VECTOR

MICROSAR Diagnostic Event Manager (DEM)

Technical Reference

Addendum for OBD II and WWH-OBD

Version 3.03.00

Authors	Thomas Dedler, Alexander Ditte, Savas Ates, Anna Bosch, Erik Jeglorz
Status	Released

Document Information

History

Author	Date	Version	Remarks
T. Dedler	2012-08-31	0.01.00	> preliminary version
T. Dedler	2012-10-02	1.00.00	> beta release
T. Dedler	2012-12-12	1.01.00	> IUMPR, DTC Combination and MIL Groups added
A. Ditte T. Dedler	2013-02-18	1.02.00	> Measurement and Calibration added > Permanent Storage Suppression added > rework of the topics “Integration”, “Freeze Frame” and “Permanent Storage”
A. Ditte	2013-04-03	1.02.01	> fixed ESCAN00066337 “Incorrect description of operation cycle usage” in *chapter 4*
A. Ditte T. Dedler	2013-04-10	1.03.00	> update of chapter *6 Post-Build* > API *Dem_GetNextIUMPRRatioDataAndDTC()* added
T. Dedler A. Ditte	2013-06-04	1.04.00	> update of Odometer value description in chapter *4.2* > extended description in chapter *3.5.5* for relative permanent DTC activation distance > limitations with event combination added
T. Dedler	2013-07-22	1.04.01	> Critical Section chapter reworked
A. Ditte	2013-07-30	1.05.00	> add multiple freeze frame in *chapter 0*
A. Ditte T. Dedler	2013-08-22	1.05.01	> add *Figure 4-1 OBD Driving Cycle State Transitions* > improved description in chapter *3.2.2* > fixed DTCStatusMask in chapter *3.9.1.7*
T. Dedler A. Ditte	2013-09-04	1.06.00	> API *Dem_GetCurrentIUMPRRatioDataAndDTC()* added > add 40 warm-up cycles reset condition to chapter *3.8.4* > introduced chapter *3.3.3* and *4.1.2*
A. Ditte	2013-11-06	1.06.01	> API *Dem_GetPermanentStorageState()* added > limitation *8.2.1 Aging Behavior* added > fixed ESCAN00071675: Incorrect indicator healing cycle threshold in configuration example > add *3.9.2.1 Service 14: Clear DTC* > update of chapter *7.2.1.1 Provided Ports*
A. Ditte	2014-01-14	1.07.00	> renamed DCM interfaces according AUTOSAR 4.1.2 > added new limitation for API Dem_ClearDTC()

T. Dedler A. Ditte	2014-03-10	1.08.00	> fixed ESCAN00073613: dependency between general denominator and numerator calculation (*chapter 3.10*) > fixed ESCAN00073876: Lock Numerator increment during active tester communication (*chapter 7*, *chapter 3.10*) > chapter *3.2.4 Dependent Secondary ECU DTCs* added > APIs *Dem_SetHideObdOccurrences()* and *Dem_GetHideObdOccurrences()* added
A. Ditte	2014-04-28	1.08.01	> Fixed ESCAN00075220: Remove IUMPR for combined events limitation (*chapter 4.7.3 and chapter 8.2*) > Changed figures in chapter *3.4.1 Storage of Legislative Freeze* Frame
A. Ditte	2014-06-16	1.09.00	> Fixed ESCAN00076296: Incorrect Return Code for API Dem_DcmGetOBDFreezeFrame (*chapter 7.1.1.15*) > Added chapter *3.12 Non-Volatile Data Management* > Update of Figure *4-2 Relation between Dem and FiM for IUMPR* > Update of chapter *6 Post-Build* > Added chapter *5 Configuration*
A. Ditte	2014-11-03	2.00.00	> Added WWH-OBD (beta) > Added *Figure 3-5* and update of *Figure 3-19* > Removed deprecated API *Dem_GetNextIUMPRRatioData()* > Added clarification for 1-trip DTCs in chapter *3.3.2*
A. Ditte	2015-05-05	02.01.00	> Finalized WWH-OBD > Added new limitation in *8.3.3* > Fixed ESCAN00086484: Missing description for permanent memory re-initialization (chapter *4.6.2*)
A. Ditte	2015-11-13	02.01.01	> Added missing ESCAN reference to 2.01.00 > Fixed numbering of *4.6.2 NVRAM Initialization*
A. Ditte	2016-01-08	02.01.02	> Fixed *ESCAN00087317*: Incorrect time for MIL on shown in figure (chapter *3.5.4*) > Fixed *ESCAN00087319*: Missing OBD II tag for PID 02 (chapter *3.8.2*) > Fixed *ESCAN00088091*: Some APIs erroneously declared as not supported (chapter *0*) > Migration to new CI-Template

A. Ditte S. Ates A. Bosch	2016-06-07	02.02.00	> Added new APIs *Dem_SetIUMPRFilter()* and *Dem_GetNumberOfFilteredIUMPR()* > Removed number of ratios returned in API *Dem_GetIUMPRGeneralData()* > Changed IUMPR data retrieval in chapter *4.7.3* > Added ratio dependent clear permanent DTC behavior to chapter *3.5.4* and *5.1.8* > Added chapter *3.2.5 Black MIL DTCs* > Added description for RatioID equals EventId to chapter 7.1.2 > Changed description for PID $01 and PID $41 calculation in chapter 3.8.1 and 3.8.7 > Added new configuration parameters for PID $01 and PID $41 calculation in chapter 5.1.8 and 5.2.2 > Renamed API Dem_SetPfcCycle() to *Dem_SetPfcCycleQualified()* > Added chapter *3.3.4 Similar Conditions* > Add new chapters regarding DTRs (*3.9.1.5 Mode $06: Request Diagnostic Test Results*, *3.11 Diagnostic Test Result, 4.8 DTR Integration*, *5.2.4 DTR* and *7.2.1.1.6 DTRCentralReport*) > Added new APIs regarding DTRs (*Dem_DcmGetAvailableOBDMIDs()*, *Dem_DcmGetNumTIDsOfOBDMID()*, *Dem_DcmGetDTRData()*, *Dem_SetDTR()* and *Dem_UpdateAvailableOBDMIDs()*)
A. Bosch A. Ditte	2016-09-27	02.02.01	> Adapted chapter *3.11 Diagnostic Test Result* > Clarifications in chapter *3.4.3*
A. Bosch	2017-05-08	03.00.00	> Fixed ESCAN00091941 (chapter *3.3.3*, chapter *3.4.3*)
A. Bosch	2017-06-20	03.01.00	> Added configuration option for blinking MIL (chapter 3.8.1, 3.8.4, 3.8.8 and 5.1.8)
E. Jeglorz	2017-06-21		> Extended chapter *3.1.1 Driving Cycle* and *4.1.1 Driving Cycle* (concerning new API *Dem_GetOperationCycleState()*)
A. Bosch	2017-07-06	03.02.00	> Adapted chapter *3.3.4 Similar Conditions*
			> Extended Table 3-10 NVRAM write frequency
S. Ates	2017-08-23	03.03.00	> Changed description for PID $01 in chapter 3.8.1
M. Heil	2017-08-28		> Document partition mapping restrictions in chapter 7

Reference Documents

No.	Source	Title	Version
[1]	Vector	MICROSAR Diagnostic Event Manager Technical Reference	see delivery
[2]	AUTOSAR	Specification of Diagnostic Event Manager	V4.2.1
[3]	AUTOSAR	Specification of Diagnostic Communication Manager	V4.2.1
[4]	AUTOSAR	Specification of Function Inhibition Manager	V2.2.0
[5]	SAE	J1979 – E/E Diagnostic Test Modes	SEP2010
[6]	ISO	15031-5 Road Vehicles – Communication between vehicle and external equipment for emissions-related diagnostics – Part 5: Emissions-related diagnostic services	-
[7]	CARB	California Code Regulations, Section 1968.2 (OBD II)	2012-08-07
[8]	ISO	14229-1 Road vehicles – Unified diagnostic services (UDS) – Part 1: Specification and requirements	-
[9]	ISO	27145-3 Road Vehicles – Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements – Part 3: Common message dictionary	-

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.

2 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.

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

Legislation	SAE International	ISO
OBDII	J1979 ([5])	ISO-15031-5 ([6])
WWH-OBD	-	ISO-27145-3 ([9]), based on 14229-1([8])

Table 2-1 Standardizations for legislative diagnostic communication

Chapters without one of the icons below handle both OBD II and WWH-OBD.

This symbol marks a chapter or passage which deals with **OBD II**.

WWH-OBD related content is marked with this symbol.

2.1 Comparison of OBD II and WWH-OBD

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

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

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

Note
The Dem can be configured either to support OBD II or WWH-OBD but not to support both at the same time.

3 Functional Characteristics

3.1 Operation Cycles

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

3.1.1 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.

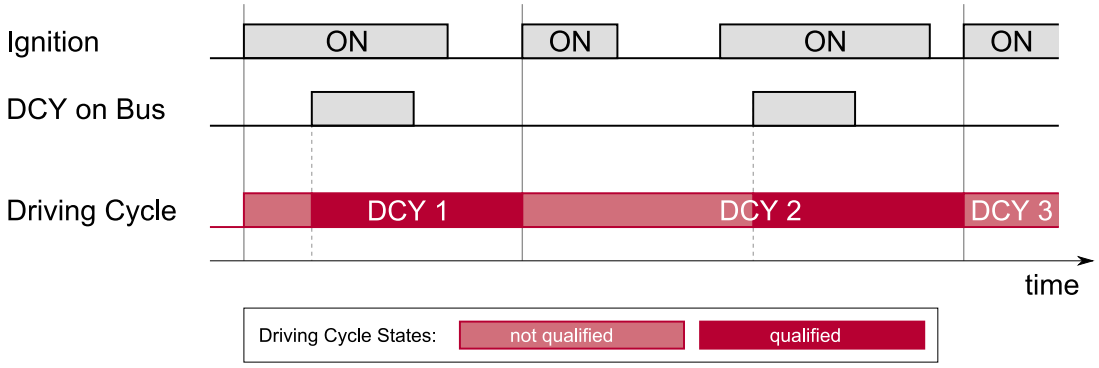

Figure 3-1 OBD Driving Cycle Behavior

3.1.2 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.

3.1.3 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 cannot be used as operation cycle in context of AUTOSAR.

3.1.4 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.

The Dem can be configured to mark the driving cycle as having met the warm-up conditions. Using this configuration option the DTC aging is delayed to the end of the DCY (refer to “WUC Delayed” in Figure 3-2).

