

Diagnostic Software Component vDem42

Technical Reference

Version 1.02.00

Authors	Yves Grau, Steffen Köhler
Status	Released

Document Information

History

Author	Date	Version	Remarks
Yves Grau, Steffen Köhler	2017-09-05	1.00.00	Initial version
Steffen Köhler	2017-10-17	1.01.00	Add event status changed callbacks
Yves Grau	2017-03-01	1.02.00	Add (General)DiagnosticInfo interface. Update <i>5.5.1 Client Server Interface</i> .

Reference Documents

No.	Source	Title	Version
[1]	AUTOSAR	AUTOSAR_SWS_DiagnosticEventManager.pdf	4.2.2
[2]	AUTOSAR	AUTOSAR_SWS_DiagnosticEventManager.pdf	4.3.1
[3]	AUTOSAR	AUTOSAR_SWS_DefaultErrorTracer.pdf	4.3.0
[4]	AUTOSAR	AUTOSAR_BasicSoftwareModules.pdf	See delivery
[5]	Vector	TechnicalReference_Diag_Asr4Dem.pdf	See delivery

Scope of the Document

This technical reference describes the general use of the diagnostic software component vDem42.



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.

Contents

1	Component History	6
2	Introduction.....	7
2.1	Architecture Overview	9
3	Functional Description	11
3.1	Features	11
3.2	Error Handling.....	11
3.2.1	Development Error Reporting.....	11
3.2.2	Production Code Error Reporting	11
4	Integration.....	12
4.1	Scope of Delivery.....	12
4.1.1	Static Files	12
4.1.2	Dynamic Files	12
4.2	Integration in Cfg5.....	12
5	API Description.....	13
5.1	Type Definitions	13
5.2	Services provided by vDem42.....	14
5.2.1	VDem42_InitMemory	14
5.2.2	VDem42_Init	14
5.2.3	VDem42_ClearDTC	15
5.2.4	VDem42_GetDTCSuppression	16
5.2.5	VDem42_SetDTCSuppression.....	17
5.2.6	VDem42_GetDTCOfEvent	18
5.2.7	VDem42_GetDebouncingOfEvent.....	19
5.2.8	VDem42_GetEventExtendedDataRecord	20
5.2.9	VDem42_GetEventExtendedDataRecordEx	21
5.2.10	VDem42_GetEventFailed	22
5.2.11	VDem42_GetEventTested.....	23
5.2.12	VDem42_GetEventFreezeFrameData.....	24
5.2.13	VDem42_GetEventFreezeFrameDataEx	25
5.2.14	VDem42_GetEventStatus	26
5.2.15	VDem42_GetFaultDetectionCounter.....	27
5.3	Services used by vDem42	28
5.4	Callback Functions.....	28
5.4.1	VDem42_CBStatusEvt_<Event>_EventStatusChanged	29
5.4.2	VDem42_GlobalCbK_StatusChanged	29

5.5	Service Ports	30
5.5.1	Client Server Interface	30
5.5.1.1	Provide Ports on vDem42 Side	30
5.5.1.1.1	Cdd	30
5.5.1.1.2	ControlDTCSTuppression	30
5.5.1.1.3	GeneralCBStatusEvt43	30
5.5.1.1.4	CBEvtUdsStatusChanged_ <EvtName>_<CallbackName>	30
5.5.1.1.5	GeneralEvtInfo422 / EvtInfo_<EvtName>	31
5.5.1.2	Require Ports on vDem42 Side.....	31
5.5.1.2.1	ClearDTC_vDem42.....	31
5.5.1.2.2	ControlDTCSTuppression_vDem42	31
5.5.1.2.3	GeneralCBStatusEvt422	32
5.5.1.2.4	CBStatusEvt_<EvtName>_ <CallbackName>	32
5.5.1.2.5	GeneralEvtInfo43	32
6	Glossary and Abbreviations	33
6.1	Glossary	33
6.2	Abbreviations	33
7	Contact.....	34

Illustrations

Figure 2-1	System Architecture Overview	7
Figure 2-2	AUTOSAR 4.2 Architecture Overview	9
Figure 2-3	Interfaces to adjacent modules of the vDem42	10

Tables

Table 1-1	Component history.....	6
Table 2-1	Interfaces in scope of vDem42	8
Table 3-1	Supported features	11
Table 4-1	Static files	12
Table 4-2	Generated files	12
Table 5-1	Type definitions.....	13
Table 5-2	VDem42_Init.....	14
Table 5-3	VDem42_ClearDTC.....	15
Table 5-4	VDem42_GetDTCsuppression	16
Table 5-5	VDem42_SetDTCsuppression	17
Table 5-6	VDem42_GetDTCOfEvent.....	18
Table 5-7	VDem42_GetDebouncingOfEvent	19
Table 5-8	VDem42_GetEventExtendedDataRecord	20
Table 5-9	VDem42_GetEventExtendedDataRecordEx	21
Table 5-10	VDem42_GetEventFailed	22
Table 5-11	VDem42_GetEventTested	23
Table 5-12	VDem42_GetEventFreezeFrameData	24
Table 5-13	VDem42_GetEventFreezeFrameDataEx	25
Table 5-14	VDem42_GetEventStatus.....	26
Table 5-15	VDem42_GetFaultDetectionCounter	27
Table 5-16	Services used by the vDem42	28
Table 5-17	VDem42_CBStatusEvt_<Event>_EventStatusChanged	29
Table 5-18	VDem42_GlobalCbK_StatusChanged.....	29
Table 5-19	Cdd.....	30
Table 5-20	ControlDTCsuppression	30
Table 5-21	GeneralCBStatusEvt43.....	30
Table 5-22	CBEventUdsStatusChanged_<EventName>_<CallbackName>	30
Table 5-23	GeneralEvtInfo422 / EvtInfo_<EventName>	31
Table 5-24	ClearDTC_vDem42	31
Table 5-25	ControlDTCsuppression_vDem42.....	31
Table 5-26	GeneralCBStatusEvt422.....	32
Table 5-27	CBStatusEvt_<EventName>_<CallbackName>.....	32
Table 5-28	GeneralEvtInfo43.....	32
Table 6-1	Glossary	33
Table 6-2	Abbreviations.....	33

