

Document Title	Specification of Diagnostic Communication Manager
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	18

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R19-11

Document Change History			
Date	Release	Changed by	Description
2019-11-28	R19-11	AUTOSAR Release Management	<ul style="list-style-type: none"> • Incorporation of Concept NVData Handling Enhancements • Incorporation of PeriodicDID Scheduler Type2 • Renaming of the SRS_Diagnostics to RS_Diagnostics • minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation • Changed Document Status from Final to published

2018-10-31	4.4.0	AUTOSAR Release Management	<ul style="list-style-type: none"> • Incorporation of Concept 636 Security Extensions • Rework of SenderReceiver interface support for DIDs: Atomic SenderReceiver interfaces added. • Rework of SenderReceiver interface support for controlling DIDs via service InputOutputControlByIdentifier (0x2F) • Support added for input signals for the RequestRoutineResults (0x03) subfunction of the RoutineControl (0x31) service • minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2017-12-08	4.3.1	AUTOSAR Release Management	<ul style="list-style-type: none"> • Cleanup SRS_Diagnostic requirement traceability • Fix Dcm/Dem interactions inconsistencies • Add constraints requirements for parameter configuration • minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2016-11-30	4.3.0	AUTOSAR Release Management	<ul style="list-style-type: none"> • Redesign interfaces between Dem and Dcm • Rework Security Access management • Add management for parallel support for OBD and UDS protocols • Clarify usage of Diagnosis scaling • minor corrections / clarifications / editorial changes; For details please refer to the BWCStatement

2015-07-31	4.2.2	AUTOSAR Release Management	<ul style="list-style-type: none"> Specify the NRCs to be sent by the Dcm in case of Dem interfaces return negative values. Clarify Routine operation prototypes Debugging support marked as obsolete Minor corrections / clarifications / editorial changes; For details please refer to the Change Documentation
2014-10-31	4.2.1	AUTOSAR Release Management	<ul style="list-style-type: none"> Update to ISO 14229-1:2013 (Order of NRCs, SID 0x19 and 0x28 extended subfunctions, SID 0x38) Specify security mechanisms (security Lock time, static seed). Refine service ReadDataByPeriodicIdentifier (0x2A) and provide UUDT transfer. Reorganize the configuration parameters for the routines.
2014-03-31	4.1.3	AUTOSAR Release Management	<ul style="list-style-type: none"> Added functional description for DIDRange usage Added support for bootloader interaction Revised the header file structure Editorial changes
2013-10-31	4.1.2	AUTOSAR Release Management	<ul style="list-style-type: none"> Created API tables for service interfaces Provided synchronous and asynchronous APIs for DataServices callouts Harmonization for the length parameter interpretation all over RDBI, WDBI and RC services to be in bytes Editorial changes Removed chapter(s) on change documentation
2013-03-15	4.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> Added Response on Event support Rework configuration for S/R communication Rework OBD Service \$06 management

2011-12-22	4.0.3	AUTOSAR Administration	<ul style="list-style-type: none"> • Change interaction with BswM module for mode management • Change of callout configuration management for services and sub-services processing • Synchronous and asynchronous clarification
2009-12-18	4.0.1	AUTOSAR Administration	<ul style="list-style-type: none"> • ComM_DCM_InactiveDiagnostic and ComM_DCM_ActiveDiagnostic has been defined as mandatory interfaces. • DcmDslPeriodicTxConfirmationPduld multiplicity changed and creation of DcmDslPeriodicConnection parameter in order to link the confirmation Id with TxPdu Id for PeriodicTransmission. • Dem_GetDTCOfOBDFreezeFrame, Dlt_ConditionCheckRead added as optional interfaces • DspInternal_<DiagnosticService> Api moved to mandatory internal interface to support the ECU Supplier diagnosis. • Rework of ReadData operation
2010-02-02	3.1.4	AUTOSAR Administration	<ul style="list-style-type: none"> • Add support of following UDS services : ReadMemoryByAdress, WriteMemoryByAdress, RequestDownload, RequestUpload, TransferData, RequestTransferExit, CommunicationControl, ResponseOnEvent. • Add of bootloader interaction • Add of BswM interaction • Add of IoHwAb interaction • Add of DLT interaction • Add of Signal based approach on RTE interfaces • Legal nvocation revised

2008-08-13	3.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> • Introduction of OBD support • generation of artefacts from the models according to the AUTOSAR process • Identification of requirements and correct formulation of specification items as requirements • General cleanup • Legal invocation revised
2007-12-21	3.0.1	AUTOSAR Administration	<ul style="list-style-type: none"> • Rework of the interfaces with RTE (remove of Central Diagnostic SWC concept) • Correction of issues identified on R2.1 • Document meta information extended • Small layout adaptations made
2007-01-24	2.1.15	AUTOSAR Administration	<ul style="list-style-type: none"> • "Advice for users" revised • "Revision Information" added
	2.1.14	AUTOSAR Administration	<ul style="list-style-type: none"> • Corrections in configuration chapter • Rework on interface between DCM and DEM according to changes in DEM SWS • Corrections in Sequence diagram • Addition of header files inclusions • Legal disclaimer revised
2006-11-28	2.1	AUTOSAR Administration	<ul style="list-style-type: none"> • Layout Adaptations
2006-05-16	2.0	AUTOSAR Administration	<ul style="list-style-type: none"> • Document structure adapted to common Release 2.0 SWS Template • Major changes in chapter 10 • Structure of document changed partly • Other changes see chapter 11
2005-05-31	1.0	AUTOSAR Administration	<ul style="list-style-type: none"> • Initial release

Disclaimer

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1	Introduction and functional overview	22
2	Acronyms and Abbreviations	24
2.1	Typographical Conventions	26
3	Related documentation	26
3.1	Input documents & related standards and norms	26
3.2	Related specification	27
4	Constraints and assumptions	27
4.1	Limitations	27
4.2	Applicability to car domains	29
4.3	Applicability to emission-related environments (OBD)	29
5	Dependencies to other modules	29
6	Requirements Tracing	31
7	Functional specification	45
7.1	Error Classification	45
7.1.1	Development Errors	46
7.1.2	Runtime Errors	46
7.1.3	Transient Faults	46
7.1.4	Production Errors	47
7.1.5	Extended Production Errors	47
7.2	General design elements	47
7.2.1	Submodules within the <code>Dcm</code> module	47
7.2.2	Negative Response Code (NRC)	48
7.2.3	Non-volatile information	48
7.2.4	Types	49
7.2.4.1	Atomic types overview	49
7.2.4.2	Data array types overview	49
7.2.4.3	Data types constraints	50
7.2.4.4	<code>Dcm_OpStatusType</code>	51
7.2.4.5	<code>Dcm_Cemr_{DID}Type</code>	51
7.3	Diagnostic Session Layer (DSL)	51
7.3.1	Introduction	51
7.3.2	Use cases	52
7.3.3	Interaction with other modules	52
7.3.4	Functional description	52
7.3.4.1	Overview	52
7.3.4.2	Forward requests from the <code>PduR</code> module to the <code>DSD</code> submodule	53
7.3.4.2.1	<code>Dcm_StartOfReception</code>	54
7.3.4.2.2	<code>Dcm_CopyRxData</code>	55

7.3.4.2.3	Dcm_TpRxIndication	56
7.3.4.3	Concurrent "TesterPresent" ("keep alive logic")	56
7.3.4.3.1	Dcm_CopyTxData	57
7.3.4.3.2	Dcm_TpTxConfirmation	57
7.3.4.4	Forward responses from the DSD submodule to the PduR module	58
7.3.4.5	Generic Connection Handling	59
7.3.4.6	Guarantee timing to tester by sending busy responses	60
7.3.4.7	Support of periodic transmission	60
7.3.4.8	Support of ROE transmission	61
7.3.4.8.1	ResponseOnEvent StateChar	61
7.3.4.8.1.1	Initializing Dcm (1)	62
7.3.4.8.1.2	Transition from 'ROE cleared' to 'ROE stopped' (2)	62
7.3.4.8.1.3	Transition from 'ROE stopped' to 'ROE cleared' (3)	63
7.3.4.8.1.4	Transition from 'ROE stopped' to 'ROE started' (4)	63
7.3.4.8.1.5	Transition from 'ROE started' to 'ROE stopped' (5)	63
7.3.4.8.1.6	Transition from 'ROE started' to 'ROE started' (6)	64
7.3.4.8.1.7	Transition from 'ROE started' to 'ROE cleared' (7)	64
7.3.4.8.1.8	Transition from 'ROE cleared' to 'ROE cleared' (8)	64
7.3.4.8.1.9	Transition from 'ROE cleared' to 'ROE started' (9)	64
7.3.4.8.1.10	Transition from 'ROE stopped' to 'ROE stopped' (10)	65
7.3.4.8.2	ROE sub-functions	65
7.3.4.8.3	EventWindowTime and StorageState	65
7.3.4.8.4	Pre-configuration of ResponseOnEvent	67
7.3.4.8.5	Handling of event-trigger	68
7.3.4.8.5.1	ROE event-trigger onDTCStatusChange (0x01)	68
7.3.4.8.5.2	ROE event-trigger onChangeOfDataIdentifier (0x03)	69
7.3.4.8.6	Trigger a ServiceToRespondTo	69
7.3.4.8.7	Send a ServiceToRespondTo	70
7.3.4.8.7.1	Roe transmission cycle	71
7.3.4.8.8	ResponseOnEvent in multiple client environments	71
7.3.4.9	Support of segmented response (paged-buffer)	71
7.3.4.10	Support of ResponsePending response triggered by the Application	72

7.3.4.11	Manage security level	72
7.3.4.11.1	Initialization sequence	73
7.3.4.11.2	AttemptCounter update	74
7.3.4.12	Manage session state	75
7.3.4.13	Manage authentication state	75
7.3.4.14	Keep track of active non-default sessions	77
7.3.4.15	Allow to modify timings	78
7.3.4.16	Handling of different diagnostic protocols	79
7.3.4.16.1	Different service tables	79
7.3.4.16.2	Prioritization of protocol	79
7.3.4.16.3	Preemption of protocol	80
7.3.4.16.4	Parallel OBD and UDS protocol processing	81
7.3.4.16.5	Detection of protocol start	82
7.3.4.16.6	Protocol stop	83
7.3.4.17	Manage resources	83
7.3.4.18	Communication Mode Handling	83
7.3.4.18.1	No Communication	84
7.3.4.18.2	Silent Communication	85
7.3.4.18.3	Full Communication	85
7.3.4.18.4	Diagnostic Activation State	86
7.4	Diagnostic Service Dispatcher (DSD)	87
7.4.1	Introduction	87
7.4.2	Use cases	87
7.4.2.1	Receive a request message and transmit a positive response message	87
7.4.2.2	Receive a request message and suppress a positive response	88
7.4.2.3	Receive a request message and suppress a negative response	88
7.4.2.4	Receive a request message and transmit a negative response message	88
7.4.2.5	Send a positive response message without corresponding request	89
7.4.2.6	Segmented Responses (paged-buffer)	89
7.4.3	Interaction of the DSD with other modules	90
7.4.3.1	Interaction of the DSD with the DSL main functionality	90
7.4.3.2	Interaction of the DSD with the DSP	90
7.4.4	Functional Description of the DSD	91
7.4.4.1	Support checking the diagnostic service identifier and adapting the diagnostic message	91
7.4.4.2	Handling of "suppressPosRspMsgIndicationBit"	92
7.4.4.3	Verification functionality	93
7.4.4.3.1	Verification of the diagnostic service access rights	93
7.4.4.3.2	Verification of the Diagnostic Session	95
7.4.4.3.3	Verification of the Service Security Access levels	96

	7.4.4.3.4	Verification of the Service mode dependencies	97
	7.4.4.4	Check format and subfunction support	97
	7.4.4.4.1	Verification of the Manufacturer Application environment/permission	97
	7.4.4.4.2	Verification of the Supplier Application environment/permission	98
	7.4.4.5	Distribution of diagnostic message to DSP submodule	99
	7.4.4.6	Assemble positive or negative response	99
	7.4.4.6.1	Positive Response	99
	7.4.4.6.2	Negative Response	99
	7.4.4.6.3	Suppression of response	100
	7.4.4.7	Initiate transmission	100
7.5		Diagnostic Service Processing (DSP)	101
	7.5.1	General	101
	7.5.1.1	Check format and subfunction support	102
	7.5.1.2	Assemble response	102
	7.5.1.3	Negative Response Codes handling	102
	7.5.1.4	Diagnostic mode declaration groups	103
	7.5.1.5	Environmental condition dependent execution	104
	7.5.1.6	Sender/Receiver Communication	107
	7.5.1.7	Passing SwDataDefProps properties from DEXT file to the Dcm Service SW-C	108
	7.5.1.7.1	DcmDspDiagnosticDataElementRef workflow	108
	7.5.1.7.2	DcmDspAlternativeDataType.DcmApplicationDataType workflow	109
	7.5.1.8	Asynchronous call behavior	109
	7.5.2	UDS Services	110
	7.5.2.1	General behavior using DEM interfaces	111
	7.5.2.2	Service 0x10 - Diagnostic Session Control	113
	7.5.2.3	Service 0x11 - ECUReset	113
	7.5.2.4	Service 0x14 - Clear Diagnostic Information	114
	7.5.2.5	Service 0x19 - Read DTC Information	116
	7.5.2.5.1	Subfunctions 0x01, 0x07, 0x11 and 0x12	116
	7.5.2.5.2	Subfunctions 0x02, 0x0A, 0x0F, 0x13, 0x15 and 0x17	117
	7.5.2.5.3	Subfunction 0x08	120
	7.5.2.5.4	Subfunction 0x09	121
	7.5.2.5.5	Subfunctions 0x06/0x10/0x19	122
	7.5.2.5.6	Subfunction 0x03	124
	7.5.2.5.7	Subfunctions 0x04 and 0x18	126
	7.5.2.5.8	Subfunction 0x05	128
	7.5.2.5.9	Subfunctions 0x0B, 0x0C, 0x0D and 0x0E	129
	7.5.2.5.10	Subfunction 0x14	130
	7.5.2.5.11	Subfunction 0x42	131
	7.5.2.5.12	Subfunction 0x55	133

7.5.2.6	Service 0x22 - ReadDataByIdentifier	134
7.5.2.6.1	UDS DID	137
7.5.2.6.2	OBD DID	138
7.5.2.7	Service 0x24 - ReadScalingDataByIdentifier	140
7.5.2.8	Service 0x27 - SecurityAccess	141
7.5.2.9	Service 0x28 - CommunicationControl	143
7.5.2.10	Service 0x29 - Authentication	146
7.5.2.10.1	De-authentication	148
7.5.2.10.2	Verify Certificates	148
7.5.2.10.3	Proof of ownership client	153
7.5.2.10.4	Definition and verification of roles	155
7.5.2.10.5	Definition and verification of white lists	156
7.5.2.10.6	AuthenticationConfiguration	159
7.5.2.11	Service 0x2A - ReadDataByPeriodicIdentifier	159
7.5.2.11.1	Scheduler PeriodicTransmission	162
7.5.2.12	Service 0x2C - DynamicallyDefineDataIdentifier	171
7.5.2.13	Service 0x2E - WriteDataByIdentifier	173
7.5.2.14	Service 0x2F - InputOutputControlByIdentifier	176
7.5.2.15	Service 0x31 - RoutineControl	187
7.5.2.16	Service 0x3E - Tester Present	194
7.5.2.17	Service 0x3D - WriteMemoryByAddress	194
7.5.2.18	Service 0x23 - ReadMemoryByAddress	196
7.5.2.19	Service 0x34 - RequestDownload	197
7.5.2.20	Service 0x35 - RequestUpload	198
7.5.2.21	Service 0x36 - TransferData	199
7.5.2.22	Service 0x37 - RequestTransferExit	201
7.5.2.23	Service 0x38 - RequestFileTransfer	201
7.5.2.24	Service 0x85 - ControlDTCSetting	203
7.5.2.25	Service 0x87 - LinkControl	204
7.5.3	OBD Services	204
7.5.3.1	Overview	204
7.5.3.2	General behavior	205
7.5.3.3	Service \$01 - Request Current Powertrain Diagnostic Data	206
7.5.3.4	Service \$02 - Request Power Train FreezeFrame Data	208
7.5.3.4.1	Service \$02 - PID\$02	209
7.5.3.4.2	Service \$02 - availability PID	209
7.5.3.4.3	Service \$02 - other PIDs	209
7.5.3.5	Service \$03 \$07 \$0A - Obtaining DTCs	210
7.5.3.6	Service \$04 - Clear/reset emission-related diagnostic information	212
7.5.3.7	Service \$06 - Request On-Board Monitoring Test-results for Specific Monitored Systems	213
7.5.3.7.1	General requirements	213
7.5.3.7.2	Test results obtained via Dem interaction	213

7.5.3.8	Service \$08 - Request Control of On-Board System, Test or Component	214
7.5.3.9	Service \$09 - Request Vehicle Information	215
7.5.4	Interaction usecases	216
7.5.4.1	Jump to Bootloader	217
7.5.4.2	Jump due to ECUReset	219
7.5.4.3	Jump from Bootloader / ECUReset	220
7.5.4.4	Flags management	220
7.5.4.4.1	Jump to Bootloader	220
7.5.4.4.2	Jump from Bootloader	221
7.6	Error notification	221
7.7	Synchronous and Asynchronous implementation	221
7.8	DID configuration	222
7.8.1	Individual DID	222
7.8.2	DID ranges	224
7.9	Startup behavior	225
8	API specification	225
8.1	Imported types	225
8.2	Type definitions	226
8.2.1	Dcm_StatusType	226
8.2.2	Dcm_CommunicationModeType	227
8.2.3	Dcm_ConfigType	228
8.2.4	Dcm_ReturnReadMemoryType	228
8.2.5	Dcm_ReturnWriteMemoryType	229
8.2.6	Dcm_EcuStartModeType	229
8.2.7	Dcm_ProgConditionsType	230
8.2.8	Dcm_MsgItemType	230
8.2.9	Dcm_MsgType	231
8.2.10	Dcm_MsgLenType	231
8.2.11	Dcm_MsgAddInfoType	231
8.2.12	Dcm_IdContextType	232
8.2.13	Dcm_MsgContextType	232
8.2.14	Dcm_ExtendedOpStatusType	234
8.3	Function definitions	234
8.3.1	Functions provided for other BSW components	234
8.3.1.1	Dcm_Init	234
8.3.1.2	Dcm_GetVersionInfo	235
8.3.1.3	Dcm_DemTriggerOnDTCStatus	235
8.3.1.4	Dcm_GetVin	236
8.3.1.5	Dcm_BndMWriteBlockFinish	237
8.3.2	Functions provided to BSW modules and to SW-Cs	237
8.3.2.1	Dcm_SetDeauthenticatedRole	237
8.3.2.2	Dcm_GetSecurityLevel	238
8.3.2.3	Dcm_GetSesCtrlType	238
8.3.2.4	Dcm_GetActiveProtocol	239

8.3.2.5	Dcm_ResetToDefaultSession	239
8.3.2.6	Dcm_TriggerOnEvent	240
8.3.2.7	Dcm_SetActiveDiagnostic	240
8.4	Callback notifications	241
8.4.1	Dcm_StartOfReception	242
8.4.2	Dcm_CopyRxData	243
8.4.3	Dcm_TpRxIndication	244
8.4.4	Dcm_CopyTxData	244
8.4.5	Dcm_TpTxConfirmation	246
8.4.6	Dcm_TxConfirmation	246
8.4.7	Dcm_ComM_NoComModeEntered	247
8.4.8	Dcm_ComM_SilentComModeEntered	247
8.4.9	Dcm_ComM_FullComModeEntered	247
8.4.10	Dcm_CsmAsyncJobFinished	248
8.4.11	Dcm_KeyMAsyncCertificateVerifyFinished	248
8.5	Callout Definitions	249
8.5.1	Dcm_ReadMemory	249
8.5.2	Dcm_WriteMemory	250
8.5.3	Dcm_SetProgConditions	251
8.5.4	Dcm_GetProgConditions	252
8.5.5	Dcm_ProcessRequestAddFile	253
8.5.6	Dcm_ProcessRequestDeleteFile	254
8.5.7	Dcm_ProcessRequestReplaceFile	255
8.5.8	Dcm_ProcessRequestReadFile	256
8.5.9	Dcm_ProcessRequestReadDir	257
8.5.10	Dcm_WriteFile	258
8.5.11	Dcm_ReadFileOrDir	259
8.6	Scheduled functions	259
8.6.1	Dcm_MainFunction	259
8.7	Expected interfaces	260
8.7.1	Mandatory interfaces	260
8.7.2	Optional interfaces	260
8.7.3	Configurable interfaces	264
8.7.3.1	SecurityAccess	264
8.7.3.1.1	GetSeed	264
8.7.3.1.2	CompareKey	266
8.7.3.1.3	GetSecurityAttemptCounter	266
8.7.3.1.4	SetSecurityAttemptCounter	267
8.7.3.2	DataServices	268
8.7.3.2.1	ReadData	268
8.7.3.2.2	WriteData	269
8.7.3.2.3	ReadDataLength	272
8.7.3.2.4	ConditionCheckRead	273
8.7.3.2.5	GetScalingInformation	274
8.7.3.2.6	ReturnControlToECU	275
8.7.3.2.7	ResetToDefault	277

	8.7.3.2.7.1	Synchronous interface	277
	8.7.3.2.7.2	Asynchronous interface	277
	8.7.3.2.8	FreezeCurrentState	278
	8.7.3.2.8.1	Synchronous interface	278
	8.7.3.2.8.2	Asynchronous interface	279
	8.7.3.2.9	ShortTermAdjustment	280
	8.7.3.2.9.1	Synchronous interface	280
	8.7.3.2.9.2	Asynchronous interface	281
8.7.3.3		DataServices_DIDRange	283
	8.7.3.3.1	IsDidAvailable	283
	8.7.3.3.2	ReadDidData	283
	8.7.3.3.3	WriteDidData	284
	8.7.3.3.4	ReadDidRangeDataLength	285
8.7.3.4		InfoTypesServices	285
	8.7.3.4.1	GetInfotypeValueData	285
8.7.3.5		RoutineServices	286
	8.7.3.5.1	Xxx_Start Operation	286
	8.7.3.5.2	Xxx_StartConfirmation Operation	287
	8.7.3.5.3	Xxx_Stop Operation	288
	8.7.3.5.4	Xxx_StopConfirmation Operation	289
	8.7.3.5.5	Xxx_RequestResults Operation	289
	8.7.3.5.6	Xxx_RequestResultsConfirmation Operation	290
8.7.3.6		RequestControlServices	291
	8.7.3.6.1	RequestControl callout	291
8.7.3.7		CallbackDCMRequestServices	292
	8.7.3.7.1	StartProtocol	292
	8.7.3.7.2	StopProtocol	292
8.7.3.8		ServiceRequestNotification	293
	8.7.3.8.1	Indication	293
	8.7.3.8.2	Confirmation	294
8.7.3.9		ClearDTCCheckFnc	295
8.7.3.10		UploadDownloadServices	295
	8.7.3.10.1	ProcessRequestDownload	295
	8.7.3.10.2	ProcessRequestTransferExit	296
	8.7.3.10.3	ProcessRequestUpload	297
	8.7.3.10.4	ProcessTransferDataRead	298
	8.7.3.10.5	ProcessTransferDataWrite	299
8.8		Dcm as Service-Component	300
	8.8.1	Implementation Data Types	300
	8.8.1.1	Dcm_OpStatusType	300
	8.8.1.2	Dcm_ConfirmationStatusType	300
	8.8.1.3	Dcm_SecLevelType	301
	8.8.1.4	Dcm_SesCtrlType	301
	8.8.1.5	Dcm_ProtocolType	302
	8.8.1.6	Dcm_NegativeResponseCodeType	303
	8.8.1.7	Dcm_DataElementType_{Data}Type	306