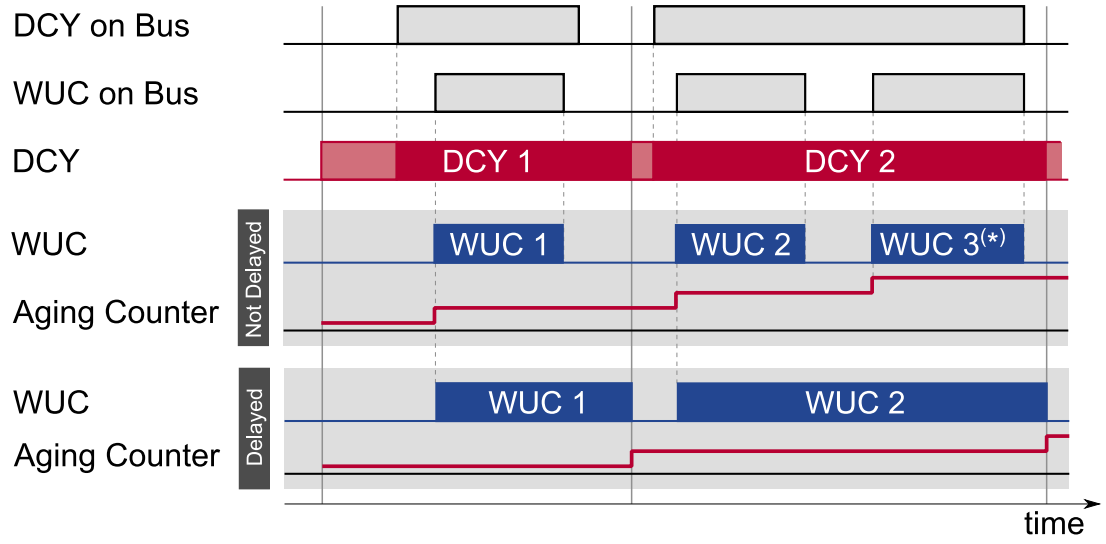

Figure 3-2 Warm-up Cycle Behavior

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.

3.2 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 WWH-OBD uses the UDS protocol for diagnostic communication the 3 byte UDS DTC is used as OBD DTC number.

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

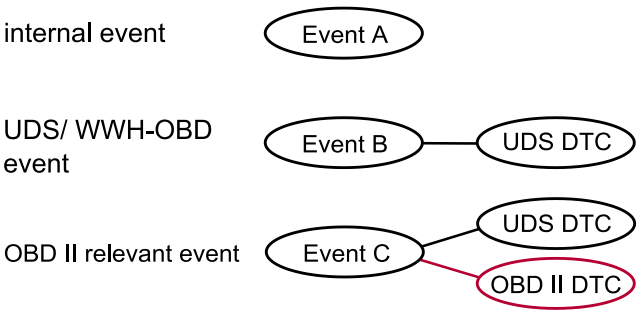

Figure 3-3 Relation between Events and DTCs

Additionally to these basic relations, DTCs and Events can also be grouped using DTC combination and MIL Groups which is described in the following subchapters.

3.2.1 DTC Combination

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.

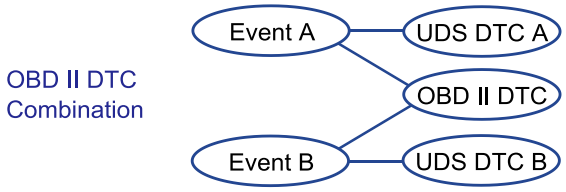

Figure 3-4 Relation between Events and DTCS with DTC combination

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.

3.2.2 MIL Groups

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. This affects only the detection of a failure. Healing and Aging is still performed individually for each event.

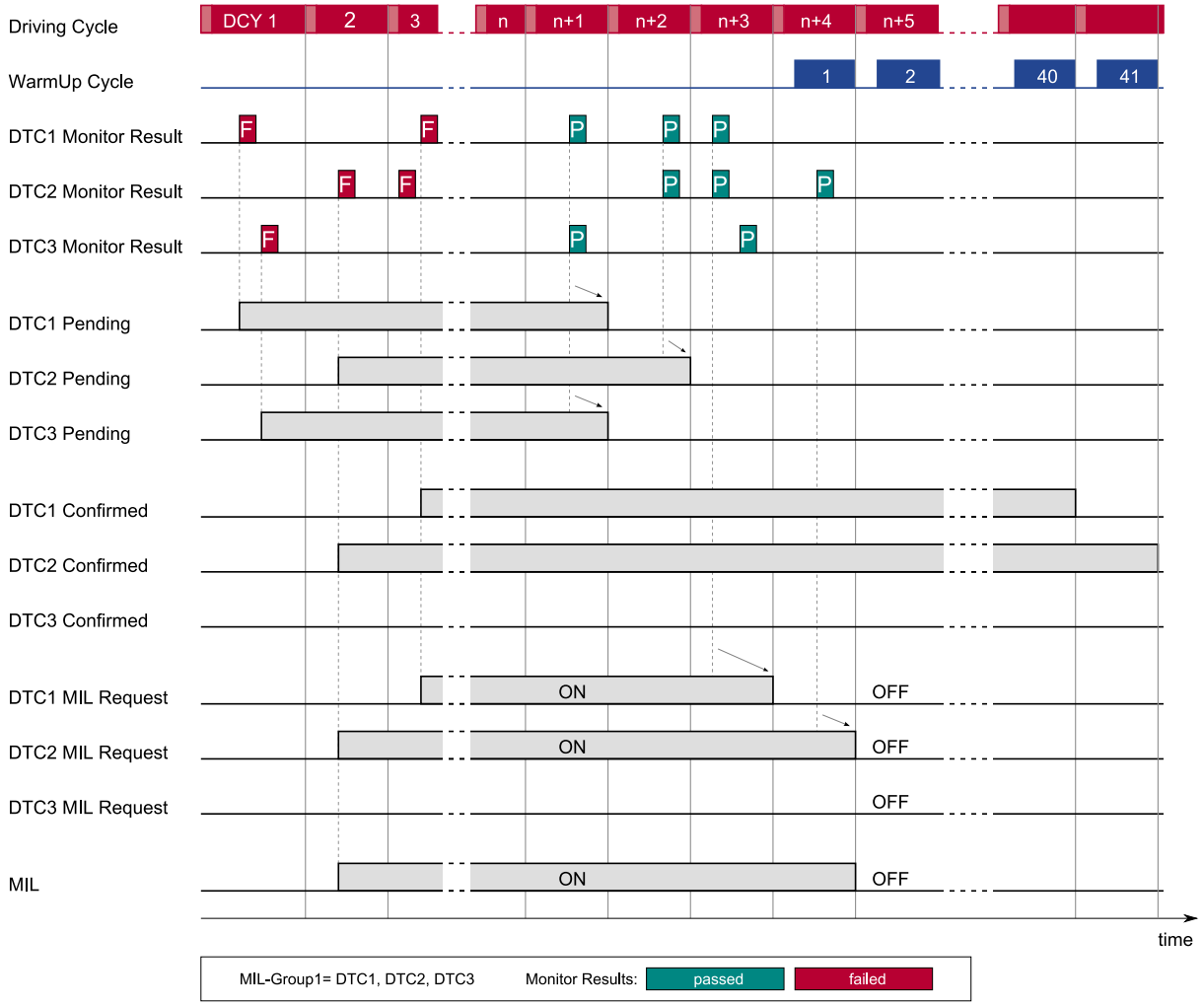

Figure 3-5 MIL Group behavior

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
- If an event is both assigned to a MIL group and to event combination (see description in [1]), the common de-bouncing behavior of the MIL group will be used instead of the individual de-bouncing which is defined for combined events.

3.2.3 DTC Class

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

For WWH-OBD the DTCs are classified dependent on their effect to the emissions if the fault occurs. The DTC class of a fault directly influences the malfunction indicator illumination scheme which can be derived from the activation mode described in chapter 3.6.

DTC Class	Priority	Classification
A	> Class B1 DTCs	If this DTC occurs it is assumed that the OBD threshold limits (OTL) are exceeded.
B1	> Class B2 DTCs	The DTC has the potential to lead to emissions above the OBD threshold limits.
B2	> Class C DTCs	The malfunction is assumed to influence the emissions but not in a level exceeding the OBD threshold limits.
C	> No-Class DTCs	The DTC is assumed to influence the emissions but not in a level above the regulated emission limits.
None	< Class A, B1, B2 and C DTCs	The DTC does not influence the emissions.

Table 3-1 DTC Class

3.2.4 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 3-8.

Diagnostic Service \ Dependent Secondary ECU DTC Availability	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	■		
$14 – ClearDiagnosticInformation	■		

Table 3-2 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 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.

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

3.2.5 Black MIL DTCs

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

Since a black MIL DTC cannot trigger the MIL indicator no permanent DTC will be stored and therefore no respective Mode $0A entry is available.

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.

3.3 Behavior of OBD Relevant Events

3.3.1 General Behavior

The behavior of OBD relevant events differ in some aspects from the behavior specified by UDS, especially concerning event status bit transitions. Figure 3-6 gives an overview of the OBD behavior.

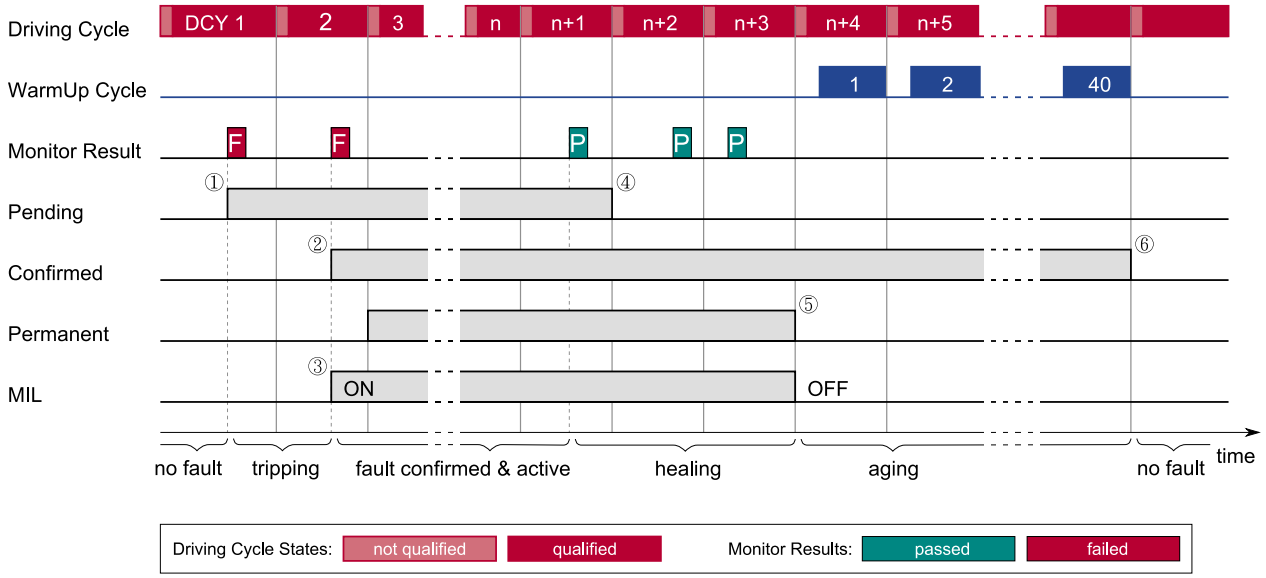

Figure 3-6 OBD failure life cycle

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

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 3.1.4). 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
An OBD DTC is always healed before it starts to age. Typically it takes 3 driving cycles to heal and after that 40 warm-up cycles to age the event.

3.3.2 Failure Confirmation

Figure 3-7 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. But only if the DCY is qualified this status change becomes externally visible, i.e. only then it affects the reporting of OBD DTCs via Mode $03 as well as the reporting of UDS DTCs via Service $19.

An exception to the behavior described above will apply to DTCs which are configured to confirm in the same driving cycle that the malfunction is detected. For those DTCs the status bits are set without the need for a qualified driving cycle.