1 Component History

The component history gives an overview over the important milestones that are supported in the different versions of the component.

Component Version	New Features
1.00.00	Initial Version
1.01.00	Event status changed callbacks added
1.02.00	Add adaptation for (General)DiagnosticInfo interface

Table 1-1 Component history

2 Introduction

This document describes the functionality, API and configuration of the software component vDem42.

Supported AUTOSAR Release*:	4.2.2/4.3	
Supported Configuration Variants:	pre-compile	
Vendor ID:	VDEM42_VENDOR_ID	30 decimal (= Vector-Informatik, according to HIS)
Module ID:	VDEM42_MODULE_ID	255 decimal (according to ref. [4])

* For the detailed functional specification please also refer to the corresponding AUTOSAR SWS.

The vDem42 is an adapter component which translates certain DEM AUTOSAR 4.3 interfaces to AUTOSAR 4.2. The vDem42 is modeled as an AUTOSAR SWC and is part of the application residing above the RTE. Therefore, adaptation is done on port interface level, as shown in Figure 2-1.

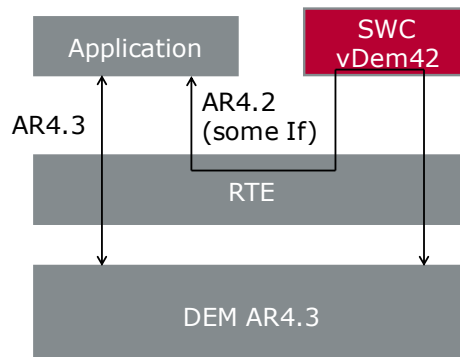


Figure 2-1 System Architecture Overview

Table 2-1 shows all AUTOSAR 4.2.2 DEM client-server interfaces and corresponding operations which are in the scope of vDem42:

C/S-Interface	Operation
Cddl	ClearDTC
DTCSTuppression	SetDTCSTuppression
	GetDTCSTuppression ¹
CallbackEventStatusChange / GeneralCallbackEventStatusChange	EventStatusChanged
DiagnosticInfo / GeneralDiagnosticInfo	GetEventStatus
	GetEventFailed
	GetEventTested
	GetDTCOfEvent
	GetFaultDetectionCounter
	GetEventFreezeFrameData
	GetEventFreezeFrameDataEx
	GetEventExtendedDataRecord
	GetEventExtendedDataRecordEx
	GetDebouncingOfEvent

Table 2-1 Interfaces in scope of vDem42

¹ Not in AUTOSAR 4.2.2 (MICROSAR DEM extension)

2.1 Architecture Overview

Figure 2-2 shows where the vDem42 is located within the AUTOSAR architecture.