8.8.1.8	Dcm_DataElementType_{Data}ArrayType	307
8.8.1.9	Dcm_DataElementType_{Data}PrimitiveType	308
8.8.1.10	Dcm_DataArrayTypeUint8_{Data}Type	309
8.8.1.11	{DID}_Struct_DataType	310
8.8.1.12	Dcm_RangeArray_{Range}Type	310
8.8.1.13	Dcm_InfoTypeServicesArray_{VehInfoData}Type	311
8.8.1.14	Dcm_RequestControlServicesInArray_{Tid}Type	311
8.8.1.15	Dcm_RequestControlServicesOutArray_{Tid}Type	311
8.8.1.16	Dcm_ScalingInfoArray_{Data}Type	312
8.8.1.17	Dcm_RequestDataOut_{Routine}_{Signal}PrimitiveType	312
8.8.1.18	Dcm_RequestDataIn_{Routine}_{Signal}PrimitiveType	313
8.8.1.19	Dcm_RequestDataOut_{Routine}_{Signal}Type	314
8.8.1.20	Dcm_RequestDataIn_{Routine}_{Signal}Type	315
8.8.1.21	Dcm_RequestDataOut_{Routine}_{Signal}ArrayType	315
8.8.1.22	Dcm_RequestDataIn_{Routine}_{Signal}ArrayType	316
8.8.1.23	Dcm_RequestFlexibleOutArrayData_{Routine}_{Signal}Type	317
8.8.1.24	Dcm_RequestFlexibleInArrayData_{Routine}_{Signal}Type	317
8.8.1.25	Dcm_StartDataIn_{Routine}_{Signal}PrimitiveType	318
8.8.1.26	Dcm_StartDataIn_{Routine}_{Signal}Type	319
8.8.1.27	Dcm_StartDataIn_{Routine}_{Signal}ArrayType	319
8.8.1.28	Dcm_StartDataOut_{Routine}_{Signal}PrimitiveType	320
8.8.1.29	Dcm_StartDataOut_{Routine}_{Signal}Type	321
8.8.1.30	Dcm_StartDataOut_{Routine}_{Signal}ArrayType	321
8.8.1.31	Dcm_StartFlexibleInArrayData_{Routine}_{Signal}Type	322
8.8.1.32	Dcm_StartFlexibleOutArrayData_{Routine}_{Signal}Type	323
8.8.1.33	Dcm_StopDataIn_{Routine}_{Signal}PrimitiveType	323
8.8.1.34	Dcm_StopDataIn_{Routine}_{Signal}Type	324
8.8.1.35	Dcm_StopDataIn_{Routine}_{Signal}ArrayType	324
8.8.1.36	Dcm_StopDataOut_{Routine}_{Signal}PrimitiveType	325
8.8.1.37	Dcm_StopDataOut_{Routine}_{Signal}Type	326
8.8.1.38	Dcm_StopDataOut_{Routine}_{Signal}ArrayType	326
8.8.1.39	Dcm_StopFlexibleInArrayData_{Routine}_{Signal}Type	327
8.8.1.40	Dcm_StopFlexibleOutArrayData_{Routine}_{Signal}Type	328
8.8.1.41	Dcm_KeyArray_{SecurityLevel}Type	328
8.8.1.42	Dcm_SeedArray_{SecurityLevel}Type	329
8.8.1.43	Dcm_SecurityAccessDataRecordArray_{SecurityLevel}Type	329
8.8.1.44	Dcm_RequestDataArrayType	329
8.8.1.45	Dcm_ControlMask_{DID}Type	330
8.8.1.46	Dcm_inputOutputControlParameterType	330

8.8.1.47	Dcm_IOOperationRequest_{DID}Type	331
8.8.1.48	Dcm_IOOperationResponseType	331
8.8.1.49	Dcm_DidSupportedType	332
8.8.1.50	Dcm_FileAndDirNameType	332
8.8.1.51	Dcm_ResponseDataArrayType	333
8.8.1.52	Dcm_AuthenticationRoleType	333
8.8.1.53	Dcm_ControlMask_{Data}ArrayType	333
8.8.1.54	Dcm_Cemr_{DID}Type	334
8.8.2	Sender-Receiver-Interfaces	335
8.8.2.1	DataServices_{DID}	335
8.8.2.2	DataServices_{Data}	335
8.8.2.3	IOControlRequest_{DID}	336
8.8.2.4	IOControlResponse_{DID}	336
8.8.3	Client-Server-Interfaces	337
8.8.3.1	SecurityAccess_{SecurityLevel}	337
8.8.3.2	DataServices_{Data}	340
8.8.3.2.1	ReadData	356
8.8.3.2.2	WriteData	357
8.8.3.2.3	ReadDataLength	357
8.8.3.2.4	ConditionCheckRead	357
8.8.3.2.5	GetScalingInformation	357
8.8.3.2.6	ReturnControlToEcu	357
8.8.3.2.7	ResetToDefault	357
8.8.3.2.8	FreezeCurrentState	358
8.8.3.2.9	ShortTermAdjustment	358
8.8.3.3	DataServices_DIDRange_{Range}	358
8.8.3.4	InfotypeServices_{VehInfoData}	361
8.8.3.5	RoutineServices_{RoutineName}	362
8.8.3.6	RequestControlServices_{Tid}	376
8.8.3.7	CallbackDCMRequestServices	377
8.8.3.8	ServiceRequestNotification	379
8.8.3.9	UploadDownloadServices	381
8.8.3.10	RequestFileTransfer	387
8.8.3.11	DCMServices	395
8.8.3.12	DCM_Roe	397
8.8.3.13	Authentication	397
8.8.4	NvDataInterface	398
8.8.4.1	DataServices_{DID}	398
8.8.5	Ports	398
8.8.5.1	Dcm_CallbackDCMRequestServices_{Name}	400
8.8.5.2	DataServices_DIDRange_{Range}	400
8.8.5.3	DataServices_{DID}	400
8.8.5.4	DataServices_{Data}	401
8.8.5.5	IOControlRequest_{DID}	403
8.8.5.6	IOControlResponse_{DID}	403
8.8.5.7	DCM_Roe_{RoeName}	403

8.8.5.8	DCMServices	404
8.8.5.9	InfotypeServices_{VehInfoData}	404
8.8.5.10	RequestControlServices_{Tid}	404
8.8.5.11	RequestFileTransfer	405
8.8.5.12	ServiceRequestManufacturerNotification_{Name}	405
8.8.5.13	ServiceRequestSupplierNotification_{Name}	405
8.8.5.14	RoutineServices_{RoutineName}	406
8.8.5.15	SecurityAccess_{SecurityLevel}	406
8.8.5.16	Dcm_DiagnosticSessionControlModeSwitchInterface	406
8.8.5.17	Dcm_EcuResetModeSwitchInterface	407
8.8.5.18	Dcm_ModeRapidPowerShutDownModeSwitchInterface	407
8.8.5.19	Dcm_CommunicationControlModeSwitchInterface_{ComMChannelName}	407
8.8.5.20	Dcm_ControlDTCSettingModeSwitchInterface	408
8.8.5.21	Dcm_ResponseOnEventModeSwitchInterface_{RoeEventID}	408
8.8.5.22	Dcm_SecurityAccessModeSwitchInterface	408
8.8.5.23	Dcm_UploadDownloadServices	409
8.8.5.24	Dcm_Authentication_{Connection}	409
8.8.6	ModeDeclarationGroups	409
8.8.6.1	DcmDiagnosticSessionControl	409
8.8.6.2	DcmEcuReset	410
8.8.6.3	DcmModeRapidPowerShutDown	410
8.8.6.4	DcmCommunicationControl	411
8.8.6.5	DcmControlDTCSetting	411
8.8.6.6	DcmResponseOnEvent	412
8.8.6.7	DcmSecurityAccess	412
8.8.6.8	DcmAuthenticationState	412
8.8.7	Mode-Switch-Interfaces	413
8.8.7.1	Dcm_DiagnosticSessionControlModeSwitchInterface	413
8.8.7.2	Dcm_EcuResetModeSwitchInterface	413
8.8.7.3	Dcm_ModeRapidPowerShutDownModeSwitchInterface	413
8.8.7.4	Dcm_CommunicationControlModeSwitchInterface	414
8.8.7.5	Dcm_ControlDTCSettingModeSwitchInterface	414
8.8.7.6	Dcm_ResponseOnEventModeSwitchInterface	414
8.8.7.7	Dcm_SecurityAccessModeSwitchInterface	415
8.8.7.8	Dcm_AuthenticationStateModeSwitchInterface	415
8.9	External diagnostic service processing	415
8.9.1	<Module>_<DiagnosticService>	416
8.9.2	<Module>_<DiagnosticService>_<SubService>	416
8.10	Internal interfaces (not normative)	417
8.10.1	DslInternal_SetSecurityLevel	417
8.10.2	DslInternal_SetSesCtrlType	417
8.10.3	DsplInternal_DcmConfirmation	417

8.10.4	DslInternal_ResponseOnOneEvent	418
8.10.5	DslInternal_ResponseOnOneDataByPeriodicId	418
8.10.6	DsdInternal_StartPagedProcessing	418
8.10.7	DsplInternal_CancelPagedBufferProcessing	418
8.10.8	DsdInternal_ProcessPage	418
9	Sequence diagrams	419
9.1	Overview	419
9.2	DSL (Diagnostic Session Layer)	419
9.2.1	Start Protocol	419
9.2.2	Process Busy behavior	420
9.2.3	Update Diagnostic Session Control when timeout occurs	421
9.2.4	Process single response of ReadDataByPeriodicIdentifier	422
9.2.5	Process single event-triggered response of ResponseOnEvent	423
9.2.6	Process concurrent requests	424
9.2.7	Interface to ComManager	426
9.2.7.1	Handling in Default Session	426
9.2.7.2	Handling in Non-Default Session	427
9.2.7.3	Session transitions	427
9.2.7.4	Communication States	428
9.2.8	Receive request message and transmit negative response message	430
9.2.9	Process Service Request with paged-buffer	431
9.2.10	Process copy data in reception	433
9.2.11	Process copy data in transmission	434
9.3	DSP (Diagnostic Service Processing)	434
9.3.1	Interface DSP - DEM (service 0x19, 0x14, 0x85)	434
9.3.2	Interface special services	434
9.3.2.1	Process Diagnostic Session Control	434
9.3.2.2	Process Tester Present	435
9.3.2.3	Process Security Access	436
9.3.2.4	Process ResponseOnEvent OnDtcChange	437
9.3.2.5	Process ResponseOnEvent OnChangeOfDataIdentifier	438
9.3.2.6	Process Jump to Bootloader	439
10	Configuration specification	439
10.1	How to read this chapter	440
10.2	Containers and configuration parameters	440
10.2.1	Dcm	441
10.2.2	DcmConfigSet	442
10.2.2.1	DcmPageBufferCfg	442
10.2.2.2	DcmProcessingConditions	443
10.2.3	DcmDsd	446
10.2.3.1	DcmDsd	446
10.2.3.2	DcmDsdService	448
10.2.3.3	DcmDsdServiceRequestManufacturerNotification	452

10.2.3.4	DcmDsdServiceRequestSupplierNotification	453
10.2.3.5	DcmDsdServiceTable	453
10.2.3.6	DcmDsdSubService	454
10.2.4	DcmDsl	458
10.2.4.1	DcmDsl	458
10.2.4.2	DcmDslBuffer	459
10.2.4.3	DcmDslCallbackDCMRequestService	460
10.2.4.4	DcmDslDiagResp	461
10.2.4.5	DcmDslProtocol	462
10.2.4.6	DcmDslProtocolRow	463
10.2.4.7	DcmDslConnection	471
10.2.4.8	DcmDslMainConnection	473
10.2.4.9	DcmDslProtocolRx	476
10.2.4.10	DcmDslProtocolTx	478
10.2.4.11	DcmDslPeriodicTransmission	479
10.2.4.12	DcmDslPeriodicConnection	480
10.2.4.13	DcmDslResponseOnEvent	481
10.2.5	DcmDsp	483
10.2.5.1	DcmDspReadDTCInformation	489
10.2.5.2	DcmDspReadDTCInformationUserDefinedFault- Memory	490
10.2.5.3	DcmDspAuthentication	491
10.2.5.4	DcmDspAuthenticationConnection	495
10.2.5.5	Communication Control	502
10.2.5.5.1	DcmDspComControl	502
10.2.5.5.2	DcmDspComControlAllChannel	503
10.2.5.5.3	DcmDspComControlSetting	504
10.2.5.5.4	DcmDspComControlSpecificChannel	505
10.2.5.5.5	DcmDspComControlSubNode	506
10.2.5.6	DcmDspCommonAuthorization	508
10.2.5.7	DIDs	510
10.2.5.7.1	DcmDspDid	510
10.2.5.7.2	DcmDspDidInfo	514
10.2.5.7.3	DcmDspDidRead	516
10.2.5.7.4	DcmDspDidSignal	519
10.2.5.7.5	DcmDspDidSupportInfo	520
10.2.5.7.6	DcmDspDidRange	521
10.2.5.7.7	DcmDspDidWrite	526
10.2.5.8	DcmDspControlDTCSetting	529
10.2.5.9	Data elements	531
10.2.5.9.1	DcmDspData	531
10.2.5.9.2	DcmDspDiagnosisScaling	546
10.2.5.9.3	DcmDspArgumentScaling	547
10.2.5.9.4	DcmDspAlternativeArgumentData	548
10.2.5.9.5	DcmDspTextTableMapping	549
10.2.5.9.6	DcmDspAlternativeDataInterface	550

10.2.5.9.7	DcmDspAlternativeDataType	552
10.2.5.9.8	DcmDspAlternativeDiagnosticDataElement	553
10.2.5.9.9	DcmDataElementInstance	553
10.2.5.9.10	DcmSubElementInDataElementInstance	554
10.2.5.9.11	DcmSubElementInImplDataElementInstance	555
10.2.5.9.12	DcmDspDidDataSupportInfo	556
10.2.5.9.13	DcmDspDataInfo	557
10.2.5.10	DcmDspDidControl	558
10.2.5.11	DcmDspDidControlEnableMask	564
10.2.5.12	Ecu Reset	564
10.2.5.12.1	DcmDspEcuReset	564
10.2.5.12.2	DcmDspEcuResetRow	565
10.2.5.13	Memory	566
10.2.5.13.1	DcmDspMemory	566
10.2.5.13.2	DcmDspMemoryTransfer	567
10.2.5.13.3	DcmDspAddressAndLengthFormatIdentifier	568
10.2.5.13.4	DcmDspMemoryIdInfo	569
10.2.5.13.5	DcmDspMemoryTransferIdInfo	571
10.2.5.13.6	DcmDspReadMemoryRangeByLabelInfo	572
10.2.5.13.7	DcmDspReadMemoryRangeInfo	574
10.2.5.13.8	DcmDspWriteMemoryRangeByLabelInfo	577
10.2.5.13.9	DcmDspWriteMemoryRangeInfo	580
10.2.5.14	PIDs	585
10.2.5.14.1	DcmDspPid	585
10.2.5.14.2	DcmDspPidSupportInfo	589
10.2.5.14.3	DcmDspPidData	590
10.2.5.14.4	DcmDspPidService01	592
10.2.5.14.5	DcmDspPidService02	594
10.2.5.14.6	DcmDspPidDataSupportInfo	595
10.2.5.15	DcmDspRequestControl	596
10.2.5.16	DcmDspRequestFileTransfer	598
10.2.5.17	Response on Event	600
10.2.5.17.1	DcmDspRoe	600
10.2.5.17.2	DcmDspRoeEvent	602
10.2.5.17.3	DcmDspRoeEventProperties	604
10.2.5.17.4	DcmDspRoeOnChangeOfDataIdentifier	604
10.2.5.17.5	DcmDspRoeOnDTCStatusChange	605
10.2.5.17.6	DcmDspRoeEventWindowTime	605
10.2.5.18	Routines	607
10.2.5.18.1	DcmDspRoutine	607
10.2.5.18.2	DcmDspRequestRoutineResults	609
10.2.5.18.3	DcmDspRequestRoutineResultsIn	613
10.2.5.18.4	DcmDspRequestRoutineResultsInSignal	614
10.2.5.18.5	DcmDspRequestRoutineResultsOut	616
10.2.5.18.6	DcmDspRequestRoutineResultsOutSignal	617
10.2.5.18.7	DcmDspStartRoutine	619

10.2.5.18.8	DcmDspStartRoutineIn	623
10.2.5.18.9	DcmDspStartRoutineInSignal	625
10.2.5.18.10	DcmDspStartRoutineOut	628
10.2.5.18.11	DcmDspStartRoutineOutSignal	629
10.2.5.18.12	DcmDspStopRoutine	632
10.2.5.18.13	DcmDspStopRoutineIn	636
10.2.5.18.14	DcmDspStopRoutineInSignal	638
10.2.5.18.15	DcmDspStopRoutineOut	641
10.2.5.18.16	DcmDspStopRoutineOutSignal	642
10.2.5.19	Session Security and Modes	645
10.2.5.19.1	DcmDspSecurity	645
10.2.5.19.2	DcmDspSecurityRow	647
10.2.5.19.3	DcmDspSession	653
10.2.5.19.4	DcmDspSessionRow	655
10.2.5.19.5	DcmModeCondition	657
10.2.5.19.6	DcmSwcDataElementValue	661
10.2.5.19.7	DcmSwcDataElementPrimitive	661
10.2.5.19.8	DcmSwcDataElementArray	661
10.2.5.19.9	DcmSwcDataElementArrayElement	662
10.2.5.19.10	DcmModeRule	663
10.2.5.20	DcmDspVehInfo	665
10.2.5.21	DcmDspVehInfoData	667
10.2.5.22	DcmDspPeriodicTransmission	669
10.2.5.23	DcmDspClearDTC	674
10.2.6	DcmGeneral	676
10.3	Protocol Configuration Example	681
10.4	Published Information	682
A	Not applicable requirements	682

1 Introduction and functional overview

The *Dcm* SWS describes the functionality, the API, and the configuration of the AUTOSAR Basic Software module *Dcm* (Diagnostic Communication Manager). The *Dcm* module provides a common API for diagnostic services. The functionality of the *Dcm* module is used by external diagnostic tools during the development, manufacturing or service.

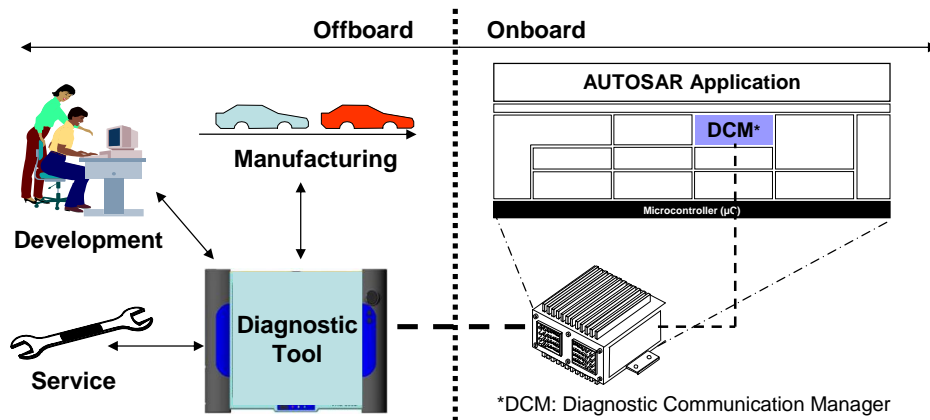


Figure 1.1: Overview of the communication between the external diagnostic tools and the onboard AUTOSAR Application

The *Dcm* module ensures diagnostic data flow and manages the diagnostic states, especially diagnostic sessions and security states. Furthermore, the *Dcm* module checks if the diagnostic service request is supported and if the service may be executed in the current session according to the diagnostic states. The *Dcm* module provides the OSI-Layers 5 to 7 of Table 1: Diagnostic protocols and OSI-Layer.

OSI-Layer	Protocols				
7	UDS-Protocol - ISO14229-1 [1]				Legislated OBD - ISO15031-5 [2]
6	-	-	-	-	-
5	ISO15765-3	-	-	-	ISO 15765-4
4	ISO15765-2	-	-	-	-
3	ISO15765-2	-	-	-	ISO 15765-4
2	CAN-Protocol	LIN-Protocol	FlexRay	MOST	ISO 15765-4
1	CAN-Protocol	LIN-Protocol	FlexRay	MOST	ISO 15765-4

Table 1.1: Diagnostic protocols and OSI-Layers

At OSI-level 7, the *Dcm* module provides an extensive set of ISO14229-1 [1] services. In addition, the *Dcm* module provides mechanisms to support the OBD services \$01 -

\$0A defined in documents [3, SAE J1979] and [2, ISO 15031-5]. With these services, Autosar OBD functionality is capable of meeting all light duty OBD regulations worldwide (California OBDII, EOBD, Japan OBD, and all others). At OSI-level 5, the Dcm module handles the network-independent sections of the following specifications:

- ISO15765-3 [4]: Implementation of unified diagnostic services (UDS on CAN)
- ISO15765-4 [5]: Requirements for emission-related systems, Chapter 5 "Session Layer"

In the AUTOSAR Architecture the Diagnostic Communication Manager is located in the Communication Services (Service Layer).

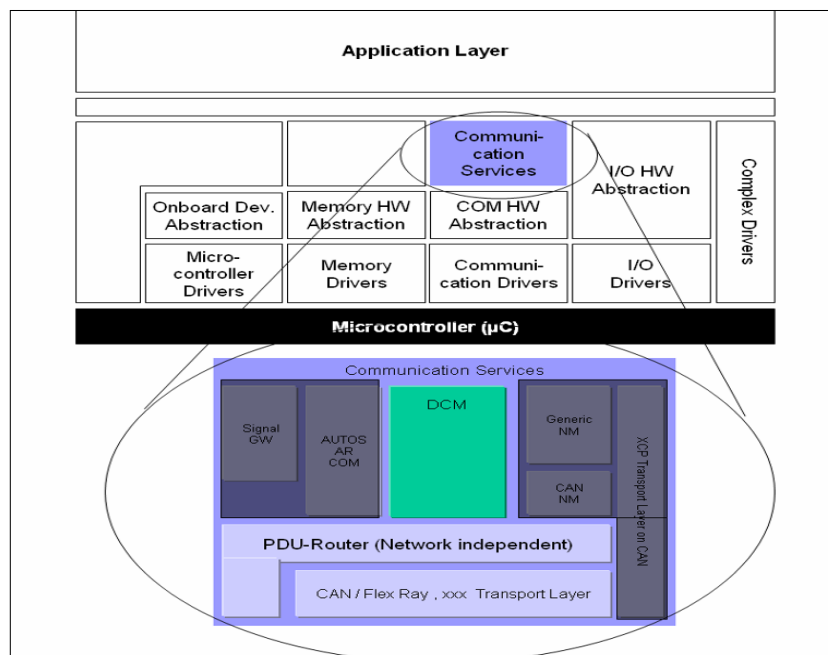


Figure 1.2: Position of the Dcm module in AUTOSAR Architecture

The Dcm module is network-independent. All network-specific functionality (the specifics of networks like CAN, LIN, FlexRay or MOST) is handled outside of the Dcm module. The PDU Router (PduR) module provides a network-independent interface to the Dcm module. The Dcm module receives a diagnostic message from the PduR module. The Dcm module processes and checks internally the diagnostic message. As part of processing the requested diagnostic service, the Dcm will interact with other BSW modules or with SW-Components (through the RTE) to obtain requested data or to execute requested commands. This processing is very service-specific. Typically, the Dcm will assemble the gathered information and send a message back through the PduR module.