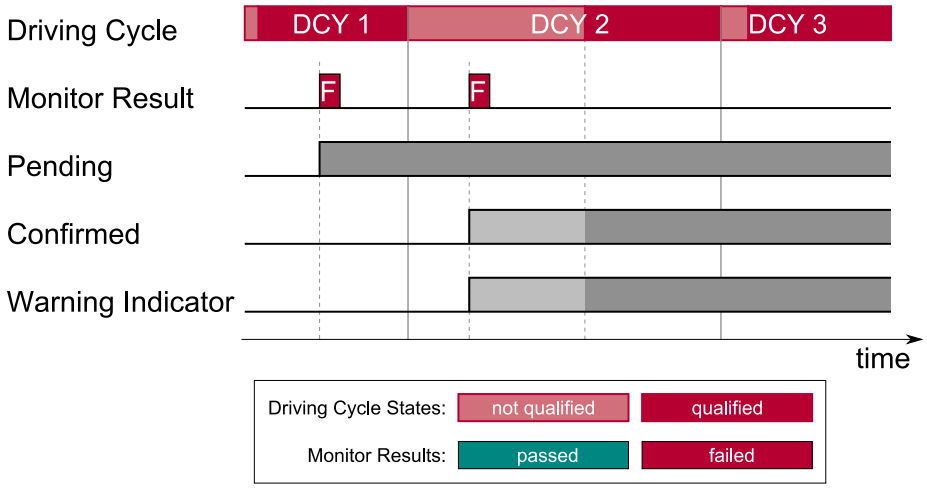

Figure 3-7 Confirmation for OBD relevant events

3.3.3 Event Displacement

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

In addition to the event displacement described in [1] an OBD related event which

- has stored the Mode $02 freeze frame **Or**
- has the pending bit set **Or**
- currently requests the MIL indicator

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 a displaced event has stored the legislative freeze frame, the freeze frame is cleared.

If a displaced event has stored a permanent DTC, the permanent DTC is set to the same state as after a Mode $04 request (see chapter 3.5.4).

3.3.4 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, independent 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 3-8 gives a more detailed view of similar conditions and the impact on the trip count and pending state of an event.

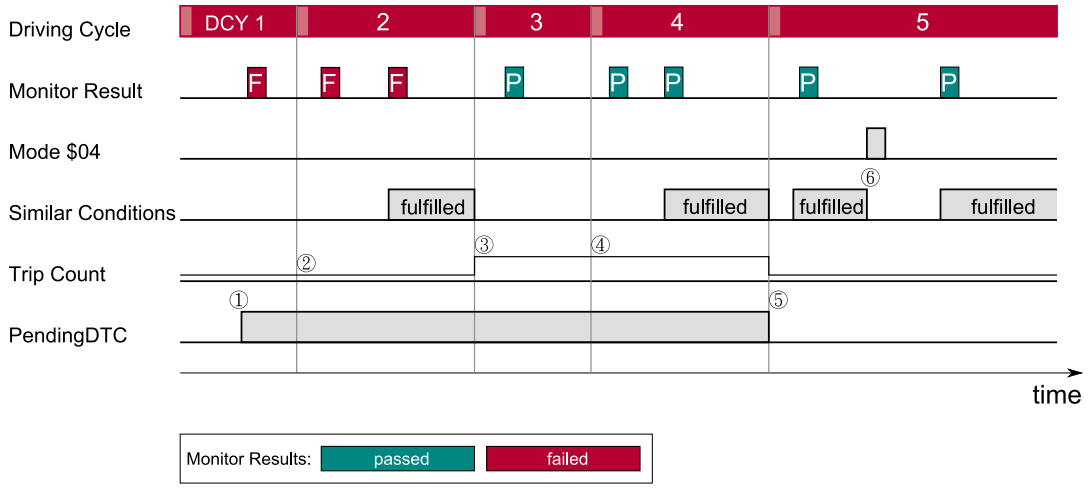

Figure 3-8 Similar conditions

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’.

3.4 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.

3.4.1 Storage of Legislative Freeze Frame for OBD II

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

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.

3.4.2 Storage of Legislative Freeze Frame for WWH-OBD

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

Each OBD related DTC that becomes a pending DTC stores an OBD freeze frame which can be accessed by diagnostic service $19.

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

3.4.3 Reset of Freeze Frame

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

- The event is cleared by a diagnostic service request (e.g. Mode $04, see 3.9.1.4).
- The event is displaced (see chapter 3.3.3).
- If the associated fault has aged.
- If the freeze frame was stored for the pending DTC whereas the DTC is not in confirmed state and starts to age.

3.4.4 Reporting of Freeze Frame with Mode $02

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

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 event is confirmed and the DCY is qualified
- Freeze Frame is stored with confirmed, but Mode $02 reporting is suppressed until the DCY is qualified

3.4.5 Reporting of Freeze Frame with $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.

3.5 Permanent DTCs

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

3.5.1 Definition

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

Permanent DTCs are reported separately from all other DTCs using the OBD specific service Mode $0A (see 3.9.1.7) or the UDS service $19 15.

3.5.2 Storage

Emission related systems must reserve non-volatile memory for failure codes of at least four permanent DTCs. For this, the Dem provides the OBD specific DTC origin type DEM_DTC_ORIGIN_PERMANENT_MEMORY.

Internally, the PDTC is stored immediately as soon as the MIL indicator is commanded, but it is only reported by Mode $0A (see 3.9.1.7) after the driving cycle has been restarted.

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 3-9, where the PDTC of Event1 occupies the last free entry in permanent memory. The storage of Event2 is then delayed until Event1 is healed.

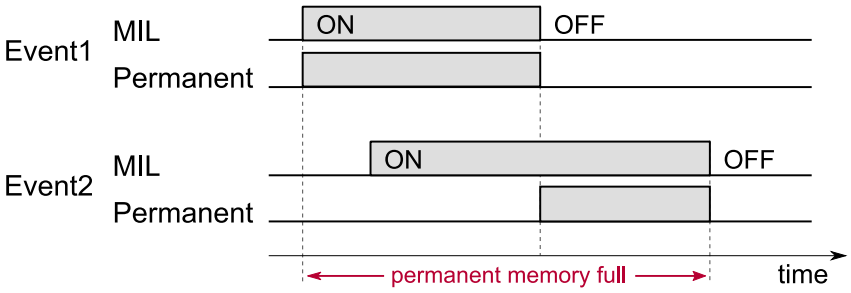

Figure 3-9 Storage of PDTC storage if permanent memory is full

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.

3.5.3 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.

3.5.4 Erasing of Permanent DTCs

Generally, as long as a DTC has permanent status it can't 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) 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 need not 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

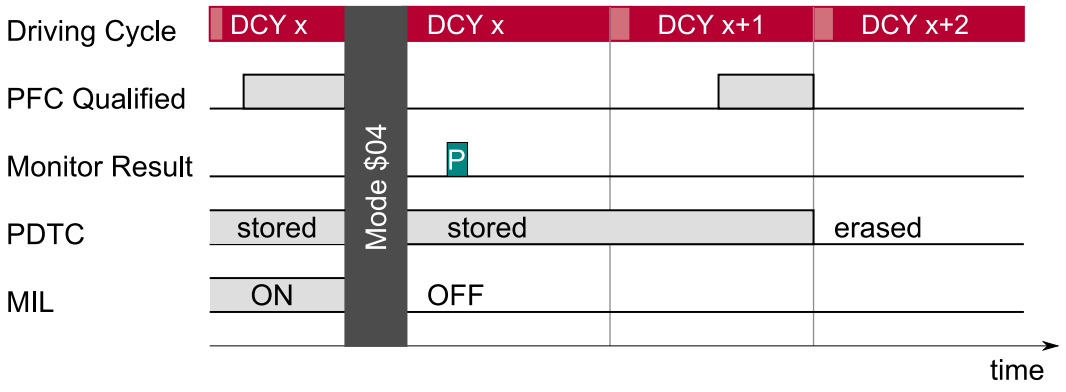

Figure 3-10 Erasing of permanent DTC – failure condition disappeared

Figure 3-10 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 PDTC is erased at the end of the driving cycle, if both conditions have been fulfilled.

Example
Erasing of permanent DTC if failure toggles

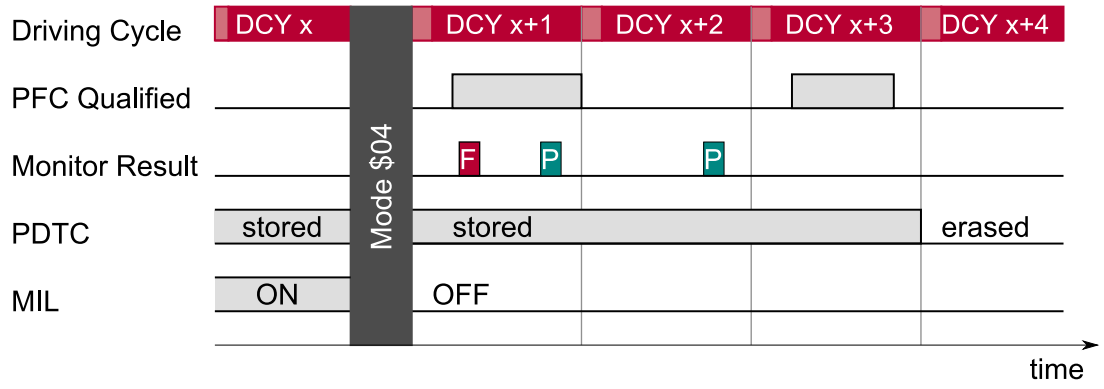

Figure 3-11 Erasing of permanent DTC – failure condition toggles

Figure 3-11 shows the behavior when the event is tested passed but also failed after the clear request. Although the conditions are given, the PDTC 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 PDTC is removed from memory.

Example
Erasing of permanent DTC if failure condition is still present

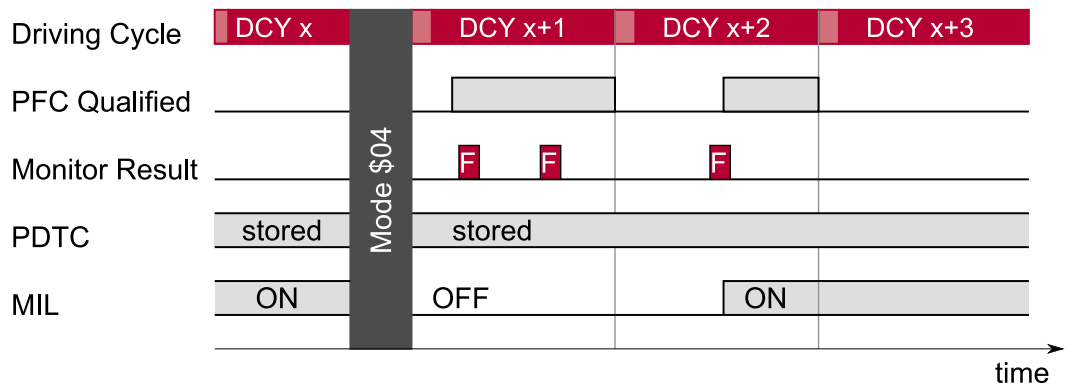

Figure 3-12 Erasing of permanent DTC – failure condition still present

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 3-12, a two-trip configuration is assumed), the clear request is not executed and the PDTC will remain in memory.

To remove in this last example the PDTC 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 PDTC. 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 PDTC is finally erased. Even if the MIL indicator is re-activated by the event, the conditions to remove the PDTC remain the same as shown in Figure 3-10 and Figure 3-11.

