 4-52 Port PowerTakeOff

5 Glossary and Abbreviations

5.1 Glossary

Term	Description
Black MIL DTC	An OBD relevant DTC which does not trigger the MIL.
MIL indicator	A warning indicator assigned to the MIL which is managed by the Dem. The warning indicator only provides the information that the related indicator (e.g. lamp in the dashboard) shall be requested, the de-/activation must be handled by the application or a different ECU. Each event that currently requests an indicator will have set the warning indicator requested bit in the status byte.
OBD relevant DTC	A DTC which has the parameter DemFunctionalGroupIdentifier set to 0x33.
OBD relevant event	An event which is attached to an OBD relevant DTC.

Table 5-1 Glossary

5.2 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
BSW	Basis Software
CARB	California Air Resources Board
CCR	California Code Regulations
DCM	Diagnostic Communication Manager
DCY	OBD Driving Cycle
DEM	Diagnostic Event Manager
DET	Development Error Tracer
DLT	Diagnostic Log and Trace
DTC	Diagnostic Trouble Code
DTR	Diagnostic Test Result
ECU	Electronic Control Unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
EOBD	European On-Board Diagnostics
FF	Freeze Frame
FID	Function Identifier
FiM	Function Inhibition Manager
GTR	Global Technical Regulation
HIS	Hersteller Initiative Software
ID	Identification
ISO	International Organization for Standardization

ISR	Interrupt Service Routine
IUMPR	In-Use Monitor Performance Ratio
MAD	Monitor Activity Denominator
MAN	Monitor Activity Numerator
MAR	Monitor Activity Ratio
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)
MID	Monitor Identifier
MIL	Malfunction Indicator Lamp
NVRAM	Non-volatile Random Access Memory
OBD	On-Board Diagnostic
OBDonUDS	OBD on UDS
OTL	OBD Threshold Limits
PFC	Permanent Fault Code
PID	Parameter Identifier
PTO	Power take-off
RAM	Random Access Memory
ROM	Read-Only Memory
RTE	Runtime Environment
SAE	Society of Automotive Engineers
SchM	Schedule Manager
SMAR	Stored Monitor Activity Ratio
SPN	Suspect Parameter Number
SRS	Software Requirement Specification
SWC	Software Component
SWS	Software Specification
TID	Test Identifier
UDS	Unified Diagnostic Services
WUC	OBD Warm-Up Cycle
WWH-OBD	World Wide Harmonized On-Board Diagnostics
ZEV-OBD	Zero Emission Vehicle

Table 5-2 Abbreviations

6 Contact

Visit our website for more information on

- News
- Products
- Demo software
- Support
- Training data
- Addresses

www.vector.com