2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the <MODULE_NAME> module that are not included in the [6, AUTOSAR glossary].

Abbreviation / Acronym:	Description:
Application Layer	The Application Layer is placed above the RTE. Within the Application Layer the AUTOSAR Software-Components are placed.
Atomic Sender/Receiver interface	An atomic sender-receiver interface can be used to group DID data elements into one record data element prototype. All data elements can be read or write having a single read or write operation.
Channel	A link at which a data transfer can take place. If there is more than one Channel, there is normally some kind of ID assigned to the Channel.
Diagnostic Channel	A link at which a data transfer between a diagnostic tool and an ECU can take place. Example: An ECU is connected via CAN and the diagnostic channel has an assigned CAN-ID. Diagnostic channels connected to other bus-systems such as MOST, FlexRay, LIN, etc. are also possible.
External Diagnostic Tool	<p>A device which is NOT permanently connected to the vehicle communication network. This External Diagnostic Tool can be connected to the vehicle for various purposes, as e.g. for:</p> <ul style="list-style-type: none"> • development • manufacturing • service (in a garage) <p>Example External Diagnostic Tools are:</p> <ul style="list-style-type: none"> • a diagnostic tester • an OBD scan tool <p>The External Diagnostic Tool is to be connected by a mechanic to gather information from "inside" the car.</p>
Freeze Frame	A set of the vehicle/system operation conditions at a specific time.
Functional Addressing	The diagnostic communication model where a group or all nodes of a specific communication network receive a message from one sending node (1-n communication). This model is also referred to as 'broadcast' or 'multicast'. OBD communication will always be done in the Functional Addressing mode.
Internal Diagnostic Tool	<p>A device/ECU which is connected to the vehicle communication network. The Internal Diagnostic Tool can be used for:</p> <ul style="list-style-type: none"> • advanced event tracking • advanced analysis • for service <p>The behavior of the Internal Diagnostic Tool can be the same as of an External Diagnostic Tool. The notion of "Internal Diagnostic Tool" does not imply that it is included in each ECU as an AUTOSAR Software-Component.</p>

Abbreviation / Acronym:	Description:
Physical Addressing	The diagnostic communication model where a node of a specific communication network receives a message from one sending node (1-1 communication). This model is also referred to as 'unicast'.
UDS Service	this refers to a UDS Service as defined in ISO14229-1 [1].
OBD Service	This refers to an OBD Service as defined in ISO15031-5 [2].
AddressAndLengthFormat Identifier	Defines the number of bytes used for the memoryAddress and memorySize parameter in the request messages.
OBD Scan tool	See definition External Diagnostic Tool.
periodic transmission rate	Time interval value that defines the time between two calls of a periodic data identifier transmission. The value is configuration specific and there are separate values for fast, medium and slow periodic data transmission. The configured value is always an integer multiple of the Dcm main task time.

Terms:	Description:
API	Application Programming Interface
CAN	Controller Area Network
CEMR	ControlEnableMaskRecord
Dcm	Diagnostic Communication Manager
Dem	Diagnostic Event Manager
Det	Default Error Tracer
DID	Data Identifier
DSD	Diagnostic Service Dispatcher (submodule of the Dcm module)
DSL	Diagnostic Session Layer (submodule of the Dcm module)
DSP	Diagnostic Service Processing (submodule of the Dcm module)
DTC	Diagnostic Trouble Codes
ID	Identifier
LIN	Local Interconnect Network
MCU	Micro-Controller Unit
MOST	Media Orientated System Transport
NRC	Negative Response Code
OBD	On-Board Diagnosis
OSI	Open Systems Interconnection
PDID	Periodic Data Identifier, periodically send by the Dcm after a request of ReadDataByPeriodicIdentifier
PDU	Protocol Data Unit
PID	Parameter Identifier
RCRRP	Response correctly received - response pending
RID	Routine Identifier
ROE	ResponseOnEvent
RTE	Runtime Environment
SAP	Service Access Point
SDU	Service Data Unit
SID	Service Identifier
SW-C	Software-Component
TP	Transport Protocol
UDS	Unified Diagnostic Services
Xxx_	Placeholder for an API provider
SPRMIB	suppressPosRspMsgIndicationBit

2.1 Typographical Conventions

This document uses the following typographical conventions:

- see configuration parameter `myConfigurationParameter`: this is a reference to a configuration parameter which can be found in Chapter 10.
- `myFunction()`: this is a function provided or required by the module as defined in Chapter 8

3 Related documentation

3.1 Input documents & related standards and norms

- [1] Unified diagnostic services (UDS) – Part 1: Specification and requirements (Release 2013-03)
<http://www.iso.org>
- [2] Road vehicles – Communication between vehicle and external equipment for emission-related diagnostic – Part 5: Emission-related diagnostic services.
<http://www.iso.org>
- [3] SAE J1979
- [4] Diagnostics on controller area network (CAN) – Part 3: Implementation of unified diagnostic services (UDS on CAN) (Release 2004 10-06)
- [5] Diagnostics on controller area network (CAN) – Part 4: Requirements for emission-related systems (Release 2005 01-04)
- [6] Glossary
AUTOSAR_TR_Glossary
- [7] General Specification of Basic Software Modules
AUTOSAR_SWS_BSWGeneral
- [8] General Requirements on Basic Software Modules
AUTOSAR_SRS_BSWGeneral
- [9] ISO 17356-3: Road vehicles – Open interface for embedded automotive applications – Part 3: OSEK/VDX Operating System (OS)
- [10] Specification of PDU Router
AUTOSAR_SWS_PDURouter
- [11] Road vehicles – Diagnostics on Controller Area Networks (CAN) – Part2: Network layer services
- [12] Specification of Diagnostic Event Manager
AUTOSAR_SWS_DiagnosticEventManager

- [13] Road vehicles – Communication between vehicle and external equipment for emission-related diagnostic – Part 6: Diagnostic trouble code definitions
<http://www.iso.org>
- [14] Specification of NVRAM Manager
AUTOSAR_SWS_NVRAMManager
- [15] Specification of Crypto Service Manager
AUTOSAR_SWS_CryptoServiceManager
- [16] Specification of Key Manager
AUTOSAR_SWS_KeyManager
- [17] Specification of I/O Hardware Abstraction
AUTOSAR_SWS_IOHardwareAbstraction

3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules [7, SWS BSW General] , which is also valid for Diagnostic Communication Manager.

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

4 Constraints and assumptions

4.1 Limitations

The following limitations apply when using the `Dcm` module:

- The `Dcm` module does not provide any diagnostic multi-channel capabilities. This means that parallel requests of a tester addressed to different independent functionalities cannot be processed by a single `Dcm` module. Furthermore, the concept currently implemented does not take more than one instance of a `Dcm` module residing in one ECU into account. As the legislator requires that emission-related service requests according to ISO 15031-5 [2] shall be processed prior to any enhanced diagnostic requests, the `Dcm` module provides a protocol switching mechanism based on protocol prioritization.
- `UDS` Service AccessTimingParameter (0x83) is not supported by the ISO standards in `CAN` and `LIN`. Also it is not planned to support this service with `FlexRay`. Therefore no support for this service is planned.
- Subfunction `onComparisonOfValues` of Service `ResponseOnEvent` is not supported in the current release.

- Subfunction onTimerInterrupt of Service ResponseOnEvent is not supported in the current release.
- UDS Service SecuredDataTransmission (0x84) is not supported in the current release.
- The Dcm SWS does not cover any SAE J1939 related diagnostic requirements.
- Due to DEM limitation, the diagnostic service \$19 05 is limited to the OBD legislative freeze frame.
- Management of IOControl service without InputOutputControlParameter in request and response is not supported
- The length of controlState parameter in IOControl request and response has to be of same size (due to the one configuration parameter DcmDspDataByteSize)
- Same layout of a DID which is used in RDBI, WDBI or IOCBI services
- The user optional parameter DTCSettingControlOptionRecord in the ControlDTCSetting request is only supported if it corresponds to a groupOfAllDTCs (0xFFFFFFFF) value. In other cases it has to be managed in a vendor specific implementation.
- Only the ControlDTCSetting sub-functions 0x01 and 0x02 are supported.
- The handling of infrastructure errors reported by the RTE during DCM/DEM <-> SW-C interactions is missing from the SWS and might have to be taken into account by implementers if they need it.
- The Dcm does not support DLT for ROE
- The ROE ServiceToRespondTo does not support PageBuffering
- ROE only supports sub-function listed in Table 2
- DID range feature cannot be applied for services DynamicallyDefineDataIdentifier, ReadDataByPeriodicIdentifier and InputOutputControlById
- AUTOSAR Dcm is not intended to be used in the bootloader
- PeriodicTransmission is not possible on FlexRay, as ISO 14229-4 demands header information (address information (source and target address) and FPL (Frame Payload length)). This information can't be filled with the specified concept of IF interface.
- The specification of the transformer for intra ecu communication between the Dcm module and the NvBlockSoftwareComponentType is not standardized in the current AUTOSAR release. For this scenario custom transformers implemented by a complex driver can be used. To elaborate on this the responsible stakeholder (usually the OEM) needs to specify the custom transformer from a behavioral point of view in a separate document (this might include definition of byte-ordering or alignment). If there is the necessity to define transformer specific attributes in

the model this can be done using special data groups in `UserDefinedTransformationDescription` and `UserDefinedTransformationSignalProps`. For the configuration of this scenario, a `DataPrototypeMapping` shall exist for the affected `SenderReceiverInterfaces` of the `Dcm` module and the `NvBlockSoftwareComponentType` which refers to a `DataTransformation` in the role `firstToSecondDataTransformation`. This `DataTransformation` shall reference exactly one `TransformationTechnology` in the role `transformerChain` with the `transformerClass` attribute set to "serializer" and may compose a `UserDefinedTransformationDescription` in the role `transformationDescription`.

- In certain situations the `Dcm` module is capable to process diagnostic requests in parallel. This possibility is explicitly limited of OBD in parallel to UDS protocol processing. No other protocol combination can be processed in parallel. Particularly the use case of parallel processing of two or more UDS protocol requests or WWH-OBD and UDS protocols is not supported.
- For UDS service 0x29, the `Dcm` supports only the sub-functions for PKI. Authentication via challenge response is not supported.
- For UDS service 0x29, secure diagnostic communication with Diffie-Hellmann key exchange is not supported.

4.2 Applicability to car domains

The `Dcm` module can be used for all car domains.

4.3 Applicability to emission-related environments (OBD)

This `Dcm` SWS is intended to fulfill the emission related requirements given by legislator. However, the supplier of the emission related system is responsible to fulfill the OBD requirements. Certain requirements cannot be fulfilled by the `Dcm` module by itself, but need to be considered at the level of the entire ECU or system. Example: During the integration of the `Dcm` module within the system, the timing requirements (50ms response time) must be fulfilled.

For WWH-OBD only the `FunctionalGroupIdentifier` 0x33 is currently supported.

5 Dependencies to other modules

The AUTOSAR Diagnostic Communication Manager (DCM) has interfaces and dependencies to the following Basic Software modules and SW-Cs:

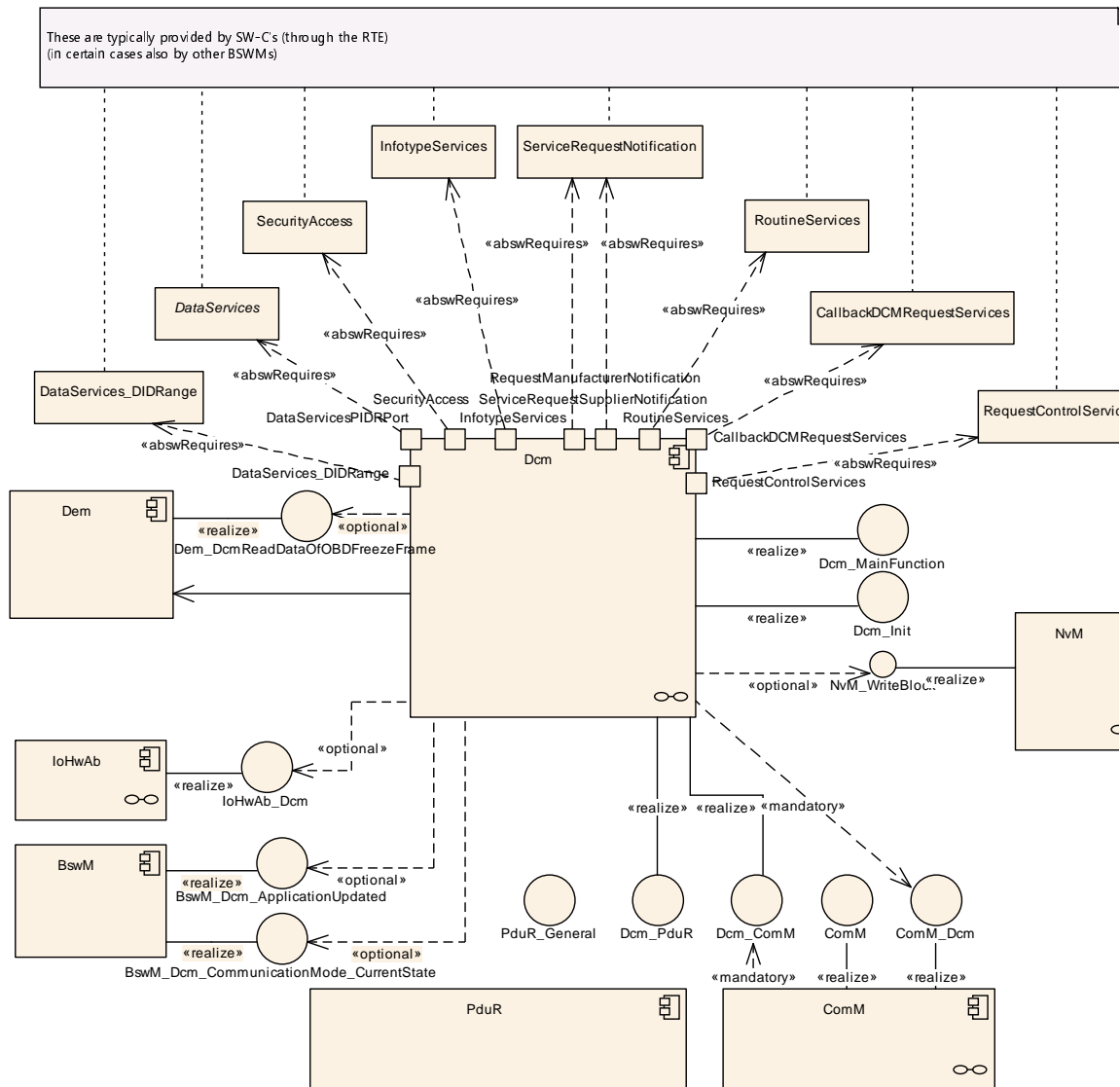


Figure 5.1: Interaction of the Dcm with other modules

- Diagnostic Event Manager (DEM): The DEM module provides function to retrieve all information related to fault memory such that the Dcm module is able to respond to tester requests by reading data from the fault memory.
- Protocol Data Unit Router (PduR module): The PduR module provides functions to transmit and receive diagnostic data. Proper operation of the Dcm module presumes that the PduR interface supports all service primitives defined for the Service Access Point (SAP) between diagnostic application layer and underlying transport layer (see ISO14229-1 [1], chapter 5 Application layer services).
- Communication Manager (ComM): The ComM module provides functions such that the Dcm module can indicate the states "active" and "inactive" for diagnostic communication. The Dcm module provides functionality to handle the communication requirements "Full-/ Silent-/ No-Communication". Additionally, the Dcm module provides the functionality to enable and disable Diagnostic Communication if requested by the ComM module.

- **SW-C** and **RTE**: The **Dcm** module has the capability to analyze the received diagnostic request data stream and handles all functionalities related to diagnostic communication such as protocol handling and timing. Based on the analysis of the request data stream the **Dcm** module assembles the response data stream and delegates routines or IO-Control executions to SW-Cs .If any of the data elements or functional states cannot be provided by the **Dcm** module itself the **Dcm** requests data or functional states from SW-Cs via port-interfaces or from other BSW modules through direct function-calls.
- **BswM**: The **Dcm** notifies the **BswM** that the application was updated if the initialization of the **Dcm** is the consequence of a jump from the bootloader . The **Dcm** also indicates to the **BswM** a communication mode change.
- **Crypto Service Manager (Csm)**: The crypto service module provides a wide range of cryptographic algorithms. The **Csm** is used for authentication calculation.
- **Key Manager (KeyM)**: The key manager module provides support for certificate handling and APIs to realize authenticated diagnostics via certificates.

6 Requirements Tracing

The following tables reference the requirements specified in AUTOSAR_RS_RS_Diagnostics as well as [8] and links to the fulfillment of these. Please note that if column “Satisfied by” is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[RS_Diag_04002]	The Diagnostic event (fault) management shall be established as Basic SW Module	[SWS_Dcm_NA_00999]
[RS_Diag_04003]	Network independent design	[SWS_Dcm_00030]
[RS_Diag_04005]	Manage Security Access level handling	[SWS_Dcm_00020] [SWS_Dcm_00033] [SWS_Dcm_00252] [SWS_Dcm_00338] [SWS_Dcm_01397] [SWS_Dcm_01535] [SWS_Dcm_CONSTR_6083]
[RS_Diag_04006]	Manage session handling	[SWS_Dcm_00022] [SWS_Dcm_00250] [SWS_Dcm_00339] [SWS_Dcm_01373] [SWS_Dcm_01374] [SWS_Dcm_01375] [SWS_Dcm_01376] [SWS_Dcm_01377] [SWS_Dcm_01378] [SWS_Dcm_01535]
[RS_Diag_04007]	Provide a diagnostic service handling for the applications involved in diagnostic functionality	[SWS_Dcm_NA_00999]
[RS_Diag_04011]	Provide diagnostic state information to applications	[SWS_Dcm_00338] [SWS_Dcm_00339] [SWS_Dcm_00340] [SWS_Dcm_01321] [SWS_Dcm_01322]

Requirement	Description	Satisfied by
[RS_Diag_04015]	Timing handling according to ISO15765-3	[SWS_Dcm_00027] [SWS_Dcm_00030] [SWS_Dcm_00143] [SWS_Dcm_00144] [SWS_Dcm_00311]
[RS_Diag_04016]	Support "Busy handling" by sending a negative response 0x78	[SWS_Dcm_00024]
[RS_Diag_04019]	Provide confirmation after transmit diagnostic responses to the application	[SWS_Dcm_NA_00999]
[RS_Diag_04020]	Suppress responses to diagnostic tool requests	[SWS_Dcm_00001] [SWS_Dcm_00200]
[RS_Diag_04021]	Handling of different diagnostic sessions in parallel	[SWS_Dcm_00015]
[RS_Diag_04024]	Access and handle specific data elements and data element groups if requested by an external scan tool	[SWS_Dcm_NA_00999]
[RS_Diag_04031]	Notify the Function Inhibition Manager (FIM) upon changes of the event status in order to process them according to the SW components dependencies	[SWS_Dcm_NA_00999]
[RS_Diag_04032]	Different diagnostic addresses shall be supported by multiple (physical) channels	[SWS_Dcm_NA_00999]
[RS_Diag_04033]	Support the upload/download services for reading/writing data in an ECU in an extended and manufacturer specific diagnostic session	[SWS_Dcm_00496] [SWS_Dcm_00499] [SWS_Dcm_00502] [SWS_Dcm_00503] [SWS_Dcm_00504] [SWS_Dcm_00505] [SWS_Dcm_01417] [SWS_Dcm_01418] [SWS_Dcm_01419] [SWS_Dcm_01420] [SWS_Dcm_01421] [SWS_Dcm_01422]
[RS_Diag_04057]	Classification of events for series production, OBD and expert usage	[SWS_Dcm_NA_00999]
[RS_Diag_04058]	Ability to access different event memories	[SWS_Dcm_00004] [SWS_Dcm_00005] [SWS_Dcm_00077] [SWS_Dcm_00279] [SWS_Dcm_00293] [SWS_Dcm_00295] [SWS_Dcm_00378] [SWS_Dcm_00383] [SWS_Dcm_00384] [SWS_Dcm_00388] [SWS_Dcm_00389] [SWS_Dcm_00393] [SWS_Dcm_00465] [SWS_Dcm_01147] [SWS_Dcm_01263]
[RS_Diag_04059]	Configuration of timing parameters	[SWS_Dcm_NA_00999]
[RS_Diag_04063]	Process a dedicated event identifier for each monitoring path to support an autonomous handling of different events/ faults	[SWS_Dcm_NA_00999]
[RS_Diag_04064]	Provide configurable buffer sizes for storage of the events, status information and environmental data	[SWS_Dcm_NA_00999]