3.5.5 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 4.2) 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 can’t be disabled. At the time of activation, all events that are commanding the MIL indicator will be stored as permanent.

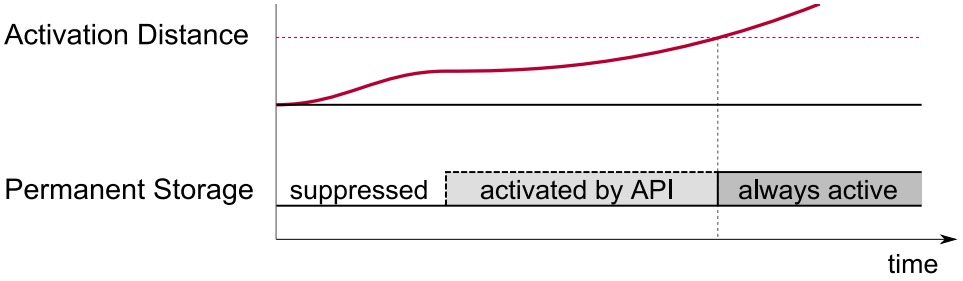

Figure 3-13 Permanent DTC Storage Suppression

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.

3.6 Activation Mode

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 3-3).

Activation Mode	Condition	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 3-3 Activation Mode Conditions

Due to multiple faults with different DTC classes can be active at the same time the activation modes will overlap each other. Activation mode 1 is defined with the lowest priority and activation mode 4 with the highest priority.

3.6.1 Single B1 Counter

The 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 a Class B1 DTC again fulfills the conditions within 3 driving cycles, the counter continuous to increment. Whereas the counter will be set to zero, if no Class B1 DTC fulfills the conditions within 3 driving cycles.

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

Refer to Figure 3-15 for an example of the B1 counter.

3.6.2 Activation Mode Examples

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

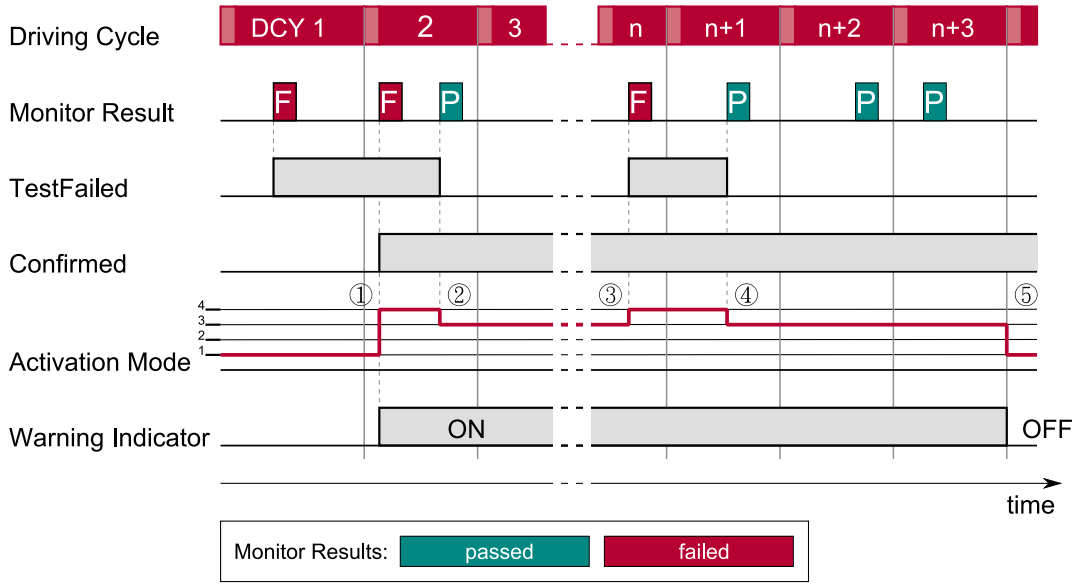

Figure 3-14 Activation Mode example for Class A DTC

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.

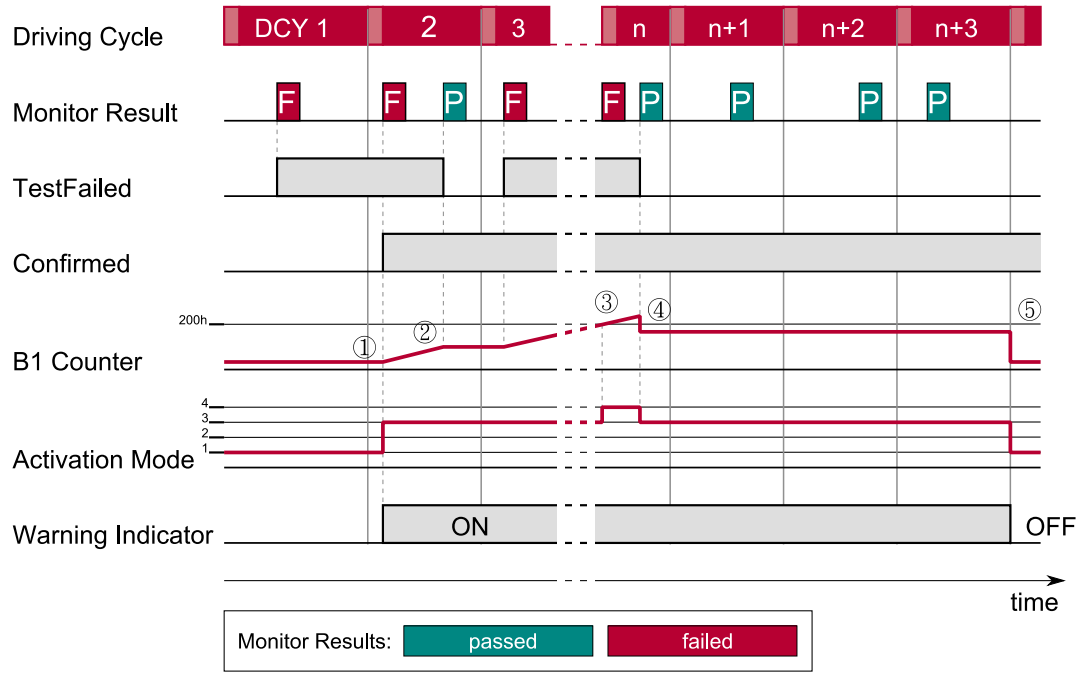

Figure 3-15 Activation Mode example for Class B1 DTC

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 failure-free cycles the activation mode 1 is entered.

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

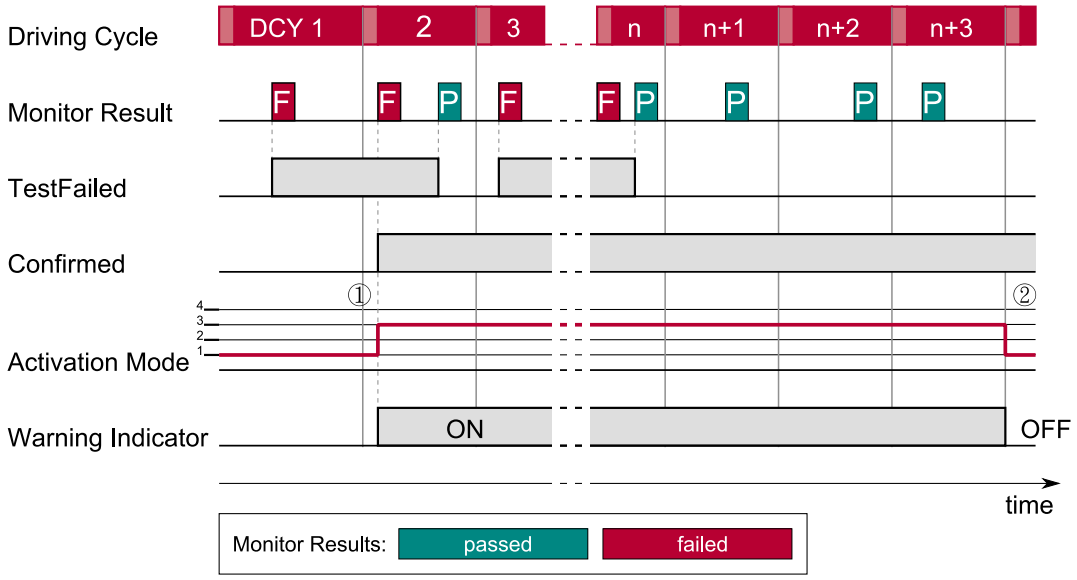

Figure 3-16 Activation Mode example for Class B2 DTC

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.

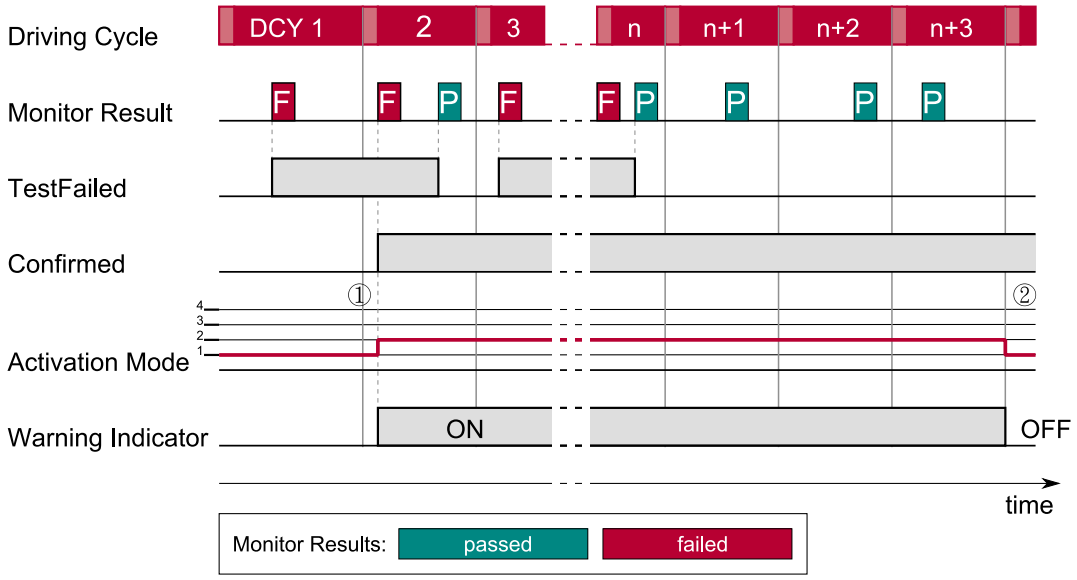

Figure 3-17 Activation Mode example for Class C DTC

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.

3.7 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 3-18 depicts the behavior described above.