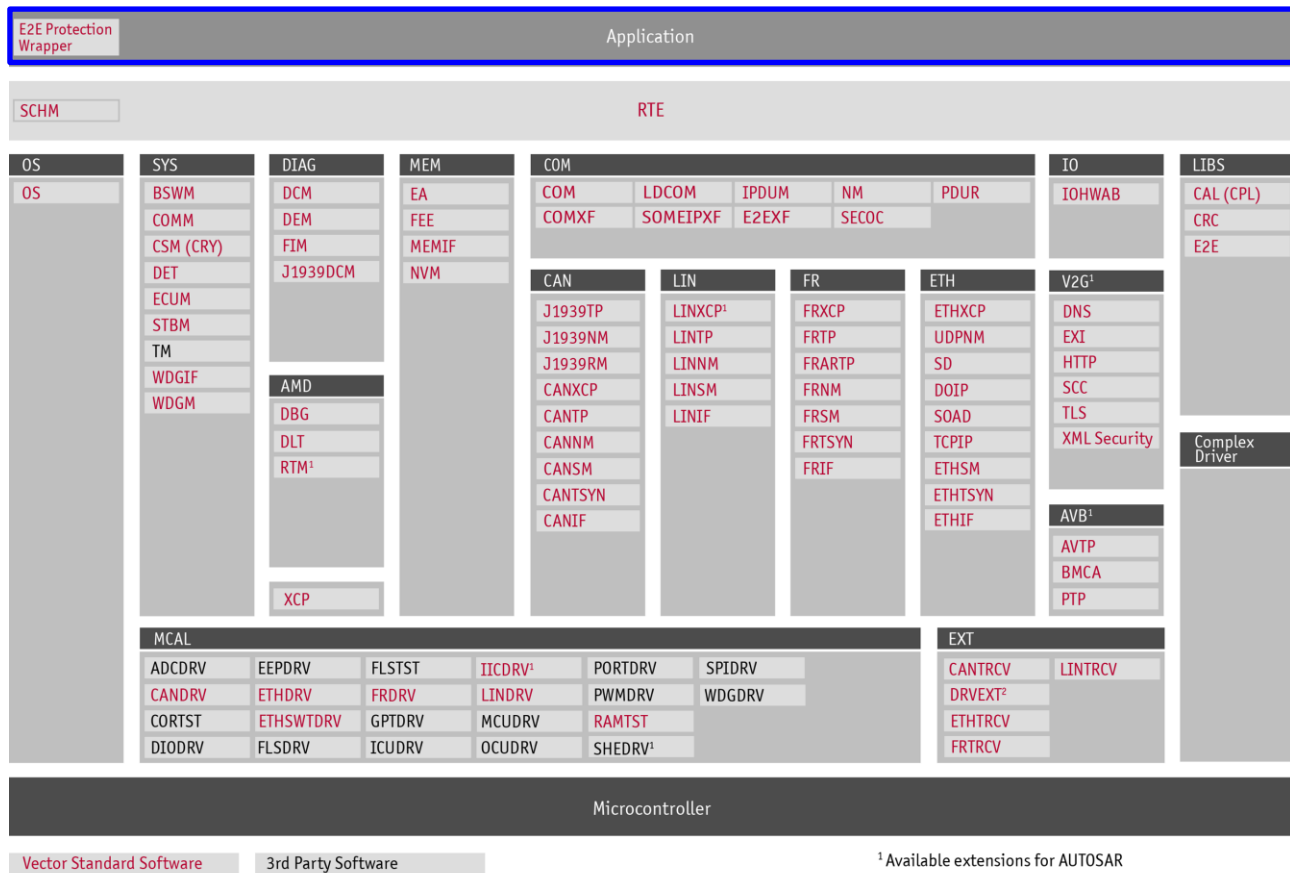


Figure 2-2 AUTOSAR 4.2 Architecture Overview

Figure 2-3 shows the interfaces to adjacent modules of the vDem42. These interfaces are described in chapter 5.