Requirement	Description	Satisfied by
[RS_Diag_04067]	Provide the diagnostic status information according to ISO 14229-1	[SWS_Dcm_00293] [SWS_Dcm_00378]
[RS_Diag_04068]	The diagnostic in AUTOSAR shall support event specific debounce counters to improve signal quality internally (According to ISO 14229-1 Appendix D)	[SWS_Dcm_NA_00999]
[RS_Diag_04071]	Process events according to their defined importance like priority and/or severity	[SWS_Dcm_NA_00999]
[RS_Diag_04077]	Uses standard mechanisms provided by persistency modules	[SWS_Dcm_NA_00999]
[RS_Diag_04085]	No description	[SWS_Dcm_NA_00999]
[RS_Diag_04086]	No description	[SWS_Dcm_NA_00999]
[RS_Diag_04087]	No description	[SWS_Dcm_NA_00999]
[RS_Diag_04089]	No description	[SWS_Dcm_NA_00999]
[RS_Diag_04090]	No description	[SWS_Dcm_NA_00999]
[RS_Diag_04091]	Notification about valid freeze frame data to applications	[SWS_Dcm_NA_00999]
[RS_Diag_04093]	Memory overflow indication	[SWS_Dcm_NA_00999]
[RS_Diag_04097]	Decentralized and modular diagnostic configuration in applications	[SWS_Dcm_NA_00999]
[RS_Diag_04098]	Interact with standard bootloader	[SWS_Dcm_00532] [SWS_Dcm_00535] [SWS_Dcm_00536] [SWS_Dcm_00592] [SWS_Dcm_00654] [SWS_Dcm_00767] [SWS_Dcm_01163] [SWS_Dcm_01177] [SWS_Dcm_01423] [SWS_Dcm_01424] [SWS_Dcm_01425] [SWS_Dcm_CONSTR_6080]
[RS_Diag_04100]	Interface for logging and tracing	[SWS_Dcm_NA_00999]
[RS_Diag_04101]	No description	[SWS_Dcm_NA_00999]
[RS_Diag_04105]	Event memory management	[SWS_Dcm_NA_00999]
[RS_Diag_04107]	Provide defensive behavior	[SWS_Dcm_NA_00999]
[RS_Diag_04109]	Provide an interface to retrieve the number of event memory entries	[SWS_Dcm_NA_00999]
[RS_Diag_04110]	SAE J1939 lamp status	[SWS_Dcm_NA_00999]
[RS_Diag_04111]	SAE J1939 Expanded-Freeze Frame	[SWS_Dcm_NA_00999]
[RS_Diag_04112]	The DEM module shall support DTCs according to SAE J1939	[SWS_Dcm_NA_00999]
[RS_Diag_04113]	Support a set of SAE J1939 DM-messages	[SWS_Dcm_NA_00999]
[RS_Diag_04115]	The optional parameter DTCSettingControlOption Record as part of UDS service ControlDTCSetting shall be limited to GroupOfDTC	[SWS_Dcm_00406] [SWS_Dcm_01063]
[RS_Diag_04117]	Configurable behavior for DTC deletion	[SWS_Dcm_NA_00999]

Requirement	Description	Satisfied by
[RS_Diag_04118]	Optionally support event displacement	[SWS_Dcm_NA_00999]
[RS_Diag_04119]	Handle the execution of diagnostic services according to the assigned diagnostic session	[SWS_Dcm_00628] [SWS_Dcm_00858] [SWS_Dcm_00859] [SWS_Dcm_01435] [SWS_Dcm_NA_00999]
[RS_Diag_04120]	Support a predefined Address AndLengthFormatIdentifier	[SWS_Dcm_NA_00999]
[RS_Diag_04121]	Provide the handling of service DynamicallyDefineDataIdentifier according to ISO 14229-1	[SWS_Dcm_NA_00999]
[RS_Diag_04123]	Harmonized Driving//WarmUp cycles	[SWS_Dcm_NA_00999]
[RS_Diag_04124]	Store the current debounce counter value non-volatile to over a power-down cycle	[SWS_Dcm_NA_00999]
[RS_Diag_04125]	Event debounce counter shall be configurable	[SWS_Dcm_NA_00999]
[RS_Diag_04126]	Configurable suppression of events	[SWS_Dcm_NA_00999]
[RS_Diag_04127]	Configurable record numbers and trigger options for DTCSnapshotRecords and DTCExtendedDataRecords	[SWS_Dcm_NA_00999]
[RS_Diag_04129]	Provide OBD-specific configuration capabilities	[SWS_Dcm_NA_00999]
[RS_Diag_04131]	Consistent event management mechanisms	[SWS_Dcm_NA_00999]
[RS_Diag_04133]	Aging for event memory entries	[SWS_Dcm_NA_00999]
[RS_Diag_04135]	Support UDS service \$38 (RequestFileTransfer)	[SWS_Dcm_NA_00999]
[RS_Diag_04136]	Configurable "confirmed" threshold	[SWS_Dcm_NA_00999]
[RS_Diag_04137]	Definition of replacement failure	[SWS_Dcm_NA_00999]
[RS_Diag_04139]	Support subfunction 0x42 of UDS service 0x19	[SWS_Dcm_NA_00999]
[RS_Diag_04140]	Aging for UDS status bits "confirmedDTC" and "testFailed SinceLastClear"	[SWS_Dcm_NA_00999]
[RS_Diag_04143]	No description	[SWS_Dcm_NA_00999]
[RS_Diag_04144]	No description	[SWS_Dcm_NA_00999]
[RS_Diag_04145]	No description	[SWS_Dcm_NA_00999]
[RS_Diag_04146]	No description	[SWS_Dcm_NA_00999]
[RS_Diag_04147]	Communication with the transport layers to receive and send diagnostic data	[SWS_Dcm_00642] [SWS_Dcm_01186]
[RS_Diag_04148]	Provide capabilities to inform applications about diagnostic data changes	[SWS_Dcm_NA_00999]
[RS_Diag_04150]	Support the primary fault memory defined by ISO 14229-1	[SWS_Dcm_NA_00999]
[RS_Diag_04151]	Event status handling	[SWS_Dcm_NA_00999]
[RS_Diag_04153]	Support generic connections	[SWS_Dcm_00849] [SWS_Dcm_01347] [SWS_Dcm_01348]
[RS_Diag_04159]	Control of DTC storage	[SWS_Dcm_00249] [SWS_Dcm_01399]

Requirement	Description	Satisfied by
[RS_Diag_04162]	Parallel fault memory access	[SWS_Dcm_01369] [SWS_Dcm_01370] [SWS_Dcm_01371] [SWS_Dcm_01372]
[RS_Diag_04163]	Parallel OBD and UDS processing	[SWS_Dcm_01365] [SWS_Dcm_01366] [SWS_Dcm_01367]
[RS_Diag_04215]	Support of UDS service Read DataByPeriodicIdentifier (0x2A)	[SWS_Dcm_00716] [SWS_Dcm_00721] [SWS_Dcm_00722] [SWS_Dcm_00820] [SWS_Dcm_00843] [SWS_Dcm_00851] [SWS_Dcm_01093] [SWS_Dcm_01094] [SWS_Dcm_01095] [SWS_Dcm_01096] [SWS_Dcm_01097] [SWS_Dcm_01098] [SWS_Dcm_01099] [SWS_Dcm_01100] [SWS_Dcm_01101] [SWS_Dcm_01103] [SWS_Dcm_01104] [SWS_Dcm_01105] [SWS_Dcm_01106] [SWS_Dcm_01107] [SWS_Dcm_01108] [SWS_Dcm_01109] [SWS_Dcm_01110] [SWS_Dcm_01111] [SWS_Dcm_01112] [SWS_Dcm_01115] [SWS_Dcm_01117] [SWS_Dcm_01118] [SWS_Dcm_01426] [SWS_Dcm_01427] [SWS_Dcm_01428] [SWS_Dcm_01568] [SWS_Dcm_01569] [SWS_Dcm_01570] [SWS_Dcm_01571] [SWS_Dcm_01572] [SWS_Dcm_01573] [SWS_Dcm_01574] [SWS_Dcm_01575] [SWS_Dcm_01576] [SWS_Dcm_01577] [SWS_Dcm_01578]
[RS_Diag_04218]	Support of UDS service 0x2F InputOutputControlByIdentifier	[SWS_Dcm_00256] [SWS_Dcm_00396] [SWS_Dcm_00397] [SWS_Dcm_00398] [SWS_Dcm_00399] [SWS_Dcm_00563] [SWS_Dcm_00564] [SWS_Dcm_00565] [SWS_Dcm_00579] [SWS_Dcm_00580] [SWS_Dcm_00581] [SWS_Dcm_00640] [SWS_Dcm_00680] [SWS_Dcm_00682] [SWS_Dcm_01273] [SWS_Dcm_01274] [SWS_Dcm_01275] [SWS_Dcm_01277] [SWS_Dcm_01313] [SWS_Dcm_01434] [SWS_Dcm_01436] [SWS_Dcm_01437] [SWS_Dcm_01438] [SWS_Dcm_01439] [SWS_Dcm_01440] [SWS_Dcm_01441] [SWS_Dcm_01554] [SWS_Dcm_91057] [SWS_Dcm_91058] [SWS_Dcm_91059] [SWS_Dcm_91060] [SWS_Dcm_91061] [SWS_Dcm_CONSTR_6084] [SWS_Dcm_CONSTR_6085] [SWS_Dcm_CONSTR_6086]
[RS_Diag_04224]	Support the UDS service 0x31 (RoutineControl) according to ISO 14229-1	[SWS_Dcm_01442] [SWS_Dcm_01443] [SWS_Dcm_01580] [SWS_Dcm_01581]

Requirement	Description	Satisfied by
[RS_Diag_04230]	Support of UDS service 0x29 (Authentication)	[SWS_Dcm_01459] [SWS_Dcm_01460] [SWS_Dcm_01461] [SWS_Dcm_01463] [SWS_Dcm_01464] [SWS_Dcm_01465] [SWS_Dcm_01466] [SWS_Dcm_01467] [SWS_Dcm_01468] [SWS_Dcm_01469] [SWS_Dcm_01477] [SWS_Dcm_01478] [SWS_Dcm_01479] [SWS_Dcm_01480] [SWS_Dcm_01481] [SWS_Dcm_01482] [SWS_Dcm_01483] [SWS_Dcm_01484] [SWS_Dcm_01485] [SWS_Dcm_01486] [SWS_Dcm_01487] [SWS_Dcm_01488] [SWS_Dcm_01489] [SWS_Dcm_01493] [SWS_Dcm_01503] [SWS_Dcm_01504] [SWS_Dcm_01506] [SWS_Dcm_01507] [SWS_Dcm_01508] [SWS_Dcm_01509] [SWS_Dcm_01510] [SWS_Dcm_01511] [SWS_Dcm_01512] [SWS_Dcm_01513] [SWS_Dcm_01515] [SWS_Dcm_01518] [SWS_Dcm_01528] [SWS_Dcm_01529] [SWS_Dcm_01530] [SWS_Dcm_01532] [SWS_Dcm_01533] [SWS_Dcm_01534] [SWS_Dcm_01535] [SWS_Dcm_01536] [SWS_Dcm_01537] [SWS_Dcm_01538] [SWS_Dcm_01542] [SWS_Dcm_01544] [SWS_Dcm_01551] [SWS_Dcm_01558] [SWS_Dcm_01559] [SWS_Dcm_01560] [SWS_Dcm_01561] [SWS_Dcm_01565] [SWS_Dcm_01579] [SWS_Dcm_CONSTR_6094] [SWS_Dcm_CONSTR_6096]
[RS_Diag_04232]	Access rights in client certificates	[SWS_Dcm_01516] [SWS_Dcm_CONSTR_6087] [SWS_Dcm_CONSTR_6089]
[RS_Diag_04233]	Access granularity of diagnostic services	[SWS_Dcm_01496] [SWS_Dcm_01514] [SWS_Dcm_01539] [SWS_Dcm_01540] [SWS_Dcm_01541] [SWS_Dcm_01545] [SWS_Dcm_01546] [SWS_Dcm_01547] [SWS_Dcm_01548] [SWS_Dcm_01549] [SWS_Dcm_01552] [SWS_Dcm_01553] [SWS_Dcm_01555] [SWS_Dcm_01556] [SWS_Dcm_01557] [SWS_Dcm_01562] [SWS_Dcm_CONSTR_6088]
[RS_Diag_04233; RS_Diag_04234]	No description	[SWS_Dcm_01522] [SWS_Dcm_01523] [SWS_Dcm_01524] [SWS_Dcm_01525] [SWS_Dcm_01526] [SWS_Dcm_01527]
[RS_Diag_04234]	Binary compatibility of white list for individual access	[SWS_Dcm_01521]
[RS_Diag_04235]	Client certificate validity	[SWS_Dcm_01470] [SWS_Dcm_01471] [SWS_Dcm_01472] [SWS_Dcm_01473] [SWS_Dcm_01474] [SWS_Dcm_01475] [SWS_Dcm_01476] [SWS_Dcm_01563]
[RS_Diag_04236]	Client certificate target identification	[SWS_Dcm_01490] [SWS_Dcm_01491] [SWS_Dcm_01492]

Requirement	Description	Satisfied by
[RS_Diag_04243]	Update of constant parameters through diagnostics	[SWS_Dcm_01582] [SWS_Dcm_01583] [SWS_Dcm_01584] [SWS_Dcm_01585] [SWS_Dcm_01586] [SWS_Dcm_01587] [SWS_Dcm_91088]
[SRS_BSW_00003]	All software modules shall provide version and identification information	[SWS_Dcm_00065]
[SRS_BSW_00005]	Modules of the μ C Abstraction Layer (MCAL) may not have hard coded horizontal interfaces	[SWS_Dcm_NA_00999]
[SRS_BSW_00006]	The source code of software modules above the μ C Abstraction Layer (MCAL) shall not be processor and compiler dependent.	[SWS_Dcm_NA_00999]
[SRS_BSW_00007]	All Basic SW Modules written in C language shall conform to the MISRA C 2012 Standard.	[SWS_Dcm_NA_00999]
[SRS_BSW_00009]	All Basic SW Modules shall be documented according to a common standard.	[SWS_Dcm_NA_00999]
[SRS_BSW_00010]	The memory consumption of all Basic SW Modules shall be documented for a defined configuration for all supported platforms.	[SWS_Dcm_NA_00999]
[SRS_BSW_00101]	The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function	[SWS_Dcm_00033] [SWS_Dcm_00034] [SWS_Dcm_00035] [SWS_Dcm_00036] [SWS_Dcm_00037]
[SRS_BSW_00158]	No description	[SWS_Dcm_NA_00999]
[SRS_BSW_00159]	All modules of the AUTOSAR Basic Software shall support a tool based configuration	[SWS_Dcm_NA_00999]
[SRS_BSW_00160]	Configuration files of AUTOSAR Basic SW module shall be readable for human beings	[SWS_Dcm_NA_00999]
[SRS_BSW_00161]	The AUTOSAR Basic Software shall provide a microcontroller abstraction layer which provides a standardized interface to higher software layers	[SWS_Dcm_NA_00999]
[SRS_BSW_00162]	The AUTOSAR Basic Software shall provide a hardware abstraction layer	[SWS_Dcm_NA_00999]
[SRS_BSW_00164]	The Implementation of interrupt service routines shall be done by the Operating System, complex drivers or modules	[SWS_Dcm_NA_00999]
[SRS_BSW_00167]	All AUTOSAR Basic Software Modules shall provide configuration rules and constraints to enable plausibility checks	[SWS_Dcm_NA_00999]

Requirement	Description	Satisfied by
[SRS_BSW_00168]	SW components shall be tested by a function defined in a common API in the Basis-SW	[SWS_Dcm_NA_00999]
[SRS_BSW_00170]	The AUTOSAR SW Components shall provide information about their dependency from faults, signal qualities, driver demands	[SWS_Dcm_NA_00999]
[SRS_BSW_00171]	Optional functionality of a Basic-SW component that is not required in the ECU shall be configurable at pre-compile-time	[SWS_Dcm_NA_00999]
[SRS_BSW_00172]	The scheduling strategy that is built inside the Basic Software Modules shall be compatible with the strategy used in the system	[SWS_Dcm_NA_00999]
[SRS_BSW_00300]	All AUTOSAR Basic Software Modules shall be identified by an unambiguous name	[SWS_Dcm_NA_00999]
[SRS_BSW_00301]	All AUTOSAR Basic Software Modules shall only import the necessary information	[SWS_Dcm_NA_00999]
[SRS_BSW_00304]	All AUTOSAR Basic Software Modules shall use the following data types instead of native C data types	[SWS_Dcm_NA_00999]
[SRS_BSW_00305]	Data types naming convention	[SWS_Dcm_NA_00999]
[SRS_BSW_00306]	AUTOSAR Basic Software Modules shall be compiler and platform independent	[SWS_Dcm_NA_00999]
[SRS_BSW_00307]	Global variables naming convention	[SWS_Dcm_NA_00999]
[SRS_BSW_00308]	AUTOSAR Basic Software Modules shall not define global data in their header files, but in the C file	[SWS_Dcm_NA_00999]
[SRS_BSW_00309]	All AUTOSAR Basic Software Modules shall indicate all global data with read-only purposes by explicitly assigning the const keyword	[SWS_Dcm_NA_00999]
[SRS_BSW_00310]	API naming convention	[SWS_Dcm_NA_00999]
[SRS_BSW_00312]	Shared code shall be reentrant	[SWS_Dcm_NA_00999]
[SRS_BSW_00314]	All internal driver modules shall separate the interrupt frame definition from the service routine	[SWS_Dcm_NA_00999]
[SRS_BSW_00318]	Each AUTOSAR Basic Software Module file shall provide version numbers in the header file	[SWS_Dcm_NA_00999]
[SRS_BSW_00321]	The version numbers of AUTOSAR Basic Software Modules shall be enumerated according specific rules	[SWS_Dcm_NA_00999]

Requirement	Description	Satisfied by
[SRS_BSW_00323]	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	[SWS_Dcm_NA_00999]
[SRS_BSW_00325]	The runtime of interrupt service routines and functions that are running in interrupt context shall be kept short	[SWS_Dcm_NA_00999]
[SRS_BSW_00327]	Error values naming convention	[SWS_Dcm_NA_00999]
[SRS_BSW_00328]	All AUTOSAR Basic Software Modules shall avoid the duplication of code	[SWS_Dcm_NA_00999]
[SRS_BSW_00330]	It shall be allowed to use macros instead of functions where source code is used and runtime is critical	[SWS_Dcm_NA_00999]
[SRS_BSW_00331]	All Basic Software Modules shall strictly separate error and status information	[SWS_Dcm_NA_00999]
[SRS_BSW_00333]	For each callback function it shall be specified if it is called from interrupt context or not	[SWS_Dcm_NA_00999]
[SRS_BSW_00334]	All Basic Software Modules shall provide an XML file that contains the meta data	[SWS_Dcm_NA_00999]
[SRS_BSW_00335]	Status values naming convention	[SWS_Dcm_NA_00999]
[SRS_BSW_00336]	Basic SW module shall be able to shutdown	[SWS_Dcm_NA_00999]
[SRS_BSW_00337]	Classification of development errors	[SWS_Dcm_00040]
[SRS_BSW_00339]	Reporting of production relevant error status	[SWS_Dcm_NA_00999]
[SRS_BSW_00341]	Module documentation shall contains all needed informations	[SWS_Dcm_NA_00999]
[SRS_BSW_00342]	It shall be possible to create an AUTOSAR ECU out of modules provided as source code and modules provided as object code, even mixed	[SWS_Dcm_NA_00999]
[SRS_BSW_00343]	The unit of time for specification and configuration of Basic SW modules shall be preferably in physical time unit	[SWS_Dcm_NA_00999]
[SRS_BSW_00344]	BSW Modules shall support link-time configuration	[SWS_Dcm_NA_00999]
[SRS_BSW_00345]	BSW Modules shall support pre-compile configuration	[SWS_Dcm_NA_00999]
[SRS_BSW_00346]	All AUTOSAR Basic Software Modules shall provide at least a basic set of module files	[SWS_Dcm_NA_00999]
[SRS_BSW_00347]	A Naming separation of different instances of BSW drivers shall be in place	[SWS_Dcm_NA_00999]

Requirement	Description	Satisfied by
[SRS_BSW_00350]	All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors.	[SWS_Dcm_NA_00999]
[SRS_BSW_00351]	Encapsulation of compiler specific methods to map objects	[SWS_Dcm_NA_00999]
[SRS_BSW_00353]	All integer type definitions of target and compiler specific scope shall be placed and organized in a single type header	[SWS_Dcm_NA_00999]
[SRS_BSW_00357]	For success/failure of an API call a standard return type shall be defined	[SWS_Dcm_NA_00999]
[SRS_BSW_00358]	The return type of init() functions implemented by AUTOSAR Basic Software Modules shall be void	[SWS_Dcm_NA_00999]
[SRS_BSW_00359]	All AUTOSAR Basic Software Modules callback functions shall avoid return types other than void if possible	[SWS_Dcm_NA_00999]
[SRS_BSW_00360]	AUTOSAR Basic Software Modules callback functions are allowed to have parameters	[SWS_Dcm_NA_00999]
[SRS_BSW_00361]	All mappings of not standardized keywords of compiler specific scope shall be placed and organized in a compiler specific type and keyword header	[SWS_Dcm_NA_00999]
[SRS_BSW_00369]	All AUTOSAR Basic Software Modules shall not return specific development error codes via the API	[SWS_Dcm_00044]
[SRS_BSW_00371]	The passing of function pointers as API parameter is forbidden for all AUTOSAR Basic Software Modules	[SWS_Dcm_NA_00999]
[SRS_BSW_00373]	The main processing function of each AUTOSAR Basic Software Module shall be named according the defined convention	[SWS_Dcm_00053]
[SRS_BSW_00374]	All Basic Software Modules shall provide a readable module vendor identification	[SWS_Dcm_NA_00999]
[SRS_BSW_00375]	Basic Software Modules shall report wake-up reasons	[SWS_Dcm_NA_00999]
[SRS_BSW_00377]	A Basic Software Module can return a module specific types	[SWS_Dcm_NA_00999]
[SRS_BSW_00378]	AUTOSAR shall provide a boolean type	[SWS_Dcm_NA_00999]

Requirement	Description	Satisfied by
[SRS_BSW_00379]	All software modules shall provide a module identifier in the header file and in the module XML description file.	[SWS_Dcm_NA_00999]
[SRS_BSW_00380]	Configuration parameters being stored in memory shall be placed into separate c-files	[SWS_Dcm_NA_00999]
[SRS_BSW_00383]	The Basic Software Module specifications shall specify which other configuration files from other modules they use at least in the description	[SWS_Dcm_NA_00999]
[SRS_BSW_00384]	The Basic Software Module specifications shall specify at least in the description which other modules they require	[SWS_Dcm_NA_00999]
[SRS_BSW_00385]	List possible error notifications	[SWS_Dcm_NA_00999]
[SRS_BSW_00386]	The BSW shall specify the configuration for detecting an error	[SWS_Dcm_NA_00999]
[SRS_BSW_00388]	Containers shall be used to group configuration parameters that are defined for the same object	[SWS_Dcm_NA_00999]
[SRS_BSW_00389]	Containers shall have names	[SWS_Dcm_NA_00999]
[SRS_BSW_00390]	Parameter content shall be unique within the module	[SWS_Dcm_NA_00999]
[SRS_BSW_00392]	Parameters shall have a type	[SWS_Dcm_NA_00999]
[SRS_BSW_00393]	Parameters shall have a range	[SWS_Dcm_NA_00999]
[SRS_BSW_00394]	The Basic Software Module specifications shall specify the scope of the configuration parameters	[SWS_Dcm_NA_00999]
[SRS_BSW_00395]	The Basic Software Module specifications shall list all configuration parameter dependencies	[SWS_Dcm_NA_00999]
[SRS_BSW_00396]	The Basic Software Module specifications shall specify the supported configuration classes for changing values and multiplicities for each parameter/container	[SWS_Dcm_NA_00999]
[SRS_BSW_00397]	The configuration parameters in pre-compile time are fixed before compilation starts	[SWS_Dcm_NA_00999]
[SRS_BSW_00398]	The link-time configuration is achieved on object code basis in the stage after compiling and before linking	[SWS_Dcm_NA_00999]
[SRS_BSW_00399]	Parameter-sets shall be located in a separate segment and shall be loaded after the code	[SWS_Dcm_NA_00999]