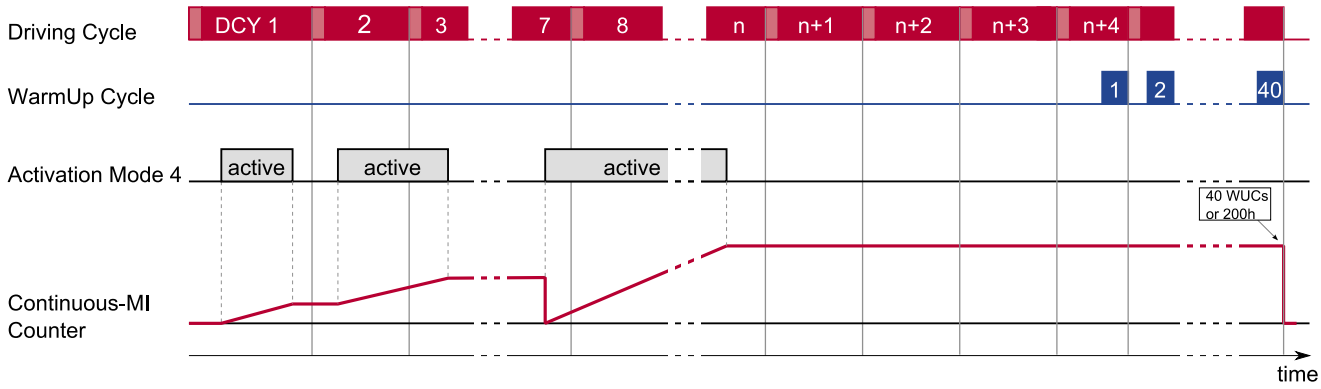

Figure 3-18 Continuous-MI counter

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.

3.8 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 [5].

3.8.1 PID $01: Monitor status since DTCs cleared

The monitor status parameter is a 4 byte value that 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.

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 5.1.8), 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 3-4 PID $01 implementation

- Related API function: Dem_DcmReadDataOfPID01()

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.

A completed monitoring function will only be reset to not completed after DTCs are cleared or an unavailable event assigned to related readiness group is set to available.

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.

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

For the completeness bit calculation only readiness group DEM_OBD_RDY_FLSYS_NONCONT is considered. If no available event is assigned to DEM_OBD_RDY_FLSYS_NONCONT, the monitor function is always reported as completed.

For the supported bit calculation both readiness groups are considered.

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.

3.8.2 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()

3.8.3 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()

3.8.4 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 3-19).

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.

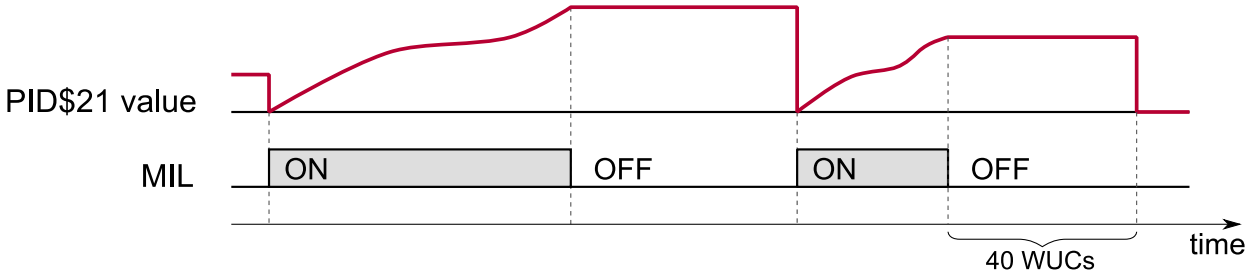

Figure 3-19 Calculation of distance with activated MIL indicator

- Related API function: Dem_DcmReadDataOfPID21()

Note
It is configurable (see configuration parameter “Dem Report Blinking Mil As Active” in chapter 5.1.8), 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.

3.8.5 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.

- Related API function: Dem_DcmReadDataOfPID30()

3.8.6 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()

3.8.7 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. Its status bits may differ to those of PID $01 for non-continuous monitors (monitors that are not executed regularly).

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 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 3-5 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)

Byte	Bit	Comprehensive Component Always Complete	Value	Description
	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 3-6 PID $41 implementation with enabled Major Monitors

- Related API function: Dem_DcmReadDataOfPID41()

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 monitoring function is reset to not completed or N/A if the driving cycle is restarted or if an event assigned to the related readiness group is disabled.

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

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

Therefore for the respective bit calculation the superset of these groups will be handled like a single readiness group.

3.8.8 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 5.1.8), if the Dem considers a blinking (and not continuous) MIL as active for the PID $4D calculation.

3.8.9 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()

3.8.10 PID $91: ECU OBD System Information

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 3-7 PID $91 implementation

- Related API function: Dem_DcmReadDataOfPID91()

3.9 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 [5] and [7]).

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].

3.9.1 OBD II Services

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

3.9.1.1 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.

3.9.1.2 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 3.8.2) is requested, then each PID that is part of the freeze frame can be read.

3.9.1.3 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
A confirmed DTC only becomes externally visible if the driving cycle has already been qualified. If a DTC was confirmed in the current driving cycle, but the driving cycle is not yet qualified when a Mode $03 request is started, the DTC will not be reported. (see also 3.3.2)

3.9.1.4 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 3.5.4).

3.9.1.5 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.

3.9.1.6 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”.

3.9.1.7 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

Because a DTC is only stored as permanent at the end of a driving cycle, it may not be reported by Mode $0A although it already commands the MIL indicator.

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.

3.9.2 UDS and WWH-OBD Services

3.9.2.1 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 3.9.1.4) is allowed.

For Clear DTC with GroupOfDTC set to **0xFFFF33** the following items will be cleared
> all DTCs (WWH-OBD and UDS only)
> activation mode
> Continuous-MI counter
> legislative freeze frame

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

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 feature Multiple OBD Freeze Frame is used and the DTC to be cleared is available in Mode $02 the oldest available Freeze Frame and its DTC become visible in Mode $02.

- If the DTC to be cleared is a **Permanent DTC** the behavior described in chapter 3.5.4 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 3.8.4.
- **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.

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]).

3.9.2.2 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 UDS DTC number of an OBD related event.

3.9.2.3 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.

3.9.2.4 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.

3.9.2.5 Service 19 15: Report DTCs with Permanent Status

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

The behavior of this service is the same as described for Mode $0A (see 3.9.1.7).

3.9.2.6 Service 19 42: Report WWH-OBD DTC by Mask Record

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

o 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”.

3.9.2.7 Service $22: Read Data by Identifier

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

Using the diagnostic service $22 the tester can retrieve the PIDs which are calculated by Dem (e.g. 0xF401 or 0xF491).

3.10 In-Use Monitoring Performance Ratio

The In-Use Monitoring 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.

3.10.1 IUMPR Counters

3.10.1.1 General Denominator

The general denominator is a ratio independent counter that is incremented if the vehicle has been operated long enough under “normal conditions”. These conditions are evaluated by the Master ECU and the result (conditions fulfilled / not fulfilled) is distributed over the bus system.

3.10.1.2 Denominator

To increment a ratio specific denominator counter, at least the general denominator conditions must be fulfilled.

Depending on the monitored functionality, additional conditions may need to be fulfilled before the denominator can be incremented. These must be reported separately to the Dem after the general denominator conditions have been fulfilled. For details how to report the conditions, please refer to chapter 4.7.1.2.

3.10.1.3 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 sufficient 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 4.7.1.3.

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.

3.10.1.4 Ignition Cycle Counter

Basically, the ignition cycle counter represents the number of qualified driving cycles. Only if for the bus signal containing the DCY signal a fault with status pending exist (e.g. temporary CAN message), the counter is not incremented in the current driving cycle.

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.

3.10.2 Counting Behavior

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

General Denominator Conditions	Denominator behavior	Numerator behavior
Inhibited	Locked until end of DCY	Locked until end of DCY
Not reached	Locked until conditions are reached	Incremented when numerator conditions are fulfilled (see 4.7.1.3)
Reached	Incremented when additional denominator conditions are fulfilled (if required; see 4.7.1.2)	Incremented when numerator conditions are fulfilled (see 4.7.1.3)

Table 3-8 Dependency between ratio specific counters and general denominator conditions

Figure 3-20 depicts when the counters are incremented and when not, depending on the related conditions. Please note that the General Conditions and the General Denominator are unique in the ECU, while all other signals are ratio specific.

The ratio specific denominator and numerator are incremented independently from each other as soon as the related conditions are reported as fulfilled (in the example: DCY0, DCY1 and DCY4).

Incrementing is locked if a malfunction has been detected for the inputs of the related monitoring function or of the general denominator. 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 (in the example: DCY2 and DCY3).

For details regarding IUMPR input fault handling, see also chapter 4.7.2.

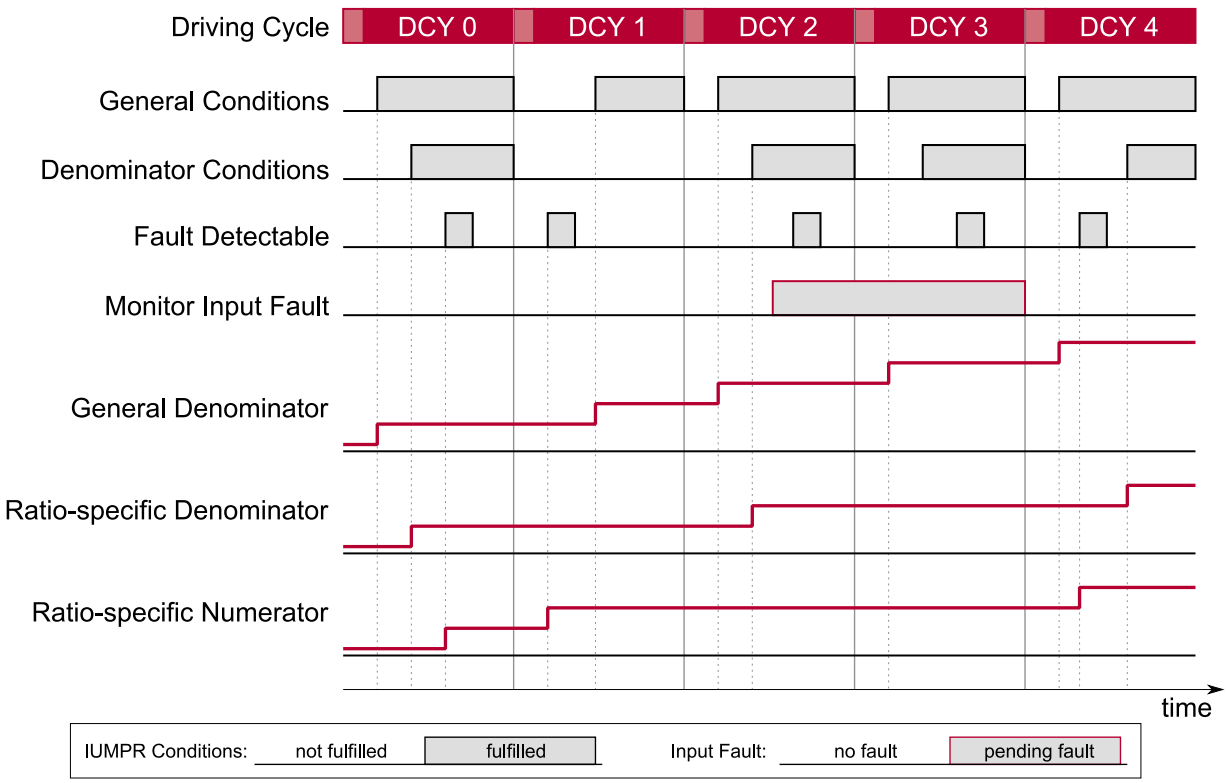

Figure 3-20 IUMPR Counting Behavior

3.10.3 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 4.7.3). Further using suppression by API `Dem_SetEventAvailable()`, the ratio of the event is not tracked in the Dem and also not available for the data reporting.