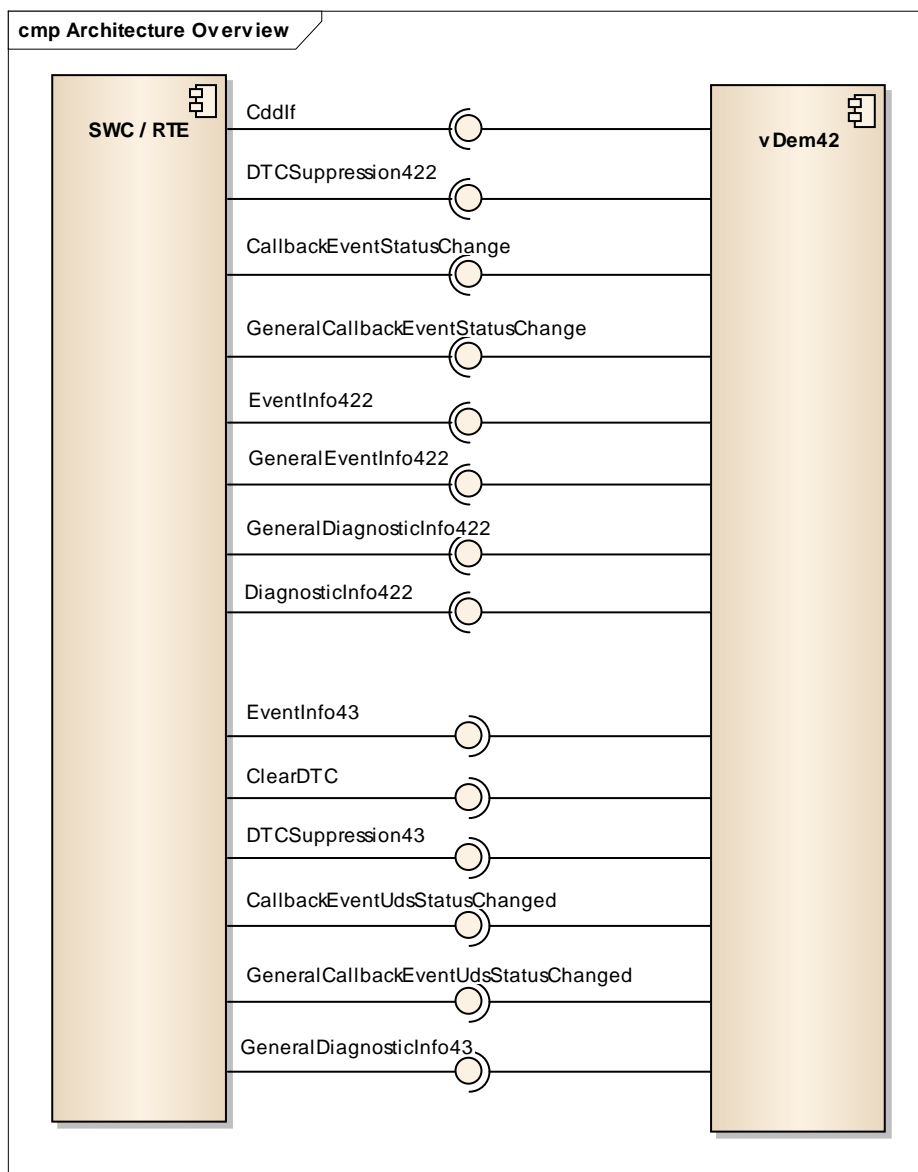


Figure 2-3 Interfaces to adjacent modules of the vDem42

Applications do not access the services of vDem42 directly. They use the service ports provided by vDem42 via the RTE. The service ports provided by the vDem42 are listed in chapter 5.4 and are defined in [1] and [2].

3 Functional Description

3.1 Features

The features listed in Table 3-1 cover the complete functionality specified for the vDem42.

Supported Features
Adapter for AUTOSAR 4.3 DEM interfaces ClearDTC, DTCSuppression, CallbackEventUdsStatusChanged and GeneralCallbackEventUdsStatusChanged to provide corresponding AUTOSAR 4.2.2 conform interfaces.
Adapter for AUTOSAR 4.3 DEM interfaces GeneralDiagnosticInfo and DiagnosticInfo to provide corresponding AUTOSAR 4.2.2 conform interfaces.

Table 3-1 Supported features



Caution

The operations ClearDTC, GetDTCSuppression and SetDTCSuppression must not be called in parallel.

3.2 Error Handling

3.2.1 Development Error Reporting

The vDem42 does not report any development errors to the DET.

3.2.2 Production Code Error Reporting

The vDem42 does not report any production errors to the DEM.

4 Integration

This chapter gives necessary information for the integration of the MICROSAR vDem42 into an application environment of an ECU.

4.1 Scope of Delivery

The delivery of the vDem42 contains the files which are described in the chapters 4.1.1 and 4.1.2:

4.1.1 Static Files

File Name	Description
vDem42.c	This is the source file of the vDem42. It contains the main functionality of the vDem42.
vDem42.h	This header file provides the vDem42 API functions.
vDem42_Type s.h	This header file provides all vDem42 data types. Do not include this file directly – include vDem42.h instead.

Table 4-1 Static files

4.1.2 Dynamic Files

The dynamic files are generated by the configuration tool DaVinci Configurator 5.

File Name	Description
vDem42_Cfg.h	This header file provides the configuration switches of the vDem42.
vDem42_Cfg.c	This source file contains functions of the vDem42.

Table 4-2 Generated files

4.2 Integration in Cfg5

The vDem42 behaves like a software component but needs to be integrated in Cfg5. This is due to the dependency on the configuration of the DEM. For the integration in Cfg5, the vDem42 provides a BSWMD file without configuration elements. Therefore, no separate configuration of the vDem42 is necessary. The vDem42 adapts its behavior according to the configuration of the DEM and generates a software component description via solving action.



Caution

The generation of the software component description is only possible if the DEM does not report validation messages with severity ERROR and code generation is successful.

5 API Description

For an interfaces overview please see Figure 2-3.

5.1 Type Definitions

The types defined by the vDem42 are described in this chapter.

Type Name	C-Type	Description	Value Range
VDem42_ReturnClearDTCType	uint8	Return type for VDem42_ClearDTC function.	E_OK
			DEM_CLEAR_WRONG_DTC
			DEM_CLEAR_WRONG_DTCORIGIN
			DEM_CLEAR_FAILED
			DEM_CLEAR_PENDING
			DEM_CLEAR_BUSY
			DEM_CLEAR_MEMORY_ERROR
VDem42_DTCFormatType	uint8	Type of DTCFormat parameter.	DEM_DTC_FORMAT_OBD
			DEM_DTC_FORMAT_UDS
			DEM_DTC_FORMAT_J1939
VDem42_DTCOriginType	uint8	Type of DTCOrigin parameter.	DEM_DTC_ORIGIN_PRIMARY_MEMORY
			DEM_DTC_ORIGIN_MIRROR_MEMORY
			DEM_DTC_ORIGIN_PERMANENT_MEMORY
			DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY
			DEM_DTC_ORIGIN_SECONDARY_MEMORY
VDem42_DebouncingStateType	uint8	Type of DebouncingState parameter.	DEM_TEMPORARILY_DEFECTIVE
			DEM_FINALLY_DEFECTIVE
			DEM_TEMPORARILY_HEALED
			DEM_TEST_COMPLETE
			DEM_DTR_UPDATE
VDem42_MaxDataValueType	uint8[]	Type of DestBuffer parameter.	-
VDem42_UdsStatusByteType	Uint8	Type of EventStatusByte.	DEM_UDS_STATUS_TF
			DEM_UDS_STATUS_TFTOC
			DEM_UDS_STATUS_PDTC
			DEM_UDS_STATUS_CDTC
			DEM_UDS_STATUS_TNCSLC
			DEM_UDS_STATUS_TFSLC
			DEM_UDS_STATUS_TNCTOC
			DEM_UDS_STATUS_WIR

Table 5-1 Type definitions

5.2 Services provided by vDem42

5.2.1 VDem42_InitMemory

Prototype	
void VDem42_InitMemory (void)	
Parameter	
void	N/A
Return code	
void	N/A
Functional Description	
VDem42_InitMemory initializes static RAM variables in case the start-up code is not used to initialize RAM.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is synchronous. > This function is reentrant. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function shall be called during start-up. 	