Requirement	Description	Satisfied by
[SRS_BSW_00400]	Parameter shall be selected from multiple sets of parameters after code has been loaded and started	[SWS_Dcm_NA_00999]
[SRS_BSW_00401]	Documentation of multiple instances of configuration parameters shall be available	[SWS_Dcm_NA_00999]
[SRS_BSW_00402]	Each module shall provide version information	[SWS_Dcm_NA_00999]
[SRS_BSW_00403]	The Basic Software Module specifications shall specify for each parameter/container whether it supports different values or multiplicity in different configuration sets	[SWS_Dcm_NA_00999]
[SRS_BSW_00404]	BSW Modules shall support post-build configuration	[SWS_Dcm_NA_00999]
[SRS_BSW_00405]	BSW Modules shall support multiple configuration sets	[SWS_Dcm_NA_00999]
[SRS_BSW_00406]	A static status variable denoting if a BSW module is initialized shall be initialized with value 0 before any APIs of the BSW module is called	[SWS_Dcm_NA_00999]
[SRS_BSW_00407]	Each BSW module shall provide a function to read out the version information of a dedicated module implementation	[SWS_Dcm_00065]
[SRS_BSW_00408]	All AUTOSAR Basic Software Modules configuration parameters shall be named according to a specific naming rule	[SWS_Dcm_NA_00999]
[SRS_BSW_00409]	All production code error ID symbols are defined by the Dem module and shall be retrieved by the other BSW modules from Dem configuration	[SWS_Dcm_NA_00999]
[SRS_BSW_00410]	Compiler switches shall have defined values	[SWS_Dcm_NA_00999]
[SRS_BSW_00411]	All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API	[SWS_Dcm_NA_00999]
[SRS_BSW_00413]	An index-based accessing of the instances of BSW modules shall be done	[SWS_Dcm_NA_00999]
[SRS_BSW_00414]	Init functions shall have a pointer to a configuration structure as single parameter	[SWS_Dcm_NA_00999]
[SRS_BSW_00415]	Interfaces which are provided exclusively for one module shall be separated into a dedicated header file	[SWS_Dcm_NA_00999]

Requirement	Description	Satisfied by
[SRS_BSW_00416]	The sequence of modules to be initialized shall be configurable	[SWS_Dcm_NA_00999]
[SRS_BSW_00417]	Software which is not part of the SW-C shall report error events only after the DEM is fully operational.	[SWS_Dcm_NA_00999]
[SRS_BSW_00419]	If a pre-compile time configuration parameter is implemented as "const" it should be placed into a separate c-file	[SWS_Dcm_NA_00999]
[SRS_BSW_00422]	Pre-de-bouncing of error status information is done within the DEM	[SWS_Dcm_NA_00999]
[SRS_BSW_00423]	BSW modules with AUTOSAR interfaces shall be describable with the means of the SW-C Template	[SWS_Dcm_NA_00999]
[SRS_BSW_00424]	BSW module main processing functions shall not be allowed to enter a wait state	[SWS_Dcm_00053]
[SRS_BSW_00425]	The BSW module description template shall provide means to model the defined trigger conditions of schedulable objects	[SWS_Dcm_NA_00999]
[SRS_BSW_00426]	BSW Modules shall ensure data consistency of data which is shared between BSW modules	[SWS_Dcm_NA_00999]
[SRS_BSW_00427]	ISR functions shall be defined and documented in the BSW module description template	[SWS_Dcm_NA_00999]
[SRS_BSW_00428]	A BSW module shall state if its main processing function(s) has to be executed in a specific order or sequence	[SWS_Dcm_NA_00999]
[SRS_BSW_00429]	Access to OS is restricted	[SWS_Dcm_NA_00999]
[SRS_BSW_00432]	Modules should have separate main processing functions for read/receive and write/transmit data path	[SWS_Dcm_NA_00999]
[SRS_BSW_00433]	Main processing functions are only allowed to be called from task bodies provided by the BSW Scheduler	[SWS_Dcm_NA_00999]
[SRS_BSW_00437]	Memory mapping shall provide the possibility to define RAM segments which are not to be initialized during startup	[SWS_Dcm_NA_00999]
[SRS_BSW_00438]	Configuration data shall be defined in a structure	[SWS_Dcm_00037]
[SRS_BSW_00439]	Enable BSW modules to handle interrupts	[SWS_Dcm_NA_00999]

Requirement	Description	Satisfied by
[SRS_BSW_00440]	The callback function invocation by the BSW module shall follow the signature provided by RTE to invoke servers via Rte_Call API	[SWS_Dcm_NA_00999]
[SRS_BSW_00441]	Naming convention for type, macro and function	[SWS_Dcm_NA_00999]
[SRS_BSW_00447]	Standardizing Include file structure of BSW Modules Implementing Autosar Service	[SWS_Dcm_NA_00999]
[SRS_BSW_00448]	Module SWS shall not contain requirements from Other Modules	[SWS_Dcm_NA_00999]
[SRS_BSW_00449]	BSW Service APIs used by Autosar Application Software shall return a Std_ReturnType	[SWS_Dcm_NA_00999]
[SRS_BSW_00450]	A Main function of a un-initialized module shall return immediately	[SWS_Dcm_NA_00999]
[SRS_BSW_00451]	Hardware registers shall be protected if concurrent access to these registers occur	[SWS_Dcm_NA_00999]
[SRS_BSW_00452]	Classification of runtime errors	[SWS_Dcm_01416] [SWS_Dcm_NA_00999]
[SRS_BSW_00453]	BSW Modules shall be harmonized	[SWS_Dcm_NA_00999]
[SRS_BSW_00454]	An alternative interface without a parameter of category DATA_REFERENCE shall be available.	[SWS_Dcm_NA_00999]
[SRS_BSW_00456]	A Header file shall be defined in order to harmonize BSW Modules	[SWS_Dcm_NA_00999]
[SRS_BSW_00457]	Callback functions of Application software components shall be invoked by the Basis SW	[SWS_Dcm_NA_00999]
[SRS_BSW_00458]	Classification of production errors	[SWS_Dcm_NA_00999]
[SRS_BSW_00459]	It shall be possible to concurrently execute a service offered by a BSW module in different partitions	[SWS_Dcm_NA_00999]
[SRS_BSW_00460]	Reentrancy Levels	[SWS_Dcm_NA_00999]
[SRS_BSW_00461]	Modules called by generic modules shall satisfy all interfaces requested by the generic module	[SWS_Dcm_NA_00999]
[SRS_BSW_00462]	All Standardized Autosar Interfaces shall have unique requirement Id / number	[SWS_Dcm_NA_00999]
[SRS_BSW_00463]	Naming convention of callout prototypes	[SWS_Dcm_NA_00999]
[SRS_BSW_00464]	File names shall be considered case sensitive regardless of the filesystem in which they are used	[SWS_Dcm_NA_00999]

Requirement	Description	Satisfied by
[SRS_BSW_00465]	It shall not be allowed to name any two files so that they only differ by the cases of their letters	[SWS_Dcm_NA_00999]
[SRS_BSW_00466]	Classification of extended production errors	[SWS_Dcm_NA_00999]
[SRS_BSW_00467]	The init / deinit services shall only be called by BswM or EcuM	[SWS_Dcm_NA_00999]
[SRS_BSW_00469]	Fault detection and healing of production errors and extended production errors	[SWS_Dcm_NA_00999]
[SRS_BSW_00470]	Execution frequency of production error detection	[SWS_Dcm_NA_00999]
[SRS_BSW_00471]	Do not cause dead-locks on detection of production errors - the ability to heal from previously detected production errors	[SWS_Dcm_NA_00999]
[SRS_BSW_00472]	Avoid detection of two production errors with the same root cause.	[SWS_Dcm_NA_00999]
[SRS_BSW_00473]	Classification of transient faults	[SWS_Dcm_NA_00999]
[SRS_BSW_00477]	The functional interfaces of AUTOSAR BSW modules shall be specified in C90	[SWS_Dcm_NA_00999]
[SRS_BSW_00478]	Timing limits of main functions	[SWS_Dcm_NA_00999]
[SRS_BSW_00479]	Interfaces for handling request from external devices	[SWS_Dcm_NA_00999]
[SRS_BSW_00480]	NullPointer Errors shall follow a naming rule	[SWS_Dcm_NA_00999]
[SRS_BSW_00481]	Invalid configuration set selection errors shall follow a naming rule	[SWS_Dcm_NA_00999]
[SRS_BSW_00482]	Get Version Informationfunction shall follow a naming rule	[SWS_Dcm_00065]

7 Functional specification

7.1 Error Classification

This section describes how the [Dcm](#) module has to treat the several error classes that may happen during the life cycle of the [Dcm](#) module.

Diagnostic-Communication-Errors are handled directly in the ISO-Protocols by NRCs.

[SWS_Dcm_00044] [The error values shall be the unique for all error types. The [Dcm](#) shall use only the values given in this chapter.] ([SRS_BSW_00369](#))

7.1.1 Development Errors

[SWS_Dcm_00040] Development Error Types [The errors and exceptions described in Table 7.1 shall be detectable by the `Dcm` module depending on its build version (development/production mode)] ([SRS_BSW_00337](#))

Type of error	Related error code	Value [hex]
Interface return-value is out of range	DCM_E_INTERFACE_RETURN_VALUE	0x02
Interface: Boundary check of buffers provided by the <code>Dcm</code> failed during interaction with another module (application, <code>Dem</code> , <code>PduR</code> , etc.)	DCM_E_INTERFACE_BUFFER_OVERFLOW	0x03
Internal: <code>Dcm</code> not initialized	DCM_E_UNINIT	0x05
<code>Dcm API</code> function with invalid input parameter	DCM_E_PARAM	0x06
<code>Dcm API</code> service invoked with NULL POINTER as parameter	DCM_E_PARAM_POINTER	0x07
<code>Dcm</code> initialisation failed	DCM_E_INIT_FAILED	0x08
Storing the <code>ProgConditions</code> failed	DCM_E_SET_PROG_CONDITIONS_FAIL	0x09

Table 7.1: Dcm development errors

7.1.2 Runtime Errors

[SWS_Dcm_01416] Runtime Error Types [The errors and exceptions described in Table 7.2 shall be detectable by the `Dcm` module depending on its build version (development/production mode).] ([SRS_BSW_00452](#))

Type of error	Related error code	Value [hex]
The <code>Dcm</code> is getting called with an invalid input parameter value or the <code>Dcm</code> has called an function and this function returns an invalid out parameter or return value.	DCM_E_INVALID_VALUE	0x01
Interface: Timeout occurred during interaction with another module (e.g. maximum number of response pending is reached, refer to [SWS_Dcm_00120])	DCM_E_INTERFACE_TIMEOUT	0x01

Table 7.2: Dcm Runtime errors

7.1.3 Transient Faults

There are no transient faults.

7.1.4 Production Errors

There are no production errors.

7.1.5 Extended Production Errors

There are no extended production errors.

7.2 General design elements

7.2.1 Submodules within the [Dcm](#) module

To define the functionality of the [Dcm](#) module, The [Dcm](#) SWS models the [Dcm](#) module as consisting of the following submodules:

- Diagnostic Session Layer (DSL) submodule: The [DSL](#) submodule ensures data flow concerning diagnostic requests and responses, supervises and guarantees diagnostic protocol timing and manages diagnostic states (especially diagnostic session and security).
- Diagnostic Service Dispatcher (DSD) submodule: The [DSD](#) submodule processes a stream of diagnostic data. The submodule:
 - Receives a new diagnostic request over a network and forwards it to a data processor.
 - Transmits a diagnostic response over a network when triggered by the data processor (e.g. by the [DSP](#) submodule).
- Diagnostic Service Processing (DSP) submodule: The [DSP](#) submodule handles the actual diagnostic service (respectively subservice) requests.

The next graphic gives an overview of the interfaces between the submodules DSP, DSD, and [DSL](#) within the [Dcm](#) module.

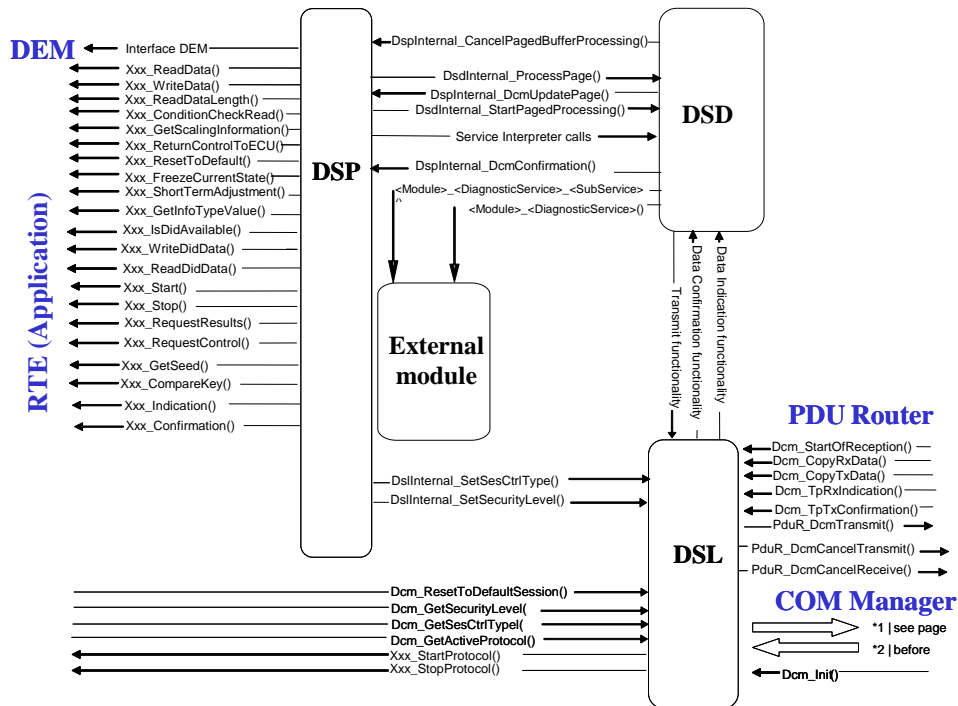


Figure 7.1: Possible interaction between the submodules in the DCM

Note: The implementation of these submodules and the interfaces between them is not mandatory. They are introduced only to improve the readability of the specification.

7.2.2 Negative Response Code (NRC)

The standards defining the UDS Services and OBD Services define the negative response codes (NRCs). The Dcm SWS uses these NRCs in the interfaces between the Dcm and other BSW modules and the SW-Cs. These NRCs are defined in the data type Dcm_NegativeResponseCodeType.

[SWS_Dcm_01075] [The order of the transmitted NRC shall be compliant with the one described in ISO14229-1 [1].]()

7.2.3 Non-volatile information

Several features of the Dcm require non-volatile information to be initialized. AUTOSAR does not describe how this information is accessed or if the information is already available when the Dcm is initialized. Therefore the access for the non-volatile information is implementation specific and has to be ensured during integration.

[SWS_Dcm_00870] [The Dcm shall check if the NvM is read out correctly. If the non-volatile information could not read out correct the Dcm shall start a default reaction. The default reaction is described in the chapter were the usage of the non-volatile data is described.]()

[SWS_Dcm_01048] [If the `Dcm` cancels a service with NvM access, it shall call `NvM_CancelJobs()`.] ()

The service is cancelled either by reaching the maximum number of RCRRP NRCs or by protocol preemption.

7.2.4 Types

[SWS_Dcm_00969] [The `Dcm` shall treat non-integer data types (e.g. `uint8[n]`) either like integer data types of the matching size or leave their contents uninterpreted in case `DcmDspDataEndianness` is configured to `OPAQUE`.] ()

[SWS_Dcm_00970] [The `Dcm` module shall interpret opaque data as `uint8[n]` and shall always map it to an n-bytes sized signal. For opaque data endianness, `DcmDspDataEndianness` has to be configured to `OPAQUE`.] ()

[SWS_Dcm_00971] [The `Dcm` shall extend the endianness conversion defined in [9], to signed data types.] ()

In [9] (Chapter 2.4) the endianness conversion is defined for unsigned data types. The associated configurations can be found in the configuration `?? DcmDspData`.

7.2.4.1 Atomic types overview

Data bit size	ATOMIC						
	1 (Byte aligned)	8	16	32	8	16	32
<code>DcmDspDidDataType</code>	BOOLEAN	UINT8	UINT16	UINT32	SINT8	SINT16	SINT32
<code>DcmDspDataEndianness</code>	N/A	N/A	LE,BE	LE,BE	N/A	LE,BE	LE,BE
<code>DcmDspDataUse Port</code>	S/R, C/S and I/O						
resulting ImplType	boolean	UINT8	UINT16	UINT32	SINT8	SINT16	SINT32

Figure 7.2: Atomic types overview

7.2.4.2 Data array types overview

Data bit size	Field (Static)						Field (Dynamic)
	8-8*N		16-16*N		32-32*N		8-8*N
<code>DcmDspDataByteSize</code>	Any						
<code>DcmDspDidDataType</code>	(size MOD 2)==0						
<code>DcmDspDataEndianness</code>	UINT8_N	SINT8_N	UINT16_N	SINT16_N	UINT32_N	SINT32_N	UINT8_DYN
<code>DcmDspDataUse Port</code>	LE,BE						N/A
<code>DcmDspDataUse Port</code>	S/R, C/S, FNC, NVM						S/R, C/S
resulting ImplType	DataArrayTypeUint8_(Data)	DataArrayTypeSint8_(Data)	DataArrayTypeUint16_(Data)	DataArrayTypeSint16_(Data)	DataArrayTypeUint32_(Data)	DataArrayTypeSint32_(Data)	DataArrayTypeUint8_(Data)

Figure 7.3: Data array types overview

7.2.4.3 Data types constraints

[SWS_Dcm_CONSTR_6002] Existence of size parameter [`DcmDspDataByteSize` shall be present if `DcmDspDataType` is set to: `UINT8_N`, `SINT8_N`, `UINT16_N`, `SINT16_N`, `UINT32_N`, `SINT32_N` or `UINT8_DYN`.]()

Note: `DcmDspDataByteSize` is not required for primitive datatypes

[SWS_Dcm_CONSTR_6035] Restrictions on size parameter for 16 Bit arrays [`DcmDspDataByteSize` shall be a multiple of 2 if the value is greater than 2 and `DcmDspDataType` is `UINT16_N` or `SINT16_N`.]()

[SWS_Dcm_CONSTR_6036] Restrictions on size parameter for 32 Bit arrays [`DcmDspDataByteSize` shall be a multiple of 4 if the value is greater than 4 and `DcmDspDataType` is `UINT32_N` or `SINT32_N`.]()

[SWS_Dcm_CONSTR_6008] Define the usage of `DcmDspRoutineParameterSize` parameter [`DcmDspRoutineParameterSize` is only required if `DcmDspRoutineSignalType` is set to `SINT8_N`, `SINT16_N`, `SINT32_N`, `UINT8_N`, `UINT16_N`, `UINT32_N` or `VARIABLE_LENGTH`.]()

[SWS_Dcm_CONSTR_6011] Only last parameters in `RID` may have a variable length [`DcmDspRoutineSignalType` with `VARIABLE_LENGTH` is only valid for the last signal.]()

[SWS_Dcm_CONSTR_6012] Existence of size parameter [`DcmDspPidDataByteSize` shall be present if `DcmDspPidDataType` is set to: `UINT8_N`, `SINT8_N`, `UINT16_N`, `SINT16_N`, `UINT32_N` or `SINT32_N`.]()

Note: `DcmDspPidDataByteSize` is not required for primitive datatypes

[SWS_Dcm_CONSTR_6040] Restrictions on size parameter for 16 Bit arrays [`DcmDspPidDataByteSize` shall be a multiple of 2 if the value is greater than 2 and `DcmDspPIDDataType` is `UINT16_N` or `SINT16_N`.]()

[SWS_Dcm_CONSTR_6041] Restrictions on size parameter for 32 Bit arrays [`DcmDspPidDataByteSize` shall be a multiple of 4 if the value is greater than 4 and `DcmDspPIDDataType` is `UINT32_N` or `SINT32_N`.]()

`UINT8` shall be used as (implementation) data type for bit lengths between 1 and 8

[SWS_Dcm_CONSTR_6038] Restrictions on datatype usage [`DcmDspDataType` shall be `UINT8_N`, in case `DcmDspDataUsePort` is equal to `USE_BLOCK_ID`.]()

[SWS_Dcm_CONSTR_6026] Usage of variable data length in case of S/R communication, NvRam access or ECU signal access [In case `DcmDspDataUsePort` is set to {`USE_DATA_SENDER_RECEIVER`, `USE_DATA_SENDER_RECEIVER_AS_SERVICE`, `USE_BLOCK_ID`, `USE_ECU_SIGNAL`}, the usage of variable data length shall be not allowed.]()

[SWS_Dcm_CONSTR_6031] [The `DcmDspData.SHORT-NAME` and `DcmDspPidData.SHORT-NAME` shall be distinct.]()

Note: Variable data length is only possible with UINT8 arrays with `DcmDspDataType` set to `UINT8_DYN`.

7.2.4.4 Dcm_OpStatusType

For the operation using the `Dcm_OpStatusType`, the `Dcm` shall work as follow :
[SWS_Dcm_00527] [At first call of an operation using the `Dcm_OpStatusType`, the `Dcm` call the operation with `OpStatus = DCM_INITIAL`.]()

[SWS_Dcm_00528] [If the value `DCM_E_FORCE_RCRRP` is returned from an operation using `Dcm_OpStatusType`, the `Dcm` shall invoke the transmit request for RCR-RP (NRC 0x78 transmission) and the `Dcm` shall not realize further invocation of the operation till RCR-RP is transmitted.]()

[SWS_Dcm_00529] [After transmit confirmation of a RCR-RP transmitted on the context of [SWS_Dcm_00528], the `Dcm` calls, from `Dcm_MainFunction` (due to call context), the operation again with `OpStatus = DCM_FORCE_RCRRP_OK`.]()

[SWS_Dcm_00530] [If a `DCM_E_PENDING` value is returned from an operation using the `Dcm_OpStatusType`, the `Dcm` call the operation on each `Dcm_MainFunction` call with `OpStatus = DCM_PENDING` as long as `DCM_E_PENDING` is returned.]()

7.2.4.5 Dcm_Cemr_{DID}Type

For ease of use in SWC, the `Dcm` generates a symbolic name to access the `CEMR` bit according to `Dcm_Cemr_{DID}Type` (see [SWS_Dcm_91087]) for each data element used in a `DID` with IO control. The SWC can work only with the generated symbolic value of the bitfield text table to mask out a certain bit. This helps to avoid confusion, while the first `RID` bit on the network controls the first parameter of the `DID` but that Bit is the most significant Bit in the MSB. The symbolic name also helps if the size of the `RID` is changing. In that case the bitmask changes as well, but the symbolic value is always updated.

7.3 Diagnostic Session Layer (DSL)

7.3.1 Introduction

[SWS_Dcm_00030] [All functional areas of the `DSL` submodule shall be in conformance with the specifications ISO14229-1 [1] and the network-independent part of ISO15765-3 [4].] (*RS_Diag_04003*, *RS_Diag_04015*)

There is no network-dependent functional area in the `DSL` submodule. Within the configuration, some parameters can be set dependent on the network.