3.11 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 using the OBD specific Mode $06 (see chapter 3.9.1.5) or the UDS service $22.

3.11.1 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. This can lead to an unavailable MID if all related DTR Ids are ignored (refer to chapter 4.8).

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* or if the assigned event is not available (see *Event Availability* in [1]) 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 3-9.

Event conditions \ Test value	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 3-9 Processing of DTR values

3.11.2 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 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 of the 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 of zero is stored. Likewise 0xFFFF is stored as upper limit if the control value DEM_DTR_CTL_NO_MAX is used.

3.11.3 Resetting the DTR values

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

3.12 Non-Volatile Data Management

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

3.12.1 NVRAM Write Frequency

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

NVRAM Item / Write Frequency	OBDII Freeze Frame Data	OBDII Permanent Data	IUMPR Data	DTR Data
At shutdown – always			■	■
At clear DTC	■	■		
Immediately – automatically if content has changed	■ [1]	■ [2]		
Immediately by API Dem_RequestNvSynchronization – if content has changed	■	■	■	■

Table 3-10 NVRAM write frequency

3.12.2 Data Recovery

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

OBDII

> If an OBD II Freeze Frame is stored for an event but the event has no primary entry, the Freeze Frame is discarded.

> If a permanent DTC is stored and the event supports the MIL indicator but the status bit WarningIndicatorRequested (bit 7) is not set, the permanent DTC is set to the same state than after Mode $04 (refer to 3.5.4).

> If an OBD II DTC currently triggers the status bit WarningIndicatorRequested (bit7) but has no permanent DTC stored, then a permanent entry is created.

[1] Only if an OBD related event becomes pending or confirmed (depends on the configuration of the freeze frame storage trigger) or ages.
[2] Only if a permanent entry is created and if the state of the entry has changed

4 Integration

4.1 Operation Cycle Handling

4.1.1 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 4-1 and Figure 4-1.

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 4-1 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 4-2 OBD Driving Cycle Qualification state

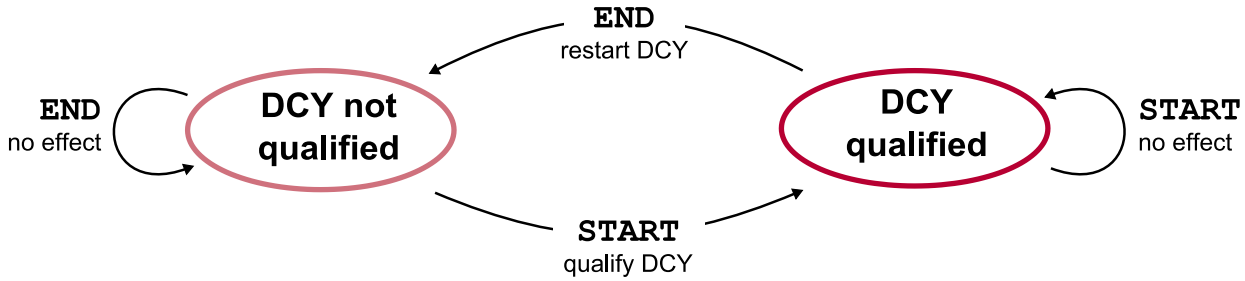

Figure 4-1 OBD Driving Cycle State Transitions

4.1.2 Warm-Up Cycle

If the Dem is configured not to delay the warm-up cycle (see chapter 3.1.4) until the end of a qualified driving cycle, the aging of DTCs using the warm-up cycle as aging cycle (at least all OBD related DTCs) is done as soon as the application starts the warm-up cycle.

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 is done at the end of a qualified driving cycle.

Caution
As the delayed warm-up cycle handling is not according the AUTOSAR definition of the operation cycles the warm-up cycle can’t be used as an operation cycle in context of AUTOSAR.

4.2 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		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		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()`.
PFC		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 3.5.4).
Ignition Cycle Hybrid		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 have to be started by using `Dem_SetOperationCycleState()`. Starting this operation cycle will increment the second ignition cycle counter, which is required for hybrid vehicles.
General Denominator Conditions		Master ECU	The conditions to increment the general denominator (sometimes called “Normed Trip”) are the same as for the PFC signal, but additionally include restrictions for the altitude in which the vehicle is operated. Like the PFC signal, it is calculated by the Master ECU and distributed over the bus system. If the signal have been seen by the application it must be reported to the Dem using the API `Dem_SetIUMPRDenCondition()`. (see 4.7.1.1)

Data	Legislation	Source	Description
Distance Information		Application	This 4 byte value with the travelled distance is needed to calculate PID $21 (3.8.4) and PID $31 (3.8.6) and to determine Permanent Storage Activation (3.5.5). 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		Application	This 4 byte value with the time since engine start is needed to calculate PID $4D (3.8.8) and PID $4E (3.8.9) The callback used by the Dem to request the engine run time is configured with the parameter “Time Since Engine Start”. 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.

Table 4-3 Required data for OBD

4.3 Configuration Aspects

4.3.1 NvM Storage

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

4.3.2 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”.

4.3.3 FiM References

For IUMPR, the Function Inhibition Manager (FiM; see [4]) is required. FiM FIDs are used to lock IUMPR counters if an input fault is detected (see also 4.7.2).

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.
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).

Table 4-4 FIDs required by IUMPR

4.3.4 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 5 for a recommended Dem configuration.

4.4 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
0xFF	Dem_DcmReadDataOfPID91
0x6B	Dem_DcmGetInfoTypeValue08
0x6C	Dem_DcmGetInfoTypeValue0B
0x71	Dem_RepIUMPRDenLock
0x72	Dem_RepIUMPRDenRelease
0x73	Dem_RepIUMPRFaultDetect
0x79	Dem_SetPtoStatus
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

Service ID	Service
0xDA	Dem_IUMPRLockNumerators
0xDD	Dem_SetIUMPRFilter
0xDE	Dem_GetNumberOfFilteredIUMPR
0xDF	Dem_UpdateAvailableOBDMIDs
0xF3	Dem_SetHideObdOccurrences
0xF4	Dem_GetHideObdOccurrences

Table 4-5 OBD specific DET service IDs

4.5 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()`

Exclusive Area 1

No OBD related APIs

Exclusive Area 2

- `Dem_DcmReadDataOfPID01()`
- `Dem_DcmReadDataOfPID21()`
- `Dem_EnablePermanentStorage()`

4.6 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.

4.6.1 NVRAM Demand

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

Table 4-6 OBD NVRAM blocks

4.6.2 NVRAM Initialization

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

Table 4-7 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. This does not apply to the permanent DTC NVRAM block, hence the application has to decide by itself if the permanent data shall be reinitialized or not.

4.7 IUMPR Integration

4.7.1 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 a ratio has been processed, because this would prevent the ratios from updating.

4.7.1.1 General Denominator

In primary ECUs, the information if the general conditions are fulfilled is received from the Master ECU and reported to the Dem by using the API `Dem_SetIUMPRDenCondition()` with ConditionId = DEM_IUMPR_GENERAL_DENOMINATOR (see 7.1.2.10).

If ConditionStatus is DEM_IUMPR_DEN_REACHED, the general denominator is incremented.

If the condition signal can’t be read by the application, it must be reported to the Dem with ConditionStatus DEM_IUMPR_DEN_STATUS_INHIBITED.

4.7.1.2 Denominator

For the ratio specific denominator, the conditions 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 denominator conditions are fulfilled.
DEM_IUMPR_DEN_COLDSTART DEM_IUMPR_DEN_EVAP DEM_IUMPR_DEN_500MI	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 denominator conditions by using `Dem_SetIUMPRDenCondition()` with the corresponding ConditionId.

Denominator Group	Description
DEM_IUMPR_DEN_PHYS_API	As for DEM_IUMPR_DEN_NONE, the denominator is incremented as soon as the general denominator conditions are fulfilled. Additionally, two APIs (`Dem_RepIUMPRDenLock()` and `Dem_RepIUMPRDenRelease()`) are provided to lock the denominator if other, non-standardized conditions need to be fulfilled before the denominator can be 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.

Table 4-8 Triggers to increment the Denominator

4.7.1.3 Numerator

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

RatioId Type	Description
DEM_RATIO_API	The API `Dem_RepIUMPRFaultDetect()` reports that the related monitoring function is able to detect a fault and that the numerator can be incremented. 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 4-9 Triggers to increment the Numerator

4.7.2 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 [4]). This BSW module uses Function Identifiers (FIDs) that are associated to Dem events to determine, if software functionalities need to be disabled.

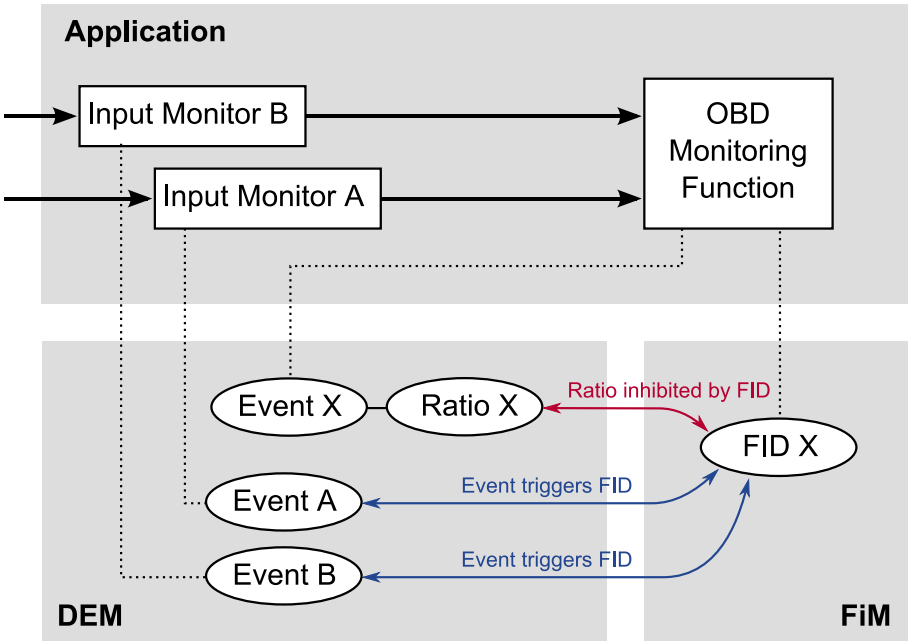

Figure 4-2 Relation between Dem and FiM for IUMPR

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.

Figure 4-2 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. To indicate that a pending fault exists for the DCY signal (and thus for the Ignition Cycle Counter), a FID representing the status of the respective bus signal must be defined and configured (see also 4.3.3).

4.7.3 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 7.1.2.11) starts this sequence, while `Dem_GetNumberOfFilteredIUMPR()` (see 7.1.2.12) will provide the number of ratios matching the given filter criteria. Each subsequent call to `Dem_GetNextIUMPRRatioDataAndDTC()` (see 7.1.2.14) 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 7.1.2.15) can be used.

Figure 4-3 provides an overview of the behavior described above.