5.2.2 VDem42_Init

Prototype	
void VDem42_Init (vDem42_ConfigType* configPtr)	
Parameter	
configPtr	Pointer to a concrete configuration root - not used for now.
Return code	
void	N/A
Functional Description	
VDem42_Init() initializes the vDem42 component.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is synchronous. > This function is reentrant. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function shall be called during start-up. 	

Table 5-2 VDem42_Init

5.2.3 VDem42_ClearDTC

Prototype	
VDem42_ReturnClearDTCType VDem42_ClearDTC (uint32 DTC, VDem42_DTCFormatType DTCFormat, VDem42_DTCOriginType DTCOrigin)	
Parameter	
DTC	The number of the DTC to clear.
DTCFormat	Defines the format of the DTC number.
DTCOrigin	Identifier of the event memory.
Return code	
VDem42_ReturnClearDTCType	<p>E_OK: Clearing is completed for the requested DTC(s).</p> <p>VDEM42_CLEAR_WRONG_DTC: The requested DTC is not valid in the context of DTCFormat and DTCOrigin.</p> <p>VDEM42_CLEAR_WRONG_DTCORIGIN: The requested DTC origin is not available.</p> <p>VDEM42_CLEAR_FAILED: The clear operation could not be completed.</p> <p>VDEM42_CLEAR_MEMORY_ERROR: The clear operation has completed in RAM, but synchronization to Nv-Ram has failed.</p> <p>VDEM42_CLEAR_PENDING: The clear operation is in progress.</p> <p>VDEM42_CLEAR_BUSY: Another clear operation is currently in progress.</p>
Functional Description	
Forwards the ClearDTC memory request to DEM.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is synchronous. > This function is non-reentrant. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function can be called from any context. 	

Table 5-3 VDem42_ClearDTC

5.2.4 VDem42_GetDTCSuppression

Prototype	
Std_ReturnType VDem42_GetDTCSuppression (uint32 DTC, VDem42_DTCFormatType DTCFormat, boolean *SuppressionStatus)	
Parameter	
DTC	The number of the DTC to get suppression status.
DTCFormat	Defines the format of the DTC number.
SuppressionStatus	Output parameter – returns TRUE if DTC is suppressed and FALSE if DTC is not suppressed.
Return code	
Std_ReturnType	E_OK: The request was processed successfully. E_NOT_OK: The function has been called with invalid parameters.
Functional Description	
Forwards the GetDTCSuppression request to DEM.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is synchronous. > This function is non-reentrant. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function can be called from any context. 	

Table 5-4 VDem42_GetDTCSuppression

5.2.5 VDem42_SetDTCSuppression

Prototype	
Std_ReturnType VDem42_SetDTCSuppression (uint32 DTC, VDem42_DTCFormatType DTCFormat, boolean SuppressionStatus)	
Parameter	
DTC	The number of the DTC to suppress.
DTCFormat	Defines the format of the DTC number.
SuppressionStatus	TRUE: Suppress the DTC. FALSE: Lift suppression of the DTC.
Return code	
Std_ReturnType	E_OK: The request was processed successfully. E_NOT_OK: The function has been called with invalid parameters.
Functional Description	
Forwards the SuppressDTC request to DEM.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is synchronous. > This function is non-reentrant. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function can be called from any context. 	

Table 5-5 VDem42_SetDTCSuppression

5.2.6 VDem42_GetDTCOfEvent

Prototype	
Std_ReturnType VDem42_GetDTCOfEvent (VDem42_EventIdType EventId, VDem42_DTCFormatType DTCFormat, uint32* DTCOfEvent)	
Parameter	
EventId	Identification of an event by assigned EventId.
DTCFormat	Defines the output-format of the requested DTC value.
DTCOfEvent	Receives the DTC value in respective format returned by this function. If the return value of the function is not E_OK, this parameter does not contain valid data.
Return code	
Std_ReturnType	E_OK: Operation was successful E_NOT_OK: The call was not successful VDEM42_E_NO_DTC_AVAILABLE: There is no DTC
Functional Description	
Forwards the GetDTCOfEvent request to DEM.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is reentrant (for different EventId). > This function is synchronous. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function can be called from any context. 	

Table 5-6 VDem42_GetDTCOfEvent

5.2.7 VDem42_GetDebouncingOfEvent

Prototype	
Std_ReturnType VDem42_GetDebouncingOfEvent (VDem42_EventIdType EventId, VDem42_DebouncingStateType* DebouncingState)	
Parameter	
EventId	Identification of an event by assigned EventId.
DebouncingState	Debouncing state of event.
Return code	
Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation failed
Functional Description	
Forwards the GetDebouncingOfEvent request to DEM.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is reentrant. > This function is synchronous. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function can be called from any context. 	