7.3.2 Use cases

The [DSL](#) submodule provides the following functionalities:

- Session handling (as required by ISO14229-1 [1] and ISO 15765-3 [4])
- Application layer timing handling (as required by ISO14229-1 [1] and ISO 15765-3 [4])
- Specific response behavior (as required by ISO14229-1 [1] and ISO 15765-3 [4])
- Authentication state handling per diagnostic connection (as required by ISO 14229-1:2018)
 - Provide authentication state per connection
 - Manage authentication state transitions

7.3.3 Interaction with other modules

The [DSL](#) has the following interaction with other modules:

- PduR module
 - PduR module provides data of incoming diagnostic requests.
 - The [DSL](#) submodule triggers output of diagnostic responses.
- [DSD](#) submodule
 - The [DSL](#) submodule informs the [DSD](#) submodule about incoming requests and provides the data.
 - The [DSD](#) submodule triggers output of diagnostic responses.
- SW-Cs / [DSP](#) submodule. The [DSL](#) submodule provides access to security and session state.
- ComM module
 - The [DSL](#) submodule guarantees the communication behavior required by the ComM module

7.3.4 Functional description

7.3.4.1 Overview

The [DSL](#) submodule provides the following functionality:

Request Handling

- Forward requests from the PduR module to the [DSD](#) submodule.

- Concurrent "TesterPresent" ("keep alive logic").

Response Handling

- Forward responses from the [DSD](#) submodule to the PduR module.
- Guarantee response timing to tester.
- Support of periodic transmission.
- Support of ResponseOnEvent (ROE) transmission.
- Support of segmented response.
- Support of ResponsePending response triggered by the application.

Security Level Handling

- Manage security level.

Session State Handling

- Manage session state.
- Keep track of active non-default sessions.
- Allows modifying timings.

Diagnostic Protocol Handling

- Handling of different diagnostic protocols.
- Manage resources.

Communication Mode Handling

- Handling of communication requirements (Full- / Silent- / No Communication).
- Indicating of active / inactive diagnostic.
- Enabling / disabling all kinds of diagnostic transmissions.

7.3.4.2 Forward requests from the PduR module to the [DSD](#) submodule

The PduR module indicates the [Dcm](#) module whenever a reception of new diagnostic request content is started on a [DcmRxPduld](#), which is assigned to the [Dcm](#) module. This is done by calling [Dcm_StartOfReception](#), which inform the [Dcm](#) module of the data size to be received and provides the data of the first frame or single frame, and allows the [Dcm](#) to reject the reception if the data size overflows its buffer size, or if the requested service is not available. The further call to [Dcm_CopyRxData](#) request the [Dcm](#) module to copy the data from the provided buffer to the [Dcm](#) buffer. If the reception of a diagnostic request is finished (when [Dcm_StartOfReception](#) succeeded) the PduR module will call [Dcm_TpRxIndication](#) to give a receive indication to the [Dcm](#) module. The [Dcm](#) shall be able to use generic connections, where the addressing

information is provided to `Dcm` by `Dcm_StartOfReception` via the `MetaData` of the `DcmRxPdu`. This addressing information must be stored and used for the response and for detection of requests from the same tester. see section 7.3.4.5 Generic Connection Handling for further details.

[SWS_Dcm_00111] [The `DSL` submodule shall forward received data to the `DSD` submodule only after a call of `Dcm_TpRxIndication` with parameter `Result = E_OK` (see [SWS_Dcm_00093]).]()

[SWS_Dcm_00241] [As soon as a request message is received (after a call of `Dcm_TpRxIndication` with parameter `Result = E_OK` (see [SWS_Dcm_00093]) and until a call to `Dcm_TpTxConfirmation` (see [SWS_Dcm_00351]) for the associated `TxDcmPduId`), the `DSL` submodule shall block the corresponding `DcmPduId`. During the processing of this request, no other request of the same `DcmDslConnection` (e.g. an enhanced session can be ended by a `OBD` session) can be received, until the corresponding response message is sent and the `DcmPduId` is released again (except for concurrent `TesterPresent` requests).]()

More descriptions of the APIs (prototype, input/output parameter) can be found in the interface description of `PduR` module [10].

It is allowed to have different `DcmPduIds` for different diagnostic communication applications. For example:

- `OBD DcmRxPduId`: for reception of `OBD` requests,
- `OBD DcmTxPduId`: for transmission of `OBD` responses,
- `UDS phys DcmRxPduId`: for reception of `UDS` physically addressed requests,
- `UDS func DcmRxPduId`: for reception of `UDS` functionally addressed requests,
- `UDS DcmTxPduId`: for transmission of `UDS` responses.

Address type (physical/functional addressing) is configured per `DcmRxPduId` (see configuration parameter `DcmDslProtocolRx`). A configuration per `DcmRxPduId` is possible because there will always be different `DcmRxPduId` values for functional and physical receptions, independent of the addressing format of the Transport Layer (extended addressing, normal addressing).

7.3.4.2.1 Dcm_StartOfReception

[SWS_Dcm_00444] [If the requested size is large than the buffer available in the DCM, the function `Dcm_StartOfReception` shall return `BUFREQ_E_OVFL` (see [SWS_Dcm_00094]).]()

[SWS_Dcm_00788] [When processing a diagnostic request and in case `DcmDslDiagRespOnSecondDeclinedRequest` is set to `TRUE`, the `Dcm` module shall return `BUFREQ_OK` on `Dcm_StartOfReception` received on new request using a different `DcmDslConnection`.]()

[SWS_Dcm_00789] [In case [\[SWS_Dcm_00788\]](#), the `Dcm` respond with a `NRC 0x21`] `()`

[SWS_Dcm_00790] [When processing a diagnostic request, the `Dcm` module shall reject (`Dcm_StartOfReception` shall return `BUFREQ_E_NOT_OK`) any new request using a different `DcmDslConnection` in case `DcmDslDiagRespOnSecondDeclinedRequest` is set to `FALSE` until the current diagnostic request processing is over.] `()`

[SWS_Dcm_00557] [When processing a diagnostic request, the `Dcm` module shall reject (`Dcm_StartOfReception` shall return `BUFREQ_E_NOT_OK`) any new diagnostic request with the same `DcmDslConnection` until the current diagnostic request processing is over. Concurrent `TesterPresent` requests will be accepted with a `BUFREQ_OK`, but not further processed, as the running diagnostic request already resets the session timeout timer (`S3Server`).] `()`

[SWS_Dcm_01145] [If the current session is a non-default session and a concurrent `TesterPresent` received on a different `DcmDslConnection`, this request will be accepted with a `BUFREQ_OK`, but not further processed. E.g. it is not resetting the session timeout timer (`S3Server`)] `()`

[SWS_Dcm_01146] [In case of [\[SWS_Dcm_01145\]](#) with reception on a higher priority protocol, this will not lead to protocol preemption.] `()`

[SWS_Dcm_00642] [When the API `Dcm_StartOfReception` is invoked with `TpSduLength` equal to 0, the value `BUFREQ_E_NOT_OK` shall be returned and no further action shall be taken.] ([RS_Diag_04147](#))

[SWS_Dcm_00655] [If the current session is a non-default session and a new diagnostic request with same or lower priority protocol than active one is detected, the `Dcm` shall act according [\[SWS_Dcm_00788\]](#), [\[SWS_Dcm_00789\]](#) and [\[SWS_Dcm_00790\]](#).] `()`

[SWS_Dcm_00656] [If the current session is the default session and a diagnostic request is in execution, for any new diagnostic request with same or lower priority protocol than active one, the `Dcm` shall act according [\[SWS_Dcm_00788\]](#), [\[SWS_Dcm_00789\]](#) and [\[SWS_Dcm_00790\]](#).] `()`

[SWS_Dcm_00833] [`Dcm_StartOfReception ()` shall be callable in interrupt context.] `()`

7.3.4.2.2 Dcm_CopyRxData

[SWS_Dcm_00443] [If `Dcm_StartOfReception` returns `BUFREQ_OK`, the further call to `Dcm_CopyRxData` shall copy the data from the buffer provided in info parameter) to the `Dcm` buffer and update the `bufferSizePtr` parameter with remaining free place in `Dcm` receive buffer after completion of this call.] `()`

[SWS_Dcm_00996] [When the [API Dcm_CopyRxData](#) is invoked with SduLength from info equal to 0, the value BUFREQ_OK shall be returned and bufferSizePtr shall be filled with the remaining size of the Rx buffer.]()

Note: The size of the Rx buffer is based on the buffer length, which is returned in the parameter RxBufferSizePtr of [API Dcm_StartOfReception](#). **[SWS_Dcm_00342]** [After starting to copy the received data (see [\[SWS_Dcm_00443\]](#)), the Dcm module shall not access the receive buffer until it is notified by the service [Dcm_TpRxIndication](#) about the successful completion or unsuccessful termination of the reception.]()

Note: [Dcm_TpRxIndication](#) is only expected when [Dcm_StartOfReception](#) succeeded

[SWS_Dcm_00831] [[Dcm_CopyRxData](#) shall be callable in interrupt context.]()

7.3.4.2.3 Dcm_TpRxIndication

[SWS_Dcm_00344] [If [Dcm_TpRxIndication](#) is called with parameter Result different from E_OK, then the Dcm module shall not evaluate the buffer assigned to the I-PDU, which is referenced in parameter DcmRxPdulId.]()

Rationale for [\[SWS_Dcm_00344\]](#): It is undefined which part of the buffer contains valid data in this case

[SWS_Dcm_00345] [[Dcm_TpRxIndication](#) shall be callable in interrupt context.]()

7.3.4.3 Concurrent "TesterPresent" ("keep alive logic")

It is possible, that functional "TesterPresent" commands are sent by the tester in parallel to physical requests/responses. This is called "keep alive logic" in ISO14229-1 [1]. This functional "TesterPresent" will be received on a separate DcmRxPdulId (UDS func DcmRxPdulId), which is belonging to the same [DcmDslConnection](#) as the physical request. A Dcm-internal receive buffer which is not configured explicitly, is used in this case. Due to that reason, the functional TesterPresent (and only functional TesterPresent without response) is handled in the following way:

[SWS_Dcm_00112] [When the PduR module calls [Dcm_TpRxIndication](#) with parameter Result=E_OK (see [\[SWS_Dcm_00093\]](#)) and if the request is a "TesterPresent" command with "suppressPosRspMsgIndicationBit" set to TRUE (SID equal to 0x3E, subfunction equal to 0x80), the DSL submodule shall reset the session timeout timer (S3Server).]()

[SWS_Dcm_00113] [When the PduR module calls [Dcm_TpRxIndication](#) with parameter Result = E_OK (see [\[SWS_Dcm_00093\]](#)) and if the request is a "TesterPresent" command with "suppressPosRspMsgIndicationBit" set to TRUE (SID equal to

0x3E, subfunction equal to 0x80), the `DSL` submodule shall not forward this request to the `DSD` submodule for further interpretation. `]()`

Rationale for `[SWS_Dcm_00113]`: Because of bypassing the functional "TesterPresent" in the `DSL` submodule, the `Dcm` module is able to receive and process next physical requests without any delay.

[SWS_Dcm_01168] [The `Dcm` shall handle a tester present request as concurrent request only if it was received on a functional address with "suppressPosRspMsgIndicationBit" set to TRUE.] `()`

7.3.4.3.1 Dcm_CopyTxData

If the copied data is smaller than the length requested to transmit within the service `PduR_DcmTransmit()` the `Dcm` module will be requested by the service `Dcm_CopyTxData` to provide another data when the current copied data have been transmitted.

[SWS_Dcm_00346] [If the function `Dcm_CopyTxData` is called and the `Dcm` module successfully copied the data in the buffer provided in info parameter, then the function shall return `BUFREQ_OK`.] `()`

[SWS_Dcm_00350] [Caveats of `Dcm_CopyTxData`:

- The value of parameter `availableDataPtr` of function `Dcm_CopyTxData` shall not exceed the number of Bytes still to be sent.
- If this service returns `BUFREQ_E_NOT_OK` the transmit requests issued by calling the service `PduR_DcmTransmit()` is still not finished. A final confirmation (indicating an error with call of service `Dcm_TpTxConfirmation`) is required to finish this service and to be able to start another transmission (call to `PduR_DcmTransmit()`). So it is up to the transport protocol to confirm the abort of transmission.

`]()`

[SWS_Dcm_00832] [`Dcm_CopyTxData` shall be callable in interrupt context.] `()`

7.3.4.3.2 Dcm_TpTxConfirmation

[SWS_Dcm_00352] [If the function `Dcm_TpTxConfirmation` is called, then the `Dcm` module shall unlock the transmit buffer.] `()`

[SWS_Dcm_00353] [If the function `Dcm_TpTxConfirmation` is called, then the `Dcm` module shall stop error handling (Page buffer timeout, $P2ServerMax/P2*ServerMax$ timeout).] `()`

[SWS_Dcm_00354] [`Dcm_TpTxConfirmation` shall be callable in interrupt context (e.g. from a transmit interrupt)] `()`

For transmission via FlexRay the following restriction has to be considered: Since the FlexRay Specification does not mandate the existence of a transmit interrupt, the exact meaning of this confirmation (i.e. "transfer into the FlexRay controller's send buffer" OR "transmission onto the FlexRay network") depends on the capabilities of the FlexRay communication controller and the configuration of the FlexRay Interface.

7.3.4.4 Forward responses from the DSD submodule to the PduR module

[SWS_Dcm_00114] [The [DSD](#) submodule shall request the [DSL](#) submodule for transmission of responses.] ()

[SWS_Dcm_00115] [When the diagnostic response of a [DcmDslMainConnection](#) is ready, the [DSL](#) submodule shall trigger the transmission of the diagnostic response to the PduR module by calling [PduR_DcmTransmit\(\)](#) using the corresponding [DcmDslProtocolTxPduRef](#) parameter as PduId.] ()

[SWS_Dcm_01072] [In case of [PeriodicTransmission](#), the [Dcm](#) shall provide in the call to [PduR_DcmTransmit\(\)](#) the full payload data and expect no call to [Dcm_CopyTxData](#).] ()

[SWS_Dcm_01073] [In case of [PeriodicTransmission](#), the [Dcm](#) will be called for periodic transmission with [Dcm_TxConfirmation](#) to indicate the transmission result.] ()

Responses are sent with the [DcmTxPduId](#), which is linked in the [Dcm](#) module configuration to the [DcmRxPduId](#), i.e. the [ID](#) the request was received with (see configuration parameter [DcmDslProtocolTx](#)) Within [PduR_DcmTransmit\(\)](#) only the length information and, for generic connections, the addressing information, is given to the PduR module. After the [Dcm](#) module has called successfully [PduR_DcmTransmit\(\)](#), the PduR module will call [Dcm_CopyTxData](#) to request the [Dcm](#) module to provide the data to be transmitted and will call [Dcm_TpTxConfirmation](#) after the complete [PDU](#) has successfully been transmitted or an error occurred. see section 7.3.4.5 "Generic Connection Handling for further details on address information handling within generic connections".

[SWS_Dcm_00117] [If the [DSL](#) submodule receives a confirmation after the complete [Dcm PDU](#) has successfully been transmitted or an error occurred by a call of [Dcm_TpTxConfirmation](#), then the [DSL](#) submodule shall forward this confirmation to the [DSD](#) submodule.] ()

[SWS_Dcm_00118] [In case of a failed transmission (failed [PduR_DcmTransmit\(\)](#) request) or error confirmation ([Dcm_TpTxConfirmation](#) with error), the [DSD](#) submodule shall not repeat the diagnostic response transmission.] ()

Note: [Dcm_TpTxConfirmation](#) is only expected when [PduR_DcmTransmit](#) succeeded.

[SWS_Dcm_01166] [If the Multiplicity of [DcmDslProtocolTx](#) is set to "0" the [Dcm](#) shall process the received diagnostic request without sending a response.] ()

More descriptions of the APIs (prototype, input/output parameter) can be found in the interface description of the PduR module [10].

7.3.4.5 Generic Connection Handling

The `Dcm` shall be able to handle generic connections, identified by `DcmPdu`s with `MetaDataItems` of type `SOURCE_ADDRESS_16` and `TARGET_ADDRESS_16`. These connections carry the actual tester address at run time. Please note that this address is not provided to the application. If the application needs to discern different testers, separate connections have to be created. Generic connections are supported for diagnostics over IP and FlexRay diagnostics, and `CAN` diagnostics using normal fixed or mixed 29 bit addressing formats according to ISO15765-2 [11]. Depending on the actual layout of the `CAN` IDs, generic connections could also be used for extended or normal and mixed 11 bit addressing formats. The `Dcm` is not aware of the actual addressing format used by `CanTp`. Several connections may reference the same `DcmPdu`.

[SWS_Dcm_CONSTR_6044] [Generic connections shall be consistent. This means that the `MetaDataItems` and the `PduLength` of all referenced PDUs of a `DcmDslConnection` (`DcmDslProtocolRxPduRef`, `DcmDslProtocolTxPduRef`, `DcmDslPeriodicTxPduRef`, `DcmDslRoeTxPduRef`) are identical.]()

[SWS_Dcm_00848] [The source address of diagnostic requests received via a generic connection must be stored. It is provided in the `MetaDataItem SOURCE_ADDRESS_16` provided via `Dcm_StartOfReception`.]()

[SWS_Dcm_00849] Target address for generic connection transmission [If the `Dcm` is about to send a response, response on event, or periodic message for a generic connection request, the `Dcm` shall set `TARGET_ADDRESS_16` to the value of the stored source address in the `MetaDataPtr` in the `PduR_DcmTransmit`.] (*RS_Diag_04153*)

[SWS_Dcm_01429] [The source address of diagnostic requests received via a generic connection shall be provided in the parameter `TesterSourceAddress` to the application [SWS_Dcm_01339], [SWS_Dcm_01340], [SWS_Dcm_01341], [SWS_Dcm_01342], [SWS_Dcm_00692], [SWS_Dcm_00694], [SWS_Dcm_00340], [SWS_Dcm_00698].]()

[SWS_Dcm_01347] [The target address of diagnostic requests received via a generic connection can be provided in the `MetaDataItem TARGET_ADDRESS_16` received via `Dcm_StartOfReception`. In this case, the `Dcm` shall ignore physical requests where the target address is not equal to the configured ECU address `DcmDspProtocolEcuAddr`.] (*RS_Diag_04153*)

[SWS_Dcm_01348] [The source address of the response transmitted via generic connections can be read from the configuration parameter `DcmDspProtocolEcuAddr`. It shall be provided to `PduR_DcmTransmit` in the `MetaDataItem SOURCE_ADDRESS_16`, if that is configured for the transmit PDU.] (*RS_Diag_04153*)

Note: If different source addresses are required for certain transmitted diagnostic messages of the same `DcmDslProtocolRow`, the `MetaDataItem SOURCE_ADDRESS_16` can be omitted from the PDUs, and the address can then be configured in the lower layers. The same is possible for physical requests, where the `TARGET_ADDRESS_16` can be omitted from the PDUs.

7.3.4.6 Guarantee timing to tester by sending busy responses

[SWS_Dcm_00024] [If the Application (or the `DSP` submodule) is able to perform a requested diagnostic task, but needs additional time to finish the task and prepare the response, then the `DSL` submodule shall send a negative response with `NRC 0x78` (Response pending) when reaching the response time (`DcmDspSessionP2ServerMax - DcmTimStrP2ServerAdjust` respectively `DcmDspSessionP2StarServerMax - DcmTimStrP2StarServerAdjust`).] (*RS_Diag_04016*)

Rationale for **[SWS_Dcm_00024]**: The `DSL` submodule guarantees the response timing to tester.

[SWS_Dcm_00119] [The `DSL` submodule shall send negative responses as required in **[SWS_Dcm_00024]** from a separate buffer.] ()

Rationale for **[SWS_Dcm_00119]**: This is needed in order to avoid overwriting the ongoing processing of requests, e.g. the application already prepared response contents in the diagnostic buffer. The number of negative responses with `NRC 0x78` (response pending) for one diagnostic request can be limited by the configuration parameter `DcmDslDiagRespMaxNumRespPend` to avoid endless `NRC 0x78` transmission in case of an application deadlock.

[SWS_Dcm_01567] [The maximum number of negative responses with `NRC 0x78` can be configured using the optional configuration parameter `DcmDslDiagRespMaxNumRespPend` (see `ECUC_Dcm_00693`). If this parameter is not configured, the default amount of negative responses with `NRC 0x78` is infinite.] ()

7.3.4.7 Support of periodic transmission

The `UDS` service `ReadDataByPeriodicIdentifier (0x2A)` allows the tester to request the periodic transmission of data record values from the ECU identified by one or more `periodicDataIdentifiers`.

[SWS_Dcm_00122] [The `Dcm` module shall send responses for periodic transmissions using a separate protocol and a separate buffer of configurable size.] ()

The `DcmDslPeriodicTransmissionConRef` configuration parameter allows linking the protocol used to receive the periodic transmission request / transmit the periodic transmission response to the protocol used for the transmission of the periodic transmission messages. Note that multiple `DcmTxPdualds` can be assigned to the periodic

transmission protocol. The `Dcm` module respects several restrictions according to the communication mode:

[SWS_Dcm_00123] [Periodic transmission communication shall only take place in Full Communication Mode.] ()

Periodic transmission events can occur when not in Full Communication Mode. So the following requirement exists:

[SWS_Dcm_00125] [The `Dcm` module shall discard periodic transmission events beside Full Communication Mode and shall not queue it for transmission.] ()

[SWS_Dcm_00126] [Periodic transmission events shall not activate the Full Communication Mode.] ()

7.3.4.8 Support of ROE transmission

With the `UDS` Service `ResponseOnEvent` (0x86), a tester requests an ECU to start or stop transmission of responses initiated by a specified event. Upon registering an event for transmission, the tester also specifies the corresponding service to respond to (e.g: `UDS` Service `ReadDataByIdentifier` 0x22).