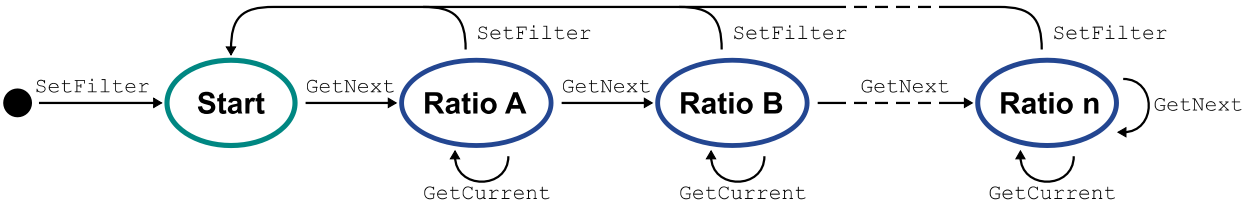

Figure 4-3 Retrieving IUMPR data

Using feature “Event Combination” (for a description, refer to [1]) each combined 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.

4.8 DTR Integration

If several DTR Ids are set invisible an OBD MID can become unavailable during runtime (refer to chapter 3.11.1). To save runtime on the retrieval of the supported OBD MIDs the API `Dem_UpdateAvailableOBDMIDs()` can be called in advance, right after a DTR Id was set invisible. Otherwise availability of the supported OBD MIDs will automatically be updated when the Dcm requests them.

5 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.

5.1 DemGeneral

Dem/DemGeneral		
Configuration Parameter	Legislation	Setting
DemEventMemoryEntryStorageTrigger		**TEST_FAILED**
DemIgnitionCycleFunctionIdRef		the FID that is disabled when the input signals to calculate the ignition cycle are defective (only relevant if IUMPR needs to be supported)
DemImmediateNvStorageSupport		**True**
DemMILIndicatorRef		the respective indicator (refer to chapter 5.1.1) that shall represent the MIL
DemMaxNumberEventEntryOBDFreezeFrame		at least **1**
DemMaxNumberEventEntryPermanent		**4**
DemMultipleTripSupport		**True**
DemOBDSupport		**PRIMARY_ECU**
DemOperationCycleStatusStorage		**True**
DemUseNvm		**True**

Table 5-1 DemGeneral configuration

5.1.1 Malfunction Indicator Lamp

Dem/DemGeneral/DemIndicator		
Configuration Parameter	Legislation	Setting
-		create an indicator element which can be marked as MIL indicator by parameter, refer to Table 5-1

Table 5-2 MIL indicator configuration

5.1.2 PID Configuration

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

Table 5-3 DemPidConfiguration configuration

5.1.3 Driving Cycle

Dem/DemGeneral/DemOperationCycle		
Configuration Parameter	Legislation	Setting
DemOperationCycleType	OBDII WWH-OBD	**OBD_DCY**

Table 5-4 Driving Cycle configuration

5.1.4 Warmup Cycle

	Dem/DemGeneral/DemOperationCycle	
Configuration Parameter	Legislation	Setting
DemOperationCycleType	OBDII WWH-OBD	**WARMUP**

Table 5-5 Warmup Cycle configuration

5.1.5 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 WWH-OBD	**IGNITION_HYBRID**

Table 5-6 Hybrid Ignition Cycle configuration

5.1.6 Data Element for Distance Information

Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass		
Configuration Parameter	Legislation	Setting
DemDataElementDataSize	OBDII WWH-OBD	**4**

Table 5-7 Distance information data element configuration

5.1.7 Data Element for Time Since Engine Start

Dem/DemGeneral/DemDataElementClass/DemExternalCSDataElementClass		
Configuration Parameter	Legislation	Setting
DemDataElementDataSize	OBDII WWH-OBD	**4**

Table 5-8 Time since engine start data element configuration

5.1.8 General OBD

Dem/DemGeneral/DemGeneralOBD		
Configuration Parameter	Legislation	Setting
DemClearPermanentDtcBehavior	OBDII	Project specific
DemDelayWarmUpCycleToDcyEnd	OBDII WWH-OBD	Project specific
DemEngineRuntimePolling	OBDII WWH-OBD	Project specific
DemOBDFreezeFrameBehavior	OBDII WWH-OBD	Project specific
DemOBDFreezeFrameInService19	OBDII	Project specific
DemOBDHideOccurrences	OBDII	Project specific
DemOBDInputDistanceInformation	OBDII WWH-OBD	the data element created for the distance information, refer to chapter 5.1.6

Dem/DemGeneral/DemGeneralOBD		
Configuration Parameter	Legislation	Setting
DemOBDLegislation	OBDII WWH-OBD	Project specific
DemOBDPermanentDtcActivationDistance	OBDII	Project specific
DemOBDPermanentMandatoryPfcAfterClear	OBDII	Project specific
DemOBDRestartDcyOnClearDTC	OBDII WWH-OBD	Project specific
DemOBDTimeSinceEngineStart	OBDII WWH-OBD	The data element created for the time since engine start, refer to chapter 5.1.7
DemOBDSupportMajorMonitors	OBDII WWH-OBD	Project specific
DemOBDPID41ComprehensiveComponentsAlwaysComplete	OBDII WWH-OBD	Project specific
DemOBDSimilarConditionsSupport	OBDII WWH-OBD	Project specific
DemReportBlinkingMilAsActive	OBDII WWH-OBD	Project specific

Table 5-9 DemGeneralOBD configuration

5.2 DemConfigSet

5.2.1 DTC

Dem/DemConfigSet/DemDTCClass		
Configuration Parameter	Legislation	Setting
DemImmediateNvStorage	WWH-OBD	**True**
DemObdDTC	OBDII	The respective DTC number have to be added here
DemWWHOBDDTCClass	WWH-OBD	Depends on the monitor functionality

Table 5-10 DTC configuration

5.2.2 Event

Dem/DemConfigSet/DemEventParameter/DemEventClass		
Configuration Parameter	Legislation	Setting
DemAgingCycleCounterThreshold	OBDII WWH-OBD	**40**
DemAgingCycleRef	OBDII WWH-OBD	The respective operation cycle that is configured as warmup cycle (refer to chapter

Dem/DemConfigSet/DemEventParameter/DemEventClass		
Configuration Parameter	Legislation	Setting
		5.1.4)
DemEventFailureCycleRef	OBDII WWH-OBD	The respective operation cycle that is configured as DCY (refer to chapter 5.1.3)
DemEventFailureCycleCounterThreshold	OBDII WWH-OBD	**1** for US and **2** for EU
DemEventPriority	OBDII WWH-OBD	greater (smaller number) than any UDS only DTCs For WWH-OBD refer to Table 3-1
DemOperationCycleRef	OBDII WWH-OBD	The respective operation cycle that is configured as DCY (refer to chapter 5.1.3)
DemWWHOBDFreezeFrameClassRef	WWH-OBD	The legislative freeze frame that shall be stored for the event.
DemEventOBDReadinessGroup	OBDII WWH-OBD	Project specific

Table 5-11 Event configuration

Dem/DemConfigSet/DemEventParameter/DemEventClass/DemIndicatorAttribute		
Configuration Parameter	Legislation	Setting
DemIndicatorHealingCycleCounterThreshold	OBDII WWH-OBD	**3**
DemIndicatorHealingCycleRef	OBDII WWH-OBD	The respective operation cycle that is configured as DCY (refer to chapter 5.1.3)
DemIndicatorRef	OBDII WWH-OBD	the respective indicator (refer to chapter 5.1.1) that shall represent the MIL

Table 5-12 Event indicator configuration

5.2.3 Legislative Freeze Frame

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

Table 5-13 Legislative freeze frame configuration

5.2.4 DTR

Dem/DemConfigSet/DemDtrs/DemDtr		
Configuration Parameter	Legislation	Setting
DemDtrCompuDenominator0	OBDII WWH-OBD	Project specific

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

Table 5-14 DTR configuration

6 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 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 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).
OBDII WWH-OBD	**MIL Group Affiliation** (refer to 3.2.2) An event can be attached to a MIL Group from a pool of pre-compile configured groups.
OBDII	**Combining OBD DTCs** (refer to 3.2.1) Multiple fault paths can be combined to one single OBD DTC by providing the same DTC number.
OBDII 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 6-1 Post-Build Loadable use cases

7 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.

7.1 Provided Interfaces

7.1.1 DCM

7.1.1.1 Dem_DcmGetInfoTypeValue08()

Prototype	
`Std_ReturnType` **`Dem_DcmGetInfoTypeValue08`** `( Dcm_OpStatusType OpStatus, uint8* Iumprdata08 )`	
Parameter	
`OpStatus`	Parameter is required for interface compatibility to Dcm. Only DCM_INITIAL will appear, because this API behaves synchronous.
`Iumprdata08`	Buffer containing the contents of InfoType $08.
Return code	
`Std_ReturnType`	Always E_OK is returned, as E_PENDING and E_NOT_OK will never appear.
Functional Description	
Service is used to request for IUMPR data according InfoType $08.	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > InfoType $08 is currently not supported; this function will always return E_NOT_OK > This function is only callable from the master partition	

Table 7-1 Dem_DcmGetInfoTypeValue08()

7.1.1.2 Dem_DcmGetInfoTypeValue0B()

Prototype	
`Std_ReturnType` **`Dem_DcmGetInfoTypeValue0B`** `( Dcm_OpStatusType OpStatus, uint8* Iumprdata0B )`	
Parameter	
`OpStatus`	Parameter is required for interface compatibility to Dcm. Only DCM_INITIAL will appear, because this API behaves synchronous.
`Iumprdata0B`	Buffer containing the contents of InfoType $0B.
Return code	
`Std_ReturnType`	Always E_OK is returned, as E_PENDING and E_NOT_OK will never appear.
Functional Description	
Service is used to request for IUMPR data according InfoType $0B.	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > InfoType $0B is currently not supported; this function will always return E_NOT_OK > This function is only callable from the master partition	

Table 7-2 Dem_DcmGetInfoTypeValue0B()

7.1.1.3 Dem_DcmReadDataOfPID01()

Prototype	
`Std_ReturnType` **`Dem_DcmReadDataOfPID01`** `( 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 [5].	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-3 Dem_DcmReadDataOfPID01()

7.1.1.4 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 reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-4 Dem_DcmReadDataOfPID1C()

7.1.1.5 Dem_DcmReadDataOfPID21()

Prototype	
`Std_ReturnType` **`Dem_DcmReadDataOfPID21`** `( uint8* PID21value )`	
Parameter	
`PID21value`	Buffer 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 reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-5 Dem_DcmReadDataOfPID21()

7.1.1.6 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 reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-6 Dem_DcmReadDataOfPID30()

7.1.1.7 Dem_DcmReadDataOfPID31()

Prototype	
`Std_ReturnType` **`Dem_DcmReadDataOfPID31`** `( uint8* PID31value )`	
Parameter	
`PID31value`	Buffer 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 reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-7 Dem_DcmReadDataOfPID31()

7.1.1.8 Dem_DcmReadDataOfPID41()

Prototype	
`Std_ReturnType` **`Dem_DcmReadDataOfPID41`** `( uint8* PID41value )`	
Parameter	
`PID41value`	Buffer 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 [5].	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-8 Dem_DcmReadDataOfPID41()

7.1.1.9 Dem_DcmReadDataOfPID4D()

Prototype	
`Std_ReturnType` **`Dem_DcmReadDataOfPID4D`** `( uint8* PID4Dvalue )`	
Parameter	
`PID4Dvalue`	Buffer 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 reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-9 Dem_DcmReadDataOfPID4D()