Table 5-7 VDem42_GetDebouncingOfEvent

5.2.8 VDem42_GetEventExtendedDataRecord

Prototype	
Std_ReturnType VDem42_GetEventExtendedDataRecord (VDem42_EventIdType EventId, uint8 RecordNumber, uint8* DestBuffer)	
Parameter	
EventId	Identification of an event by assigned EventId.
RecordNumber	Identification of requested Extended data record.
DestBuffer	The pointer to the buffer where the extended data shall be written to.
Return code	
Std_ReturnType	<p>E_OK: Operation was successful; the requested data was copied to the destination buffer.</p> <p>E_NOT_OK: The request was rejected.</p>
Functional Description	
Forwards the GetEventExtendedDataRecord request to DEM.	
Particularities and Limitations	
<p>> This function is reentrant (for different EventId).</p> <p>> This function is synchronous.</p>	
Expected Caller Context	
<p>> This function can be called from any context, with limitations see [5].</p>	

Table 5-8 VDem42_GetEventExtendedDataRecord

5.2.9 VDem42_GetEventExtendedDataRecordEx

Prototype	
Std_ReturnType VDem42_GetEventExtendedDataRecordEx (VDem42_EventIdType EventId, uint8 RecordNumber, uint8* DestBuffer, uint16* BufSize)	
Parameter	
EventId	Identification of an event by assigned EventId.
RecordNumber	Identification of requested extended data record.
DestBuffer	Pointer to the buffer where the extended data shall be written to.
BufSize	When the function is called this parameter must contain the maximum number of data bytes that can be written to the buffer. After successful execution of the operation, the variable is set to the actual number of bytes written to the buffer.
Return code	
Std_ReturnType	<p>E_OK: Operation was successful; the requested data was copied to the destination buffer.</p> <p>E_NOT_OK: The request was rejected.</p> <p>VDEM42_E_WRONG_BUFFERSIZE: The provided destination buffer is too small.</p>
Functional Description	
Forwards the GetEventExtendedDataRecordEx request to DEM.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is reentrant (for different EventId). > This function is synchronous. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function can be called from any context, with limitations see [5]. 	

Table 5-9 VDem42_GetEventExtendedDataRecordEx

5.2.10 VDem42_GetEventFailed

Prototype	
Std_ReturnType VDem42_GetEventFailed (VDem42_EventIdType EventId, Boolean* EventFailed)	
Parameter	
EventId	Identification of an event by assigned EventId.
EventFailed	TRUE – Last Failed FALSE – not Last Failed
Return code	
Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation was not successful
Functional Description	
Forwards the GetEventFailed request to DEM.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is reentrant (for different EventId). > This function is synchronous. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function can be called from any context. 	

Table 5-10 VDem42_GetEventFailed

5.2.11 VDem42_GetEventTested

Prototype	
Std_ReturnType VDem42_GetEventTested (VDem42_EventIdType EventId, Boolean* EventTested)	
Parameter	
EventId	Identification of an event by assigned EventId.
EventTested	TRUE – event tested this cycle FALSE – event not tested this cycle
Return code	
Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation was not successful
Functional Description	
Forwards the GetEventTested request to DEM.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is reentrant (for different EventId). > This function is synchronous. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function can be called from any context. 	

Table 5-11 VDem42_GetEventTested

5.2.12 VDem42_GetEventFreezeFrameData

Prototype	
Std_ReturnType VDem42_GetEventFreezeFrameData (VDem42_EventIdType EventId, uint8 RecordNumber, uint16 DataId, uint8* DestBuffer)	
Parameter	
EventId	Identification of an event by assigned EventId.
RecordNumber	This parameter is a unique identifier for a freeze frame record.
DataId	This parameter specifies the PID or DID that shall be copied to the destination buffer.
DestBuffer	Pointer to the buffer where the freeze frame data shall be written to.
Return code	
Std_ReturnType	E_OK: Operation was successful; the requested data was copied to the destination buffer. E_NOT_OK: The request was rejected.
Functional Description	
Forwards the GetEventFreezeFrameData request to DEM.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is reentrant (for different EventId). > This function is synchronous. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function can be called from any context, with limitations see [5]. 	

Table 5-12 VDem42_GetEventFreezeFrameData

5.2.13 VDem42_GetEventFreezeFrameDataEx

Prototype	
Std_ReturnType VDem42_GetEventFreezeFrameDataEx (VDem42_EventIdType EventId, uint8 RecordNumber, uint16 DataId, uint8* DestBuffer, uint16* BufSize)	
Parameter	
EventId	Identification of an event by assigned EventId.
RecordNumber	This parameter is a unique identifier for a freeze frame record.
DataId	This parameter specifies the PID or DID that shall be copied to the destination buffer.
DestBuffer	Pointer to the buffer where the freeze frame data shall be written to.
BufSize	When the function is called this parameter must contain the maximum number of data bytes that can be written to the buffer. After successful execution of the operation, the variable is set to the actual number of bytes written to the buffer.
Return code	
Std_ReturnType	<p>E_OK: Operation was successful; the requested data was copied to the destination buffer.</p> <p>E_NOT_OK: The request was rejected.</p> <p>VDEM42_E_WRONG_BUFFERSIZE: The provided destination buffer is too small</p>
Functional Description	
Forwards the GetEventFreezeFrameDataEx request to DEM.	
Particularities and Limitations	
<p>> This function is reentrant (for different EventId).</p> <p>> This function is synchronous.</p>	
Expected Caller Context	
<p>> This function can be called from any context, with limitations see [5].</p>	