[SWS_Dcm_00595] [The `ROE` functionality is enabled only if the container `DcmDslResponseOnEvent` exists.] ()

7.3.4.8.1 ResponseOnEvent StateChar

[SWS_Dcm_00871] [The `Dcm` shall support several `RoeEvents`. Each `RoeEvent` can have the states "ROE cleared", "ROE stopped" and "ROE started". The transitions from state to state are described in the following section. The Labels in the figure below represents the numbers of the sections.] ()

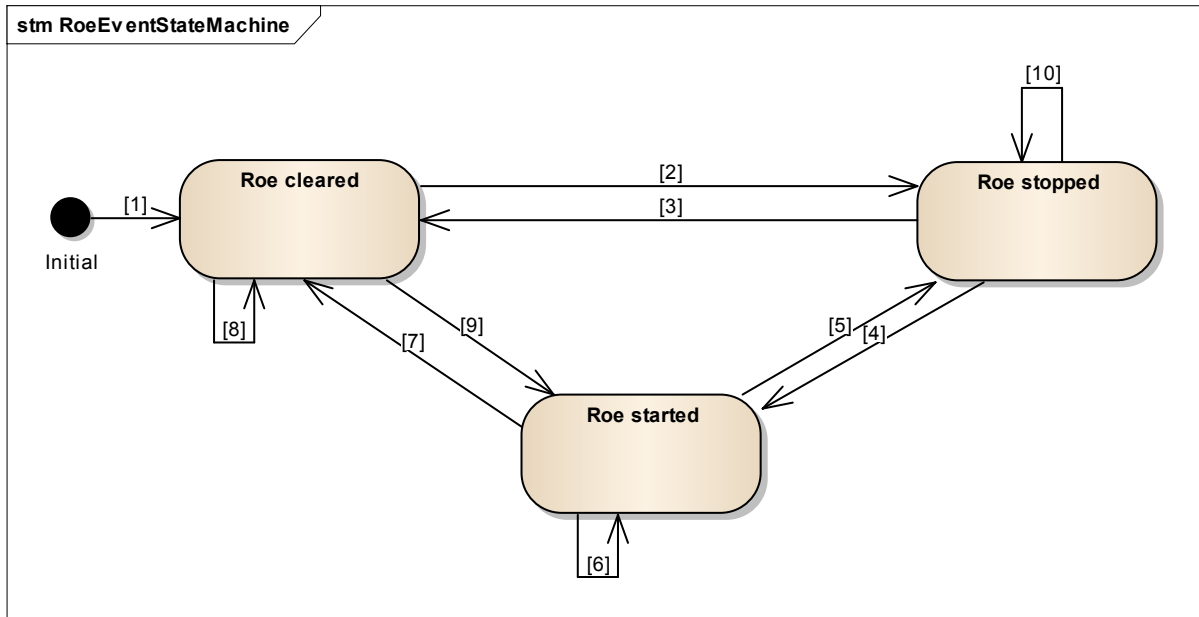


Figure 7.4: RoeEvent State Chart

7.3.4.8.1.1 Initializing Dcm (1)

[SWS_Dcm_00872] [The *Dcm* changes the state of each event to 'ROE cleared' state during *Dcm_Init*.]()

7.3.4.8.1.2 Transition from 'ROE cleared' to 'ROE stopped' (2)

[SWS_Dcm_00873] [By receiving a valid *ROE* setup request, the *RoeEvent* which is addressed in the request changes to the 'ROE stopped' state (see Table 2).]()

[SWS_Dcm_00874] [If the *RoeEvent* was setup with the *StorageState* set to 'storeEvent' and no *StartResponseOnEvent* with *StorageState* set to 'storeEvent' and an *EventWindowTime* which is active over power cycles or *clearResponseOnEvent* has been received afterwards the *Dcm* will change to 'ROE stopped' state as soon as the non-volatile information is available.]()

Note: If an Event is initialized once with *StorageState* set to 'StoreEvent', it will stay initialized until it is cleared by a *ClearResponseOnEvent* request (see also [SWS_Dcm_00897]).

[SWS_Dcm_00951] [If for a *RoeEvent* the configuration parameter *DcmDspRoeInitialEventStatus* is set to *DCM_ROE_STOPPED*, the *Dcm* will switch to 'ROE stopped' state immediatly in the initialisation.]()

Note: *DcmDspRoeInitialEventStatus* set defines an initialisation of a *RoeEvent* by configuration.

7.3.4.8.1.3 Transition from 'ROE stopped' to 'ROE cleared' (3)

[SWS_Dcm_00875] [By receiving a valid [ROE](#) request with the sub-function clearResponseOnEvent (0x06) the RoeEvents change to the 'ROE cleared' state.]()

7.3.4.8.1.4 Transition from 'ROE stopped' to 'ROE started' (4)

[SWS_Dcm_00876] [By receiving a valid [ROE](#) request with the sub-function startResponseOnEvent (0x05) all stopped RoeEvents change to the 'ROE started' state.]()

[SWS_Dcm_00902] [All RoeEvents which have been in 'ROE started' state when leaving the default session shall change back into 'ROE started' state when (re-) entering the default session.]()

[SWS_Dcm_00965] [If a valid StartResponseOnEvent request is received with a storageState set to StoreEvent and the EventWindowTime supports the StorageState in a previous power cycle, the RoeEvent shall change from 'ROE stopped' state to 'ROE started' state as soon as the non-volatile data is available. (This ROEEvent was set to 'ROE stopped' according to [\[SWS_Dcm_00951\]](#)).]()

7.3.4.8.1.5 Transition from 'ROE started' to 'ROE stopped' (5)

[SWS_Dcm_00877] [By receiving a valid [ROE](#) request with the sub-function stopResponseOnEvent (0x00) the stopped RoeEvents change to the 'ROE stopped' state.]()

[SWS_Dcm_00878] [When the eventWindowTime times out the stopped RoeEvents change to the 'ROE stopped' state.]()

[SWS_Dcm_00879] [By leaving the current session all started RoeEvents shall change to the 'ROE stopped' state.]()

Note: RoeEvents are stopped when the current session is left, independent if the session changes from a non-default session to the same or a different non-default session. By leaving the default session the current active RoeEvents are stopped and stored (in order to be re-started as soon the session changes back to the default session (see [\[SWS_Dcm_00902\]](#)).

[SWS_Dcm_00952] [If a [ROE](#) request is received with the sub-function OnDTCStatusChange and the RoeEvent is 'ROE started', the RoeEvent for OnDTCStatusChange changes to 'ROE stopped' state and the ServiceToRespondTo shall be triggered by the DTCStatusMask which is set by the new request.]()

7.3.4.8.1.6 Transition from 'ROE started' to 'ROE started' (6)

[SWS_Dcm_00880] [By receiving a valid ROE request with the sub-function StartResponseOnEvent (0x05) the Dcm answers positively and stays in 'ROE started' state. .] ()

7.3.4.8.1.7 Transition from 'ROE started' to 'ROE cleared' (7)

[SWS_Dcm_00884] [By receiving a valid ROE request with the sub-function clearResponseOnEvent (0x06) all started RoeEvents change to the 'ROE cleared' state.] ()

7.3.4.8.1.8 Transition from 'ROE cleared' to 'ROE cleared' (8)

[SWS_Dcm_00885] [If all RoeEvents are in 'ROE cleared' state and a valid stopResponseOnEvent (0x00) request is received the Dcm shall reject the request with a negative Response with NRC 0x24 (requestSequenceError).] ()

[SWS_Dcm_00886] [If all RoeEvents are in 'ROE cleared' state and a valid StartResponseOnEvent (0x05) request is received the Dcm shall reject the request with a negative Response with NRC 0x24 (requestSequenceError).] ()

[SWS_Dcm_00887] [If all RoeEvents are in 'ROE cleared' state and a valid clearResponseOnEvent (0x06) request is received the Dcm answers positively and the RoeEvents stay in 'ROEcleared' state.)] ()

[SWS_Dcm_00888] [If the non-volatile data could not be read correctly, all RoeEvents in 'ROE cleared' state remain in 'ROE cleared' state.] ()

7.3.4.8.1.9 Transition from 'ROE cleared' to 'ROE started' (9)

[SWS_Dcm_00889] [If the EventWindowTime is active over power cycles and not timed out, the Dcm shall reactivate all RoeEvents which were active in the default session during the last power cycle as soon as the non-volatile information is available.] ()

[SWS_Dcm_00890] [If a valid StartResponseOnEvent request is received with a storageState set to StoreEvent and the EventWindowTime supports the StorageState in a previous power cycle, the RoeEvent shall change to 'ROE started' state as soon as the non-volatile data is available.] ()

7.3.4.8.1.10 Transition from 'ROE stopped' to 'ROE stopped' (10)

[SWS_Dcm_00891] [If a RoeEvent is in 'ROE stopped' state and a valid stopResponseOnEvent (0x00) request is received the Dcm shall respond positively to the request and stay in the 'ROE stopped' state.] ()

[SWS_Dcm_00953] [If a ROE request is received with the sub-function OnDTCStatusChange and the RoeEvent is already 'ROE stopped' the RoeEvent for OnDTCStatusChange shall stay in 'ROE stopped' state but the event logic shall be updated with the newly received DTCStatusMask.] ()

7.3.4.8.2 ROE sub-functions

[SWS_Dcm_00892] [The Dcm shall support all ROE sub-functions marked as supported in Table 7.3.] ()

Sub function ID	Sub-function name	Kind of sub-function	ServiceTo RespondTo	Support status
0x00/0x40	stopResponseOnEvent	Control		Supported
0x01/0x41	onDTCStatusChange	Setup	0x19, 0x0E	Supported
0x02/0x42	onTimerInterrupt	Setup		Not supported
0x03/0x43	onChangeOfDataIdentifier	Setup	0x22	Supported
0x04	reportActivatedEvents	Control		Supported
0x05/0x45	StartResponseOnEvent	Control		Supported
0x06/0x46	clearResponseOnEvent	Control		Supported
0x07/0x47	onComparisonOfValues	Setup		Not supported
Other	OEM Specific	Setup		Not supported

Table 7.3: Supported sub function of Response on Event (0x86)

Note: If a user wants to support a sub-function with StorageState bit set, then it has to be explicitly configured in the DSD. The Dcm will not mask the StorageState bit internally.

[SWS_Dcm_00893] [For each setup sub function the Dcm shall only support the one fixed ServiceToRespondTo. The supported ServiceToRespondTo is listed in the table referred in [SWS_Dcm_00892].] ()

7.3.4.8.3 EventWindowTime and StorageState

The EventWindowTime and StorageState are mandatory parameter in every ROE request. They can be contradicting between the setup request and the related control request.

[SWS_Dcm_00903] [The Dcm shall evaluate the EventWindowTime from the setup request.] ()

[SWS_Dcm_00894] [The Dcm shall support in general the EventWindowTimes defined in Table 7.4.] ()

Value	Name	Active over PowerCycles
0x02	Infinity	Storage State
0x03	CurrentCycle	No
0x04	CurrentAndFollowingCycle	Yes

Table 7.4: Supported ROE EventWindowTime

[SWS_Dcm_00895] [The configuration parameter `DcmDspRoeEventWindowTime` shall contain a list of all EventWindowTimes supported for this specific Ecu.]()

[SWS_Dcm_00896] [If the Roe request contains a different EventWindowTime than configured in `DcmDspRoeEventWindowTime` the Dcm shall reject the request with a negative response with the NRC 0x31 (RequestOutOfRange).]()

[SWS_Dcm_01076] [If the Roe request has a storageState equal to storeEvent and contains an EventWindowTime that is not infinite, the Dcm shall reject the request with a negative response with the NRC 0x31 (RequestOutOfRange).]()

[SWS_Dcm_00897] [If a RoeEvent is setup with StorageState set to storeEvent the initialization shall be stored non-volatile to be restored in every following driving cycle until it is cleared (see [SWS_Dcm_00874]).]()

[SWS_Dcm_00898] [A RoeEvent shall change to 'ROE started' state at the beginning of each following power cycle until a stopResponseOnEvent request with storage StorageState set to StoreEvent is received if the RoeEvent fulfills all following conditions :

- The RoeEvent was started in default session
- The StartResponseOnEventRequest has a storageState set to 'StoreEvent'
- The setup request has the EventWindowTime infinity and the storageState was set to 'StoreEvent'.

]()

[SWS_Dcm_00905] [The EventWindowTime will end at the end of the current power cycle if all of the following conditions are fulfilled:

- The EventWindowTime is set to infinity (0x02) during the setup request
- The RoeEvent was started in default-session
- The storageState was not set in the StartResponseOnEvent request

]()

[SWS_Dcm_00900] [If a RoeEvent set up with the EventWindowTime set to CurrentAndFollowingCycle is started in default session, the EventWindowTime shall end at the end of the next power cycle or with a clearResponseOnEvent/stopResponseOnEvent request.]()

[SWS_Dcm_00901] [If a RoeEvent set up with the EventWindowTime set to CurrentCycle is started in default session, the EventWindowTime shall end at the end of the current power cycle or with a clearResponseOnEvent/stopResponseOnEvent.]()

[SWS_Dcm_00906] [If ResponseOnEvent is started in a non-default session, the EventWindowTime ends if one of the following conditions is fulfilled:

- The power cycle ends
- Receiving a clearResponseOnEvent request
- Receiving a stopResponseOnEvent request
- With any session change.

]()

[SWS_Dcm_00907] [If the EventWindowTime times out and the power cycle is not ended, the Dcm shall send a final positive Response to the setup request.]()

For the EventWindowTime infinity (0x02), ThisCycle (0x03), ThisAndNextCycle (0x04) the Dcm will not send a final response because these EventWindow Times will end at the end of an power cycle. There will also no final response if the session changes or the service is stopped with a 'stopResponseOnEvent' subfunction.

7.3.4.8.4 Pre-configuration of ResponseOnEvent

[SWS_Dcm_00908] [The Dcm shall only support Roe requests which where pre-configured in the configuration.]()

Note: The pre-configuration gives the Dcm the freedom to optimized not configured requests.

[SWS_Dcm_00909] [The Dcm supports the configuration container DcmDspRoe to configure all supported ResponseOnEvent setup requests.]()

[SWS_Dcm_00954] Pre-configuration of ROE events [If DcmDspRoeInitialEventStatus is set to DCM_ROE_STOPPED, the Dcm shall behave according RoeEvent set-up:

- StorageState set to "StoreEvent"
- EventWindowTime set to "infinity"
- DTCStatusMask set to value configured in DcmDspRoeDTCStatusMask in case of onDTCStatusChange and
- DID set to the value given with DcmDspRoeDidRef in case of onChangeOfDataIdentifier

]()

[SWS_Dcm_01323] [Likewise, when responding to the reportActivatedEvents (0x04) subfunction of the ResponseOnEvent (0x86) service, preconfigured events shall have the storageState bit set within the corresponding eventTypeOfActiveEvent byte.]()

According to [SWS_Dcm_00954] and [SWS_Dcm_00897], the pre-configuration of RoeEvents shall behave the same like received a received setup and start request in previous driving cycles. If the storageState is set in the start/stop/clearedResponseOn-EventRequest the pre configuration will be replaced with the newly received request.

7.3.4.8.5 Handling of event-trigger

7.3.4.8.5.1 ROE event-trigger onDTCStatusChange (0x01)

If a RoeEvent is in 'ROE started' state and it is configured to onDTCStatusChange (see container `DcmDspRoeEvent`), the `Dcm` triggers a ServiceToResponseTo as soon as the `Dem` is reporting a DTCStatusChange which fits to the requested DTCStatusMask. According to [SWS_Dcm_00909], the `Dcm` only supports preconfigured ROE requests. Therefore the container `DcmDspRoeOnDTCStatusChange` needs to be configured if onDTCStatusChange shall be used.

[SWS_Dcm_00912] [If the state of one RoeEvent that is configured for onDTC-StatusChange changes to 'ROE started' the `Dcm` shall evaluate the callback `Dcm_DemTriggerOnDTCStatus.`]()

[SWS_Dcm_00913] [If the state of the RoeEvent, configured to OnDTCStatusChange, leaves 'ROE started' the `Dcm` shall ignore the callback `Dcm_DemTriggerOnDTCSta-tus.`]()

[SWS_Dcm_01410] [In case a request to clear the EventMemory is processed, the `Dcm` shall ignore the callback `Dcm_DemTriggerOnDTCStatus.`]()

[SWS_Dcm_00914] [If the state of the RoeEvent is 'ROE started' for the sub-function OnDTCStatusChange shall trigger a serviceToRespondTo if `Dcm_DemTrig-gerOnDTCStatus` is called and the DTCStatusNew fits to the corresponding DTCSta-tusMask.]()

[SWS_Dcm_00915] [If an event is trigger for onDTCStatusChange, the `Dcm` shall exe-cute a serviceToResponseTo 0x19 0x0E, if the DTCStatusNew fits to the corresponding DTCStatusMask.]()

[SWS_Dcm_CONSTR_6054] Existence of DTCStatusMask [`DcmDspRoeDTCSta-tusMask` shall be present if `DcmDspRoeInitialEventStatus` is set to `DCM_ROE_-STOPPED.`]()

7.3.4.8.5.2 ROE event-trigger onChangeOfDataIdentifier (0x03)

If a RoeEvent is in 'ROE started' state and it is configured to onChangeOfDataIdentifier (see container [DcmDspRoeEvent](#)), the Dcm triggers a ServiceToResponseTo as soon as a SWC or a CDD is reporting a change of the DID referenced by [DcmDspRoeDidRef](#) (SWC or CCD reports DID change by call of [Dcm_TriggerOnEvent](#)). According to [[SWS_Dcm_00909](#)], the Dcm only supports preconfigured ROE requests. Therefore the Did in the ROE setup request with onChangeOfDataIdentifier has to be linked as [DcmDspRoeDidRef](#) in the onChangeOfDataIdentifier configuration.

[SWS_Dcm_00918] [If a ResponseOnEvent is requested as onChangeOfDataIdentifier and the requested Did is not referred as [DcmDspRoeDidRef](#) for any [DcmDspRoeEvent](#) the Dcm shall reject the request with a negative response with NRC 0x31 RequestOutOfRange.]()

[SWS_Dcm_00920] [If [Dcm_TriggerOnEvent](#) is called and the passed RoeEvent is active, the Dcm shall trigger an Event for this RoeEvent.]()

[SWS_Dcm_00921] [If an event is triggered for onChangeOfDataIdentifier, the Dcm shall execute a serviceToResponseTo 0x22 with the Did which is referred for this RoeEvent ([DcmDspRoeDidRef](#)).]()

7.3.4.8.6 Trigger a ServiceToRespondTo

[SWS_Dcm_00922] [If a ServiceToRespondTo is triggered by a RoeEvent the Dcm shall execute the ServiceToRespondTo as normal diagnostic service according to the figure 'General server response behavior' of ISO14229-1 [1].]()

[SWS_Dcm_00558] [If a ServiceToRespondTo is triggered while the Dcm is already executing a request on a different diagnostic Protocol the Dcm shall postpone the ServiceToRespondTo until the execution of the service is finalized.]()

[SWS_Dcm_00923] [The Dcm shall only process the last ServiceToRespondTo. If already a ServiceToRespondTo is postponed due to another service execution the new respond shall overwrite the previous trigger.]()

[SWS_Dcm_00924] [If a ServiceToRespondTo is executed while a Request on a different diagnostic protocol is received the ServiceToRespondTo shall be canceled.]()

[SWS_Dcm_00925] [If ServiceToRespondTo are pending when the RoeEvent changes to the 'ROE cleared' state or 'ROE stopped' state the pending RoeEvent will be removed.]()

[SWS_Dcm_00127] [If the UDS service ResponseOnEvent (0x86) is received with the subservice StartResponseOnEvent, then the DSP sub-module shall store the respective configured connectionId of the received RxPduld for all RoeEvents which will be started until the eventWindowTime times out.]()

[SWS_Dcm_00128] [The **DSP** submodule shall forward this stored connectionId as parameter in the `DslInternal_ResponseOnOneEvent()` function, where it is used to trigger a `ServiceToRespondTo`.]()

Note: The `Dcm` stores the connectionId of the protocol where the **ROE** request is received, independent if the `ServiceToResponseTo` is sent to a same or a different `TxPduId`. The connectionId links always the correct `TxPduId`, because there is only one `TxPduId` for `ServiceToRespondTo` linked to one protocol (see ConfigurationParameter `DcmDslROEConnectionRef`). If `RoeEvents` are active over power cycles the connectionId needs to be stored over power cycles.

7.3.4.8.7 Send a ServiceToRespondTo

The `Dcm` supports the transmission from `ServiceToResponseTo` on the same `TxPduId` like the **ROE** response is send (TYPE 1) or on a different `TxPduId` (TYPE 2).

[SWS_Dcm_00131] [The configured protocol buffer shall be used for transmission of the **ROE** messages (as the reception shall use a separate protocol, a separate buffer needs to be used for reception).]()

[SWS_Dcm_00926] [If a **ROE** request is received on a protocol `DcmDslMainConnection`, the `Dcm` shall send the `ServiceToRespondTo` on the protocol which is referred as `DcmDslROEConnectionRef`.]()

Note: if the `EventWindowTime` is active over more than this power cycle, the `Dcm` has to store the protocol where the event was started.**[SWS_Dcm_00927]** [If the referred Protocol for `ResponseOnEvent` (`DcmDslROEConnectionRef`) is configured for TYPE1 the `Dcm` shall send the `ServiceToRespondTo` to the same `TxPduId` as the **ROE** response is send to.]()

[SWS_Dcm_00928] [If the referred Protocol for `ResponseOnEvent` (`DcmDslROEConnectionRef`) is configured for TYPE2 the `Dcm` shall send the `ServiceToRespondTo` to the configured `TxPduId` (see configuration parameter `DcmDslRoeTxPduRef`).]()

[SWS_Dcm_00132] [The content of the `pMsgContext` pointer (**ROE** message) shall be copied into the buffer.]()

[SWS_Dcm_00133] [**ROE** communication shall only be performed in Full Communication Mode. The `Dcm` shall check the communication mode of the `DcmDslProtocolComMChannelRef` in the `DcmDslMainConnection`.]()

[SWS_Dcm_00134] [**ROE** events shall be disabled in any other Communication Mode except for the Full Communication Mode.]()

[SWS_Dcm_00135] [**ROE** events occurring in a communication mode different from Full Communication Mode shall be discarded and not queued for later transmission.]()

[SWS_Dcm_00136] [**ROE** events requested by the Application shall not activate the Full Communication Mode.]()

[SWS_Dcm_01534] Authentication check for service to respond to [On transmission of the service to respond to, the Dcm shall perform the service authentication checks and send a positive response only for services that have granted access to that connection.] (*RS_Diag_04230*)

[SWS_Dcm_CONSTR_6025] Reference to DcmDslResponseOnEvent connection [Only one `DcmDslROEConnectionRef` shall reference `DcmDslResponseOnEvent` connection.] ()

[SWS_Dcm_CONSTR_6056] Dependency for DcmDslProtocolTransType [`DcmDslProtocolTransType` shall be only present if the `Dcm_ProtocolType` is configured to `DCM_ROE_ON_CAN` or `DCM_ROE_ON_FLEXRAY` or `DCM_ROE_ON_IP`.] ()

7.3.4.8.7.1 Roe transmission cycle

[SWS_Dcm_00601] [The `Dcm` module shall respect a minimum time between two (2) consecutive Roe transmissions (see configuration parameter `DcmDspRoeInterMessageTime`)] ()

7.3.4.8.8 ResponseOnEvent in multiple client environments

[SWS_Dcm_00929] [If at least one `RoeEvent` is in 'ROE started' state the `Dcm` shall always process `ROE` request with the sub-function `clearResponseOnEvent` independent of the `DcmDslProtocol` where the request is received.] ()

[SWS_Dcm_00930] [If at least one `RoeEvent` is in 'ROE started' state the `Dcm` shall always process `ROE` request with the sub-function `stopResponseOnEvent` independent of the `DcmDslProtocol` where the request is received.] ()

[SWS_Dcm_00940] [If at least one `RoeEvent` is in 'ROE started' state the `Dcm` shall reject all `ROE` request received on a different `DcmDslProtocol` than the protocol where the `RoeEvents` were started with an `NRC 0x22` (`ConditionsNotCorrect`), except for [\[SWS_Dcm_00929\]](#) and [\[SWS_Dcm_00930\]](#).] ()

[SWS_Dcm_01045] [Only `TYPE2` messages will support parallel execution of Diagnostic response.] ()

7.3.4.9 Support of segmented response (paged-buffer)

[SWS_Dcm_00028] [If enabled (`DcmPagedBufferEnabled=TRUE`), the `Dcm` module shall provide a mechanism to send responses larger than the configured and allocated diagnostic buffer.] ()

[SWS_Dcm_CONSTR_6055] Dependency for `DcmDslProtocolMaximumResponseSize` [`DcmDslProtocolMaximumResponseSize` shall be only present if `DcmPagedBufferEnabled` is set to TRUE.]()

[SWS_Dcm_01058] [If `DcmPagedBufferEnabled` == TRUE and the generated Response for a Request is longer than `DcmDslProtocolMaximumResponseSize`, the `Dcm` shall respond with `NRC 0x14 (DCM_E_RESPONSETOOLONG)`.]()

[SWS_Dcm_01059] [If `DcmPagedBufferEnabled` == FALSE and the generated Response for a Request is longer than `Dcm_MsgContextType` structure element `resMaxDataLen`, the `Dcm` shall respond with `NRC 0x14 (DCM_E_RESPONSETOOLONG)` .]()

With paged-buffer handling the ECU is not forced to provide a buffer, which is as large as the maximum length of response. Please note:

- paged-buffer handling is for transmit only - no support for reception.
- paged-buffer handling is not available for the Application (DCM-internal use only).