7.1.1.10 Dem_DcmReadDataOfPID4E()

Prototype	
`Std_ReturnType` **`Dem_DcmReadDataOfPID4E`** `( uint8* PID4Evalue )`	
Parameter	
`PID4Evalue`	Buffer 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 reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-10 Dem_DcmReadDataOfPID4E()

7.1.1.11 Dem_DcmReadDataOfPID91()

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

Prototype	
`Std_ReturnType` **`Dem_DcmReadDataOfPID91`** `( uint8* PID91value )`	
Parameter	
`PID4Dvalue`	Buffer 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 reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-11 Dem_DcmReadDataOfPID91()

7.1.1.12 Dem_GetDTCSeverityAvailabilityMask()

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

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 7-12 Dem_GetDTCSeverityAvailabilityMask()

7.1.1.13 Dem_DcmReadDataOfOBDFreezeFrame()

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

Prototype	
`Std_ReturnType` **`Dem_DcmReadDataOfOBDFreezeFrame`** `( uint8 PID, uint8 DataElementIndexOfPID, uint8* DestBuffer, uint8* 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.	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-13 Dem_DcmReadDataOfOBDFreezeFrame()

7.1.1.14 Dem_DcmGetDTCOfOBDFreezeFrame()

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

Prototype	
`Std_ReturnType` **`Dem_DcmGetDTCOfOBDFreezeFrame`** `( uint8 FrameNumber, uint32* DTC )`	
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.
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 reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-14 Dem_DcmGetDTCOfOBDFreezeFrame()

7.1.1.15 Dem_DcmGetOBDFreezeFrameData()

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.	
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 7-15 Dem_DcmGetOBDFreezeFrameData()

7.1.1.16 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 7-16 Dem_DcmGetAvailableOBDMIDs()

7.1.1.17 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 7-17 Dem_DcmGetNumTIDsOfOBDMID()

7.1.1.18 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`	TIDindex by which the DTR Id can be identified in combination with the OBDMID.
`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 / TID combination.	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-18 Dem_DcmGetDTRData()

7.1.2 SWC and CDD

7.1.2.1 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 to the Dem for the PID $41 computation.	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-19 Dem_SetEventDisabled()

7.1.2.2 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” or the RatioId/ EventId is not available
Functional Description	
Service for reporting that faults could have been found because of all conditions are fulfilled.	
Particularities and Limitations	
> This function is reentrant. > This function is only callable from the master partition	

Table 7-20 Dem_RepIUMPRFaultDetect()

7.1.2.3 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. > This function is synchronous. > This function is only callable from the master partition	

Table 7-21 Dem_RepIUMPRDenLock()

7.1.2.4 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. > This function is synchronous. > This function is only callable from the master partition	

Table 7-22 Dem_RepIUMPRDenRelease()

7.1.2.5 Dem_SetPtoStatus()

Prototype	
`Std_ReturnType` **`Dem_SetPtoStatus`** `( boolean PtoStatus )`	
Parameter	
`PtoStatus`	Sets the status of the PTO TRUE: active FALSE: inactive
Return code	
`Std_ReturnType`	E_OK: the new PTO-status has been adopted E_NOT_OK: in all other cases.
Functional Description	
Sets the power take-off status.	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition > The PTO status is currently not supported; this function will always return E_NOT_OK	

Table 7-23 Dem_DcmSetPtoStatus()

7.1.2.6 Dem_SetPfcCycleQualified()

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

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 7-24 Dem_SetPfcCycleQualified()

7.1.2.7 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 reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-25 Dem_IUMPRLockNumerators()

7.1.2.8 Dem_EnablePermanentStorage()

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

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 reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-26 Dem_EnablePermanentStorage ()

7.1.2.9 Dem_GetPermanentStorageState()

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

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 7-27 Dem_GetPermanentStorageState ()

7.1.2.10 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 denominator condition 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“
`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 inhibited due to an error
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. > This function is synchronous. > This function is only callable from the master partition	

Table 7-28 Dem_SetIUMPRDenCondition()

7.1.2.11 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 secondary 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 7-29 Dem_SetIUMPRFilter()

7.1.2.12 Dem_GetNumberOfFilteredIUMPR()

Prototype	
`Std_ReturnType` **`Dem_GetNumberOfFilteredIUMPR`** `( uint16* NumberOfFilteredRatios )`	
Parameter	
`NumberOfFiltered Ratios`	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 4.7.3).	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-30 Dem_GetNumberOfFilteredIUMPR()

7.1.2.13 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 7-31 Dem_GetIUMPRGeneralData()

7.1.2.14 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 4.7.3)
Particularities and Limitations
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition

Table 7-32 Dem_GetNextIUMPRRatioDataAndDTC()

7.1.2.15 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 4.7.3).	
Particularities and Limitations	
> This function is not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-33 Dem_GetCurrentIUMPRRatioDataAndDTC()

7.1.2.16 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 not reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-34 Dem_SetHideObdOccurrences()

7.1.2.17 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 7-35 Dem_GetHideObdOccurrences()

7.1.2.18 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 [5].	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-36 Dem_ReadDataOfPID01()

7.1.2.19 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 reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-37 Dem_GetB1Counter()

7.1.2.20 Dem_SetDTR()

Prototype	
`Std_ReturnType Dem_SetDTR ( uint16 DTRId, sint32 TestResult, sint32 LowerLimit, sint32 UpperLimit, Dem_DTRControlType Ctrlval )`	
Parameter	
`DTRId`	Identification of a DTR element by assigned DTR Id.
`TestResult`	Test result to be stored for the DTR Id.
`LowerLimit`	Lower limit to be stored for the DTR Id.
`UpperLimit`	Upper limit to be stored for the DTR Id.
`Ctrlval`	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.
Return code	
`Std_ReturnType`	E_OK: Report of DTR result was successful. E_NOT_OK: Report of DTR result failed.
Functional Description	
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.	
Particularities and Limitations	
> This function is reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-38 Dem_SetDTR()

7.1.2.21 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 reentrant. > This function is synchronous. > This function is only callable from the master partition	

Table 7-39 Dem_UpdateAvailableOBDMIDs()

7.2 Service Ports

7.2.1 Client Server Interface

7.2.1.1 Provided Ports

7.2.1.1.1 OBDServices

Operation	Legislation	API	Port Defined Argument
SetPfcCycleQualified	OBDII	Dem_SetPfcCycleQualified	ERR{E_NOT_OK}
EnablePermanentStorage	OBDII	Dem_EnablePermanentStorage	ERR{E_NOT_OK}
GetPermanentStorageState	OBDII	Dem_GetPermanentStorageState	OUT boolean isEnabled, ERR{E_NOT_OK}
IUMPRLockNumerators	OBDII WWH-OBD	Dem_IUMPRLockNumerators	ERR{E_NOT_OK}
SetHideOccurrences	OBDII WWH-OBD	Dem_SetHideOccurrences	IN boolean DoHide, ERR{E_NOT_OK}
GetHideOccurrences	OBDII WWH-OBD	Dem_GetHideOccurrences	OUT boolean isHidden, ERR{E_NOT_OK}
ReadDataOfPID01	OBDII WWH-OBD	Dem_ReadDataOfPID01	OUT uint8 PID01Value, ERR{E_NOT_OK}
GetB1Counter	WWH-OBD	Dem_GetB1Counter	OUT uint16 B1Counter, ERR{E_NOT_OK}

Table 7-40 Port OBDServices

7.2.1.1.2 IUMPRDenominator

Operation	API	Port Defined Argument
RepIUMPRDenLock	Dem_RepIUMPRDenLock	ERR{E_NOT_OK}
RepIUMPRDenRelease	Dem_RepIUMPRDenRelease	ERR{E_NOT_OK}

Table 7-41 Port IUMPRDenominator

7.2.1.1.3 IUMPRDenominatorCondition

Operation	API	Port Defined Argument
SetIUMPRDenCondition	Dem_SetIUMPRDenCondition	IN Dem_IumprDenomCondStatusType ConditionStatus, ERR{E_NOT_OK}

Table 7-42 Port IUMPRDenominatorCondition

7.2.1.1.4 IUMPRNumerator

Operation	API	Port Defined Argument
RepIUMPRFaultDetect	Dem_RepIUMPRFaultDetect	ERR{E_NOT_OK}

Table 7-43 Port IUMPRNumerator

7.2.1.1.5 IUMPRData

Operation	API	Port Defined Argument
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}
GetCurrentIUMPRRatioData AndDTC	Dem_GetCurrentIUMPRRatio DataAndDTC	OUT uint32 UdsDtcNumber, OUT uint16 ObdDtcNumber, OUT uint16 Denominator, OUT uint16 Numerator, ERR{E_NOT_OK}
GetNextIUMPRRatioData AndDTC	Dem_GetNextIUMPRRatio DataAndDTC	OUT uint32 UdsDtcNumber, OUT uint16 ObdDtcNumber, OUT uint16 Denominator, OUT uint16 Numerator, ERR{E_NOT_OK}

Table 7-44 Port IUMPRData

7.2.1.1.6 DTRCentralReport

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

Table 7-45 Port DTRCentralReport

8 Limitations

8.1 Unsupported Features

This chapter summarizes OBD features that are not or partially supported in the current Dem release:

Feature	Comment
PTO status	The PTO status is not available
InfoType08	IUMPR data is provided via application interface (see 4.7.3)
InfoType0B	IUMPR data is provided via application interface (see 4.7.3)
EventDataChanged Notification	This callback is not triggered for OBD freeze frame storage
Suppression of OBD freeze frame	If the OBD freeze frame shall be suppressed until the causing event becomes qualified confirmed, the suppression only affects 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.
Event specific B1 counter	Only global B1 counter supported
Extended Data Record 0x90	Extended Data Record 0x90 is not supported due to a single B1 counter only supported in Dem.
OBD II DTC without a UDS DTC	Each OBD II DTC must refer at least to one UDS DTC

Table 8-1 Limitations

8.2 Incompatible Features

8.2.1 Aging Behavior

The Dem allows configuring the aging behavior to only count tested cycles toward the aging threshold (see parameter DemGeneral/DemAgingBehavior).

In that case PID $21 is healed much later than expected.

8.2.2 OBD and J1939

Feature OBD and J1939 are not allowed to be enabled at the same time. Therefore disable one of the mentioned features.

8.2.3 OBD II and WWH-OBD

Feature OBD II and WWH-OBD are not allowed to be enabled at the same time.

8.3 Uncommon Feature Behavior

8.3.1 Event Status Changed Callback

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

8.3.2 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.

8.3.3 Event Suppression

PID $01 and PID $41 do not respect event or DTC suppression.

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. Therefore it is discouraged to use event or DTC suppression for OBD related DTCs.

Consider using API `Dem_SetEventAvailable()` instead.

8.3.4 Warm-Up Cycle

If the Dem is configured to delay the warm-up cycle to the end of the driving cycle and a DTC is configured to use the warm-up cycle as operation cycle, the DTC can’t be reported in the very first driving cycle.

8.3.5 Event Combination and IUMPR

If the feature to map the ratio of an event to the environmental data is enabled and the event is a combined event, the numerator and denominator is set to 0 in the environmental data.

9 Glossary and Abbreviations

9.1 Glossary

Term	Description
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.

Table 9-1 Glossary

9.2 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
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
OTL	OBD Threshold Limits
PDTC	Permanent Diagnostic Trouble Code
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
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

Table 9-2 Abbreviations

10 Contact

Visit our website for more information on

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

www.vector.com