Table 5-13 VDem42_GetEventFreezeFrameDataEx

5.2.14 VDem42_GetEventStatus

Prototype	
Std_ReturnType VDem42_GetEventStatus (VDem42_EventIdType EventId, VDem42_UdsStatusByteType* UDSStatusByte)	
Parameter	
EventId	Identification of an event by assigned EventId.
UDSStatusByte	UDS event status byte of the requested event. If the return value of the function call is E_NOT_OK, this parameter does not contain valid data.
Return code	
Std_ReturnType	E_OK: Operation was successful E_NOT_OK: Operation was not successful
Functional Description	
Forwards the GetEventStatus request to DEM.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is reentrant (for different EventId). > This function is synchronous. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function can be called from any context. 	

Table 5-14 VDem42_GetEventStatus

5.2.15 VDem42_GetFaultDetectionCounter

Prototype	
Std_ReturnType VDem42_GetFaultDetectionCounter (VDem42_EventIdType EventId, sint8* FaultDetectionCounter)	
Parameter	
EventId	Provide the EventId value the fault detection counter is requested for.
FaultDetectionCounter	This parameter receives the Fault Detection Counter information of the requested EventId. If the return value of the function call is not E_OK this parameter does not contain valid data.
Return code	
Std_ReturnType	E_OK: Request was successful E_NOT_OK: Request failed VDEM42_E_NO_FDC_AVAILABLE: If the event does not support de-bouncing
Functional Description	
Forwards the GetFaultDetectionCounter request to DEM.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is reentrant (for different EventId). > This function is synchronous. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function can be called from any context. 	

Table 5-15 VDem42_GetFaultDetectionCounter

5.3 Services used by vDem42

In the following table services provided by other components, which are used by the vDem42 are listed. For details about prototype and functionality refer to the documentation of the providing component.

Component	API
DEM	Dem_SelectDTC
DEM	Dem_ClearDTC
DEM	Dem_GetDTCSuppression
DEM	Dem_SetDTCSuppression
DEM	Dem_GetDTCOfEvent
DEM	Dem_GetDebouncingOfEvent
DEM	Dem_GetEventExtendedDataRecordEx
DEM	Dem_GetEventFailed
DEM	Dem_GetEventTested
DEM	Dem_GetEventFreezeFrameDataEx
DEM	Dem_GetEventUdsStatus
DEM	Dem_GetFaultDetectionCounter

Table 5-16 Services used by the vDem42

5.4 Callback Functions

This chapter describes the callback functions that are implemented by the vDem42 and are invoked by the DEM. The prototypes of the callback functions are provided in the header file `Rte_vDem42.h` generated by the RTE.

5.4.1 VDem42_CBStatusEvt_<Event>_EventStatusChanged

Prototype	
void VDem42_CBStatusEvt_<Event>_EventStatusChanged (Dem_UdsStatusByteType EventStatusByteOld, Dem_UdsStatusByteType EventStatusByteNew)	
Parameter	
EventStatusByteOld	Event status before the change.
EventStatusByteNew	Event status after the change.
Return code	
void	N/A
Functional Description	
Forward event change callback from the DEM to an application.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is synchronous. > This function is reentrant. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function can be called from any context. 	

Table 5-17 VDem42_CBStatusEvt_<Event>_EventStatusChanged

5.4.2 VDem42_GlobalCbK_StatusChanged

Prototype	
void VDem42_GlobalCbK_StatusChanged (Dem_EventIdType EventId, Dem_UdsStatusByteType EventStatusByteOld, Dem_UdsStatusByteType EventStatusByteNew)	
Parameter	
EventId	ID of the event that has changed.
EventStatusByteOld	Event status before the change.
EventStatusByteNew	Event status after the change.
Return code	
void	N/A
Functional Description	
Forward global event change callback from the DEM to an application.	
Particularities and Limitations	
<ul style="list-style-type: none"> > This function is synchronous. > This function is reentrant. 	
Expected Caller Context	
<ul style="list-style-type: none"> > This function can be called from any context. 	

Table 5-18 VDem42_GlobalCbK_StatusChanged

5.5 Service Ports

5.5.1 Client Server Interface

A client server interface is related to a Provide Port at the server side and a Require Port at client side.

5.5.1.1 Provide Ports on vDem42 Side

At the Provide Ports of the vDem42 the API functions described in 5.2 are available as Runnable Entities. The Runnable Entities are invoked via Operations. The mapping from a SWC client call to an Operation is performed by the RTE. In this mapping, the RTE adds Port Defined Argument Values to the client call of the SWC, if configured.

The following sub-chapters present the Provide Ports defined for the vDem42 and the Operations defined for the Provide Ports, the API functions related to the Operations to be added by the RTE.

5.5.1.1.1 Cdd

Port interface is CddIf.

Operation	API Function
ClearDTC	VDem42_ClearDTC

Table 5-19 Cdd

5.5.1.1.2 ControlDTCSuppression

Port interface is DTCSuppression422.

Operation	API Function
GetDTCSuppression	VDem42_GetDTCSuppression
SetDTCSuppression	VDem42_SetDTCSuppression

Table 5-20 ControlDTCSuppression

5.5.1.1.3 GeneralCBStatusEvt43

Port interface is GeneralCallbackEventUdsStatusChanged.

Operation	API Function
GeneralCallbackEventUdsStatus Changed	VDem42_GeneralCallbackEventUdsStatusChanged

Table 5-21 GeneralCBStatusEvt43

5.5.1.1.4 CBEventUdsStatusChanged_ <EventName>_<CallbackName>

Port interface is CallbackEventUdsStatusChanged.

Operation	ApiFunction
CallbackEventUdsStatusChanged	VDem42_<Event>_<CallbackName>_ CallbackEventUdsStatusChanged

Table 5-22 CBEventUdsStatusChanged_<EventName>_<CallbackName>

5.5.1.1.5 GeneralEvtInfo422 / EvtInfo_<EventName>

Port interface is GeneralDiagnosticInfo422 / DiagnosticInfo422.

Operation	API Function
GetEventStatus	VDem42_GetEventStatus
GetEventFailed	VDem42_GetEventFailed
GetEventTested	VDem42_GetEventTested
GetDTCOfEvent	VDem42_GetDTCOfEvent
GetFaultDetectionCounter	VDem42_GetFaultDetectionCounter
GetEventFreezeFrameData	VDem42_GetEventFreezeFrameData
GetEventFreezeFrameDataEx	VDem42_GetEventFreezeFrameDataEx
GetEventExtendedDataRecord	VDem42_GetEventExtendedDataRecord
GetEventExtendedDataRecordEx	VDem42_GetEventExtendedDataRecordEx
GetDebouncingOfEvent	VDem42_GetDebouncingOfEvent

Table 5-23 GeneralEvtInfo422 / EvtInfo_<EventName>

5.5.1.2 Require Ports on vDem42 Side

At its Require Ports the vDem42 calls Operations. These Operations have to be provided by the SWCs by means of Runnable Entities. These Runnable Entities implement the callback functions expected by the vDem42.

The following sub-chapters present the Require Ports defined for the vDem42, the Operations that are called from the vDem42 and the related Notifications, which are described in chapter 5.3.

5.5.1.2.1 ClearDTC_vDem42

Port interface is ClearDTC.

Operation	Notification
SelectDTC	Rte_Call_ClearDTC_vDem42_SelectDTC
ClearDTC	Rte_Call_ClearDTC_vDem42_ClearDTC

Table 5-24 ClearDTC_vDem42

5.5.1.2.2 ControlDTCSTSuppression_vDem42

Port interface is DTCSTSuppression43.

Operation	Notification
GetDTCSTSuppression	Rte_Call_ControlDTCSTSuppression43_vDem42_GetDTCSTSuppression
SetDTCSTSuppression	Rte_Call_ControlDTCSTSuppression43_vDem42_SetDTCSTSuppression

Table 5-25 ControlDTCSTSuppression_vDem42

5.5.1.2.3 GeneralCBStatusEvt422

Port interface is GeneralCallbackEventStatusChange.

Operation	Notification
EventStatusChanged	Rte_Call_GeneralCBStatusEvt422_EventStatusChanged

Table 5-26 GeneralCBStatusEvt422

5.5.1.2.4 CBStatusEvt_<EventName>_<CallbackName>

Port interface is CallbackEventStatusChange.

Operation	Notification
EventStatusChanged	Rte_Call_CBStatusEvt_<Event>_<CallbackName>_EventStatusChanged

Table 5-27 CBStatusEvt_<EventName>_<CallbackName>

5.5.1.2.5 GeneralEvtInfo43

Port interface is GeneralDiagnosticInfo43.

Operation	Notification
GetEventUdsStatus	Rte_Call_GeneralEvtInfo43_GetEventUdsStatus
GetEventFailed	Rte_Call_GeneralEvtInfo43_GetEventFailed
GetEventTested	Rte_Call_GeneralEvtInfo43_GetEventTested
GetDTCOfEvent	Rte_Call_GeneralEvtInfo43_GetDTCOfEvent
GetFaultDetectionCounter	Rte_Call_GeneralEvtInfo43_GetFaultDetectionCounter
GetEventFreezeFrameDataEx	Rte_Call_GeneralEvtInfo43_GetEventFreezeFrameDataEx
GetEventExtendedDataRecordEx	Rte_Call_GeneralEvtInfo43_GetEventExtendedDataRecordEx
GetDebouncingOfEvent	Rte_Call_GeneralEvtInfo43_GetDebouncingOfEvent

Table 5-28 GeneralEvtInfo43

6 Glossary and Abbreviations

6.1 Glossary

Term	Description
EAD	Embedded Architecture Designer; generation tool for MICROSAR components

Table 6-1 Glossary

6.2 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
DEM	Diagnostic Event Manager
DET	Development Error Tracer
EAD	Embedded Architecture Designer
ECU	Electronic Control Unit
HIS	Hersteller Initiative Software
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)
PPORT	Provide Port
RPORT	Require Port
RTE	Runtime Environment
SWC	Software Component
SWS	Software Specification

Table 6-2 Abbreviations

7 Contact

Visit our website for more information on

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

www.vector.com