[SWS_Dcm_01186] [The `Dcm` shall provide the correct amount of Data requested by the `TP` or return `BUFREQ_E_BUSY` in case the requested amount of data is not available.](*RS_Diag_04147*)

Note: In case the requested amount of data is not available, the `Dcm` should fill up the paged buffer immediately.

7.3.4.10 Support of ResponsePending response triggered by the Application

In some cases, e.g. in case of routine execution, the Application needs to request an immediate `NRC 0x78 (Response pending)`, which shall be sent immediately and not just before reaching the response time (`P2ServerMax` respectively `P2*ServerMax`).

When the `Dcm` module calls an operation and gets an error status `DCM_E_FORCE_RCRRP`, the `DSL` submodule will trigger the transmission of a negative response with `NRC 0x78 (Response pending)`. This response needs to be sent from a separate buffer, in order to avoid overwriting the ongoing processing of the request.

7.3.4.11 Manage security level

[SWS_Dcm_00020] [The `DSL` submodule shall save the level of the current active security level.](*RS_Diag_04005*)

For accessing this level, the `DSL` submodule provides interfaces to:

- get the current active security level: `Dcm_GetSecurityLevel`
- set a new security level: `DslInternal_SetSecurityLevel()`

[SWS_Dcm_00033] [During `Dcm` initialization the security level is set to the value 0x00 (DCM_SEC_LEV_LOCKED).] (*SRS_BSW_00101*, *RS_Diag_04005*)

[SWS_Dcm_00139] [The `DSL` shall reset the security level to the value 0x00 (i.e. the security is enabled) under one of the following conditions: - if a transition from any diagnostic session other than the `defaultSession` to another session other than the `defaultSession` (including the currently active diagnostic session) is performed or - if a transition from any diagnostic session other than the `defaultSession` to the `defaultSession` (`DslInternal_SetSecurityLevel()`) (initiated by `UDS` Service `DiagnosticSessionControl` (0x10) or `S3Server` timeout) is performed.] ()

Only one security level can be active at a time.

[SWS_Dcm_01329] [On every security level change the `Dcm` shall update the `ModDeclarationGroup` `DcmSecurityAccess` with the new security level.] ()

[SWS_Dcm_CONSTR_6083] Dependency on `DcmDspSecurityAttemptCounterEnabled` [If `DcmDspSecurityNumAttDelay` is not configured, the `DcmDspSecurityAttemptCounterEnabled` on the same `DcmDspSecurityRow` shall be set to FALSE.] (*RS_Diag_04005*)

7.3.4.11.1 Initialization sequence

[SWS_Dcm_01154] [At initialization, for each `DcmDspSecurityRow` entry for which the `DcmDspSecurityAttemptCounterEnabled` configuration parameter is set to TRUE, the corresponding `Xxx_GetSecurityAttemptCounter` shall be called in order to get the value of the `AttemptCounter` for each of these `DcmDspSecurityRow` entries.] ()

[SWS_Dcm_01156] [If `Xxx_GetSecurityAttemptCounter` has returned `E_NOT_OK` the attempt counter shall be set to the value configured in `DcmDspSecurityNumAttDelay` of the according `SecurityLevel`.] ()

[SWS_Dcm_01351] [If any `Xxx_GetSecurityAttemptCounter` operation returns a `DCM_E_PENDING` value, the `Dcm` shall interrupt calling the `Xxx_GetSecurityAttemptCounter()` in order to resume this chain of calls within the next `Dcm_MainFunction()` cycle.] ()

Note: this may be the case when these values are stored within some specific non-volatile memory.

[SWS_Dcm_CONSTR_6076] Dependency for `DcmDspSecurityGetAttemptCounterFnc` [`DcmDspSecurityGetAttemptCounterFnc` shall be present only if `DcmDspSecurityUsePort` is set to `USE_ASYNC_FNC` and `DcmDspSecurityAttemptCounterEnabled` is set to TRUE.] ()

[SWS_Dcm_01352] [If the delay after the first call of the `Dcm_MainFunction()` which is configured in `DcmDspSecurityMaxAttemptCounterReadoutTime` has been reached and all the `Xxx_GetSecurityAttemptCounter` have not been called

yet (i.e. one operation has returned a DCM_E_PENDING status in the previous Dcm_MainFunction() cycle), the pending operation shall be cancelled by a call with the OpStatus set to DCM_CANCEL.]()

[SWS_Dcm_01353] [In the conditions of [SWS_Dcm_01352], the AttemptCounters of remaining security levels (which have not been obtained via the calls to their Xxx_GetSecurityAttemptCounter) shall be initialized with the value configured in DcmDspSecurityNumAttDelay of the according SecurityLevel.]()

[SWS_Dcm_01354] [While not all Xxx_GetSecurityAttemptCounter operations have returned a final status and the operation chain has not been cancelled, the conditionsNotCorrect (0x22) NRC shall be returned to any SecurityAccess (0x27) request-Seed subfunction request.]()

[SWS_Dcm_01355] [Once all the AttemptCounter values have been successfully or unsuccessfully retrieved (all the Xxx_GetSecurityAttemptCounter() operations have been executed and have returned a final, non-PENDING error value or the operation chain has been cancelled), if at least one of the restored AttemptCounter values is greater than or equal to the DcmDspSecurityNumAttDelay configured for its corresponding DcmDspSecurityRow, the Dcm shall start the SecurityDelayTimer with the higher value of DcmDspSecurityDelayTimeOnBoot / DcmDspSecurityDelayTime of the according DcmDspSecurityRow.]()

[SWS_Dcm_01356] [A timer (DcmDspSecurityDelayTime, DcmDspSecurityMaxAttemptCounterReadoutTime) which is configured with 0 shall be considered to have timed out instantaneously when it is started, i.e. shall have no delay effect.]()

[SWS_Dcm_CONSTR_6074] Dependency for DcmDspSecurityMaxAttemptCounterReadoutTime [DcmDspSecurityMaxAttemptCounterReadoutTime shall be a multiple and at minimum equal to DcmTaskTime.]()

7.3.4.11.2 AttemptCounter update

[SWS_Dcm_01357] [A successful sendKey subfunction request or an expired SecurityDelayTimer shall reset that security level's specific AttemptCounter.]()

[SWS_Dcm_01155] [The Dcm shall call Xxx_SetSecurityAttemptCounter() (in case the configuration parameter DcmDspSecurityAttemptCounterEnabled for the according DcmDspSecurityRow is set to TRUE) when the Dcm has changed the attempt counter to inform the application about the counter change.]()

[SWS_Dcm_CONSTR_6078] Dependency for DcmDspSecuritySetAttemptCounterFnc [DcmDspSecuritySetAttemptCounterFnc shall be present only if DcmDspSecurityUsePort is set to USE_ASYNC_FNC and the DcmDspSecurityAttemptCounterEnabled set to TRUE.]()

7.3.4.12 Manage session state

[SWS_Dcm_00022] [The `DSL` submodule shall save the state of the current active session.] ([RS_Diag_04006](#))

For accessing this variable, the `DSL` submodule provides interfaces to:

- get the current active session: `Dcm_GetSesCtrlType`
- set a new session: `DslInternal_SetSesCtrlType()`

[SWS_Dcm_00034] [During `Dcm` initialization, the session state is set to the value 0x01 ("DefaultSession").] ([SRS_BSW_00101](#))

[SWS_Dcm_01062] [The call to `Dcm_ResetToDefaultSession` allows the application to reset the current session to Default session and invokes the mode switch of the `ModeDeclarationGroupPrototype DcmDiagnosticSessionControl` by calling `SchM_Switch_<bsnp>_DcmDiagnosticSessionControl(RTE_MODE_DcmDiagnosticSessionControl_DCM_DEFAULT_SESSION)`.] ()

Example: Automatic termination of an extended diagnostic session upon exceeding of a speed limit.

7.3.4.13 Manage authentication state

The `Dcm` provides means for authenticated diagnostics. The `DSL` sub-module provides an authentication state per diagnostic connection. It initializes this state upon startup and takes care about fallback into non-authenticated states if the connection is idle for some time.

[SWS_Dcm_01477] Authentication state per connection [The `Dcm` shall provide an authentication state per configured `DcmDslConnection`.] ([RS_Diag_04230](#))

[SWS_Dcm_01478] Mode declaration group per authentication state [The `Dcm` shall provide the state of each authentication state via the `ModeDeclarationGroupPrototype DcmAuthentication_<ConnectionName>`.] ([RS_Diag_04230](#))

The `Dcm` maintains an authentication state and mirrors this state to the mode declaration group `DcmAuthentication_<ConnectionName>`. This mode declaration group is intended to be changed only by the `Dcm`, however applications changing this state have no influence on the `Dcm` authentication state.

[SWS_Dcm_01479] Authentication states [The `Dcm` shall support per connection the two authentication states:] ([RS_Diag_04230](#))

- deauthenticated
- authenticated

Upon startup, the Dcm is in deauthenticated state or restores the persisted state. A transition to authenticated state can only be done after the client successfully executed the authentication sequence. In some use cases as in production, a frequent power-on/power off sequence is performed. To keep the achieved authentication state over the power off, there is a dedicated mode rule requesting the Dcm to persist the authenticated state.

[SWS_Dcm_01480] Initialization of authentication state [If `DcmDspAuthenticationPersistStateModeRuleRef` is not configured or the mode rule referenced by `DcmDspAuthenticationPersistStateModeRuleRef` is evaluated to false, the Dcm shall initialize within `Dcm_Init` all authentication states to deauthenticated state.] (*RS_Diag_04230*)

[SWS_Dcm_01481] Initialization of persisted authentication states [If the mode rule referenced by `DcmDspAuthenticationPersistStateModeRuleRef` is evaluated to true, the Dcm shall initialize the persisted authentication state including role and white list on each connection.] (*RS_Diag_04230*)

Transitions between authenticated states are controlled by both DSL and DSP sub-modules. The DSL sub-module is in charge for fallback of authenticated state into deauthenticated state. The DSP sub-module is in charge for transition changes triggered from a client by diagnostic services.

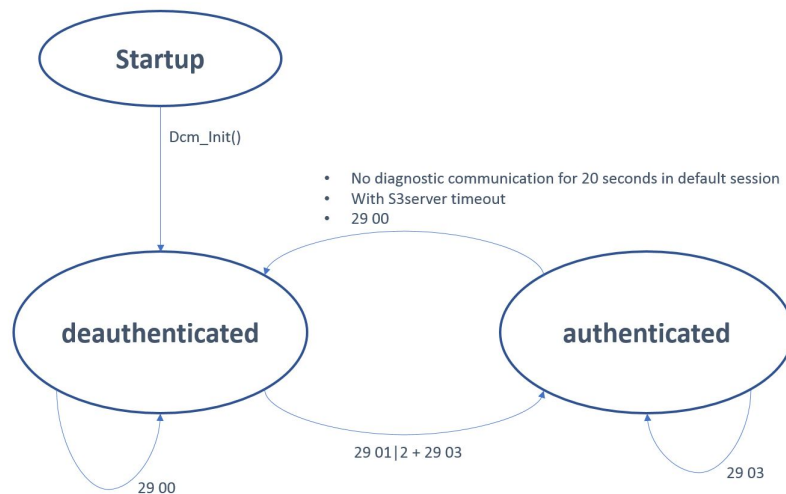


Figure 7.5: Authenticated state transitions without persistent states

[SWS_Dcm_01482] Fallback to deauthenticated session on idle connection [The Dcm shall make a transition from authenticated into deauthenticated state for a configured connection if the following conditions apply:

- The Dcm was in default session when the last diagnostic response was send on that connection and

- `DcmDspAuthenticationDefaultSessionTimeout` is configured and no valid diagnostic request was received on that connection for `DcmDspAuthenticationDefaultSessionTimeout` seconds after the last `Dcm_TpTxConfirmation` on that connection.

]([RS_Diag_04230](#))

[SWS_Dcm_01483] Fallback to deauthenticated session on S3server timeout [If the Dcm is in a non-default session and a S3server timeout occurs, the Dcm shall perform a transition from authenticated into deauthenticated state on the authentication state assigned to that connection which was in a non-default session.]([RS_Diag_04230](#))

[SWS_Dcm_01484] Clearing persisted authentication state [If the authentication state of a connection performs a transition to deauthenticated state, the Dcm shall clear all persisted authentication information on that connection.]([RS_Diag_04230](#))

[SWS_Dcm_01485] Reaction of fallback into deauthenticated state [Upon a transition from authenticated into deauthenticated state, the Dcm shall discard the current role, white list and use the configured deauthentication role from `DcmDspAuthenticationDeauthenticatedRole`.]([RS_Diag_04230](#))

In some use cases, it is desirable that the application set the role instead of using a diagnostic service with its potentially time-consuming certificate parsing. The Dcm provides the API `Dcm_SetDeauthenticatedRole` to overwrite the configured deauthentication role. The overwritten role is only valid in deauthenticated state will not be persisted and is overwritten by a role provided by certificates via service 0x29.

[SWS_Dcm_01486] Default authentication role set from SWC [If a connection is in deauthenticated state and the API `Dcm_SetDeauthenticatedRole` is called, the Dcm shall use the provided deauthenticatedRole as new role per deauthenticated state for this connection.]([RS_Diag_04230](#))

[SWS_Dcm_01487] Setting deauthenticated role by SWC only in deauthenticated state [The Dcm shall process a call of `Dcm_SetDeauthenticatedRole` only if the connection is in deauthenticated state.]([RS_Diag_04230](#))

[SWS_Dcm_01488] Lifetime of deauthenticated role by SWC [A deauthenticated role set by `Dcm_SetDeauthenticatedRole` is discarded when that connection performs a transition authenticated state.]([RS_Diag_04230](#))

[SWS_Dcm_01489] No persistency for deauthenticated roles by SWC [At startup the ECU shall always use the deauthentication state configured in `DcmDspAuthenticationDeauthenticatedRole`.]([RS_Diag_04230](#))

7.3.4.14 Keep track of active non-default sessions

[SWS_Dcm_00140] [Whenever a non-default session is active and when the session timeout (S3Server) is reached without receiving any diagnostic request, the DSL

submodule shall reset to the default session state ("DefaultSession", 0x01) and invoke the mode switch of the ModeDeclarationGroupPrototype DcmDiagnosticSessionControl by calling SchM_Switch_<bsnp>_DcmDiagnosticSessionControl(RTE_MODE_DcmDiagnosticSessionControl_DEFAULT_SESSION) .]()

Note: <bsnp> is the BSW Scheduler Name Prefix

The start / stop of S3Server timeout timer is processed as follows:

[SWS_Dcm_00141] [

Subsequent start:

- Completion of any final response message or an error indication ([Dcm_TpTx-Confirmation](#): confirmation of complete PDU or indication of an error)
- Completion of the requested action in case no response message (positive and negative) is required / allowed.
- Indicates an error during the reception of a multi-frame request message. ([Dcm_TpRxIndication](#): indication of an error)

Subsequent stop:

- Start of a multi-frame request message ([Dcm_StartOfReception](#): indicates start of PDU reception)
- Reception of single-frame request message. ([Dcm_StartOfReception](#): indicates start of PDU reception)

"Start of S3Server" means reset the timer and start counting from the beginning.]()

7.3.4.15 Allow to modify timings

[SWS_Dcm_00027] [The [Dcm](#) module shall handle the following protocol timing parameters in compliance with [4]: P2ServerMin, P2ServerMax, P2*ServerMin, P2*ServerMax, S3Server] ([RS_Diag_04015](#))

[SWS_Dcm_00143] [P2min / P2*min and S3Server shall be set to defined values: P2min = 0ms, P2*min = 0ms, S3Server = 5s.] ([RS_Diag_04015](#))

These protocol timing parameters have influence on the session layer timing (no influence on Transport Layer timing). Some of these timing parameters can be modified while protocol is active with the following means:

- [UDS](#) Service DiagnosticSessionControl (0x10)
- [UDS](#) Service AccessTimingParameter (0x83)

The [DSL](#) submodule provides the following functionalities to modify the timing parameters:

- Provide the active timing parameters,

- Set the new timing parameters. Activation of new timing values is only allowed after sending the response.

7.3.4.16 Handling of different diagnostic protocols

It is necessary to distinguish between different diagnostic protocols (e.g. [OBD](#), enhanced diagnosis ...).

[SWS_Dcm_01365] [If the API [Dcm_GetActiveProtocol](#) is called, the [Dcm](#) shall return the active [UDS](#) protocol as [Dcm_ProtocolType](#) in the [ActiveProtocol](#) parameter.] ([RS_Diag_04163](#))

[SWS_Dcm_01366] [If only an [OBD](#) protocol or no protocol is started and the API [Dcm_GetActiveProtocol](#) is called, the [Dcm](#) shall return with [ActiveProtocol](#) parameter set to `DCM_NO_ACTIVE_PROTOCOL`.] ([RS_Diag_04163](#))

Note: The [Dcm_GetActiveProtocol](#) API doesn't supply information about running [OBD](#) protocols.

7.3.4.16.1 Different service tables

For the different protocols a different set of allowed diagnostic services is valid (e.g. the [UDS](#) commands for the enhanced diagnosis, the [OBD](#) mode services for the [OBD](#) protocol). It is possible to create different service tables and link them to the diagnostic protocol.

[SWS_Dcm_00035] [With every protocol initialization, the [DSL](#) submodule sets a link to the corresponding service table (see configuration parameter [DcmDslProtocol-SIDTable](#)).] ([SRS_BSW_00101](#))

The [DSD](#) submodule uses this link for further processing of diagnostic requests.

7.3.4.16.2 Prioritization of protocol

The configuration parameter [DcmDslProtocolPriority](#) makes it possible to give each protocol its own relative priority. Possible use case: There are ECUs, communicating with a vehicle-internal diagnostic tester (running on enhanced diagnosis) and a vehicle-external [OBD](#) tester. The [OBD](#) communication must have a higher priority than the enhanced diagnosis.

[SWS_Dcm_00015] [A protocol with higher priority is allowed to preempt the already running protocol.] ([RS_Diag_04021](#))

Differentiation of diagnostic protocols is possible, because of different [DcmRxPduId](#) values (configured per protocol, see configuration parameter [DcmDslProtocolRxPduRef](#)) referenced in the protocol configuration.

7.3.4.16.3 Preemption of protocol

[SWS_Dcm_00459] [If a running diagnostic request is preempted by a higher priority request (of another protocol), the DSL submodule shall call all configured Xxx_StopProtocol() functions (see configuration parameter DcmDslCallbackDCMRequestService).]()

[SWS_Dcm_01144] [Protocol preemption can't be activated with a concurrent Tester-Present of a higher priority protocol (see also [SWS_Dcm_01146]).]()

[SWS_Dcm_00079] [In order to cancel pending transmission in lower-layer, related to the lower priority request, the Dcm module shall call PduR_DcmCancelTransmit () with the following parameters: PduId: the id of the Pdu to be canceled]()

[SWS_Dcm_00460] [When PduR_DcmCancelTransmit() returns E_NOT_OK, the Dcm module shall assume that the ongoing transmission cannot be cancelled and shall not retry to cancel the transmit request. The current protocol shall be stopped and the new one started.]()

[SWS_Dcm_01046] [If a running diagnostic request is preempted by a higher priority request (of another protocol), the Dcm shall cancel all external pending operations with Dcm_OpStatus set to DCM_CANCEL]()

[SWS_Dcm_01047] [In case an operation to the Dem is pending and the new request also requires an interaction with the Dem, the Dcm shall accept the new request and call the corresponding Dem API with the parameters from the new request.]()

[SWS_Dcm_00575] [In order to cancel pending reception in lower-layer, related to the lower priority request, the Dcm module shall call PduR_DcmCancelReceive () with the following parameters: PduId: the id of the Pdu to be canceled]()

[SWS_Dcm_00576] [When PduR_DcmCancelReceive () returns E_NOT_OK, the Dcm module shall assume that the ongoing reception cannot be cancelled and shall not retry to cancel the receiverrequest. The current protocol shall be stopped and the new one started.]()

[SWS_Dcm_00625] [A Low-priority or same-priority request can preempt a higher priority protocol if this higher priority protocol is in default session and no active request is in execution phase. In this case the DSL submodule shall call all configured Xxx_StopProtocol() functions (see configuration parameter DcmDslCallbackDCMRequestService).]()

[SWS_Dcm_00728] [The handling of protocols with equal priority shall be possible.]()

[SWS_Dcm_00727] [If a diagnostic request is already running and a second request (ClientB) can not be processed (e.g.due to priority assessment), the response behaviour depends on the configuration option parameter DcmDslDiagRespOnSecondDeclinedRequest (see [SWS_Dcm_00914]_Conf). If this configuration parameter is TRUE, a negative response with NRC 0x21 (BusyRepeatRequest) shall be issued for

the second request (see [SWS_Dcm_00788] and [SWS_Dcm_00789]). If the configuration parameter is FALSE, no response shall be issued (see [SWS_Dcm_00790]).
()

[SWS_Dcm_00729] [In case of multiple clients with different PduIDs which are requesting the same protocol, as all the connections of the same protocol are having the same priority, a second request (with the different RxPduId) will not be processed. If the configuration parameter `DcmDslDiagRespOnSecondDeclinedRequest` is TRUE, a negative response with NRC 0x21 (BusyRepeatRequest) shall be issued for the second request. If the configuration parameter is FALSE, no response shall be issued.] ()

Note:

- A multitude of RxPduIDs may be configured per `DcmDslProtocol`
- These RxPduIDs may be themselves connected to different Testers via the PduR configuration
- This means that many Testers may be configured for the same Protocol
- And this represents a non-UDS extension/use case. In order to have a UDS-compliant flow, there should be one `DcmDslProtocol` instance per Tester.

[SWS_Dcm_01050] [In case of diagnostic parallel requests, with same / lower priority than the active request then the ComM APIs (`ComM_DCM_ActiveDiagnostic`, `ComM_DCM_InactiveDiagnostic`) shall not be called.] ()

7.3.4.16.4 Parallel OBD and UDS protocol processing

[SWS_Dcm_01367] [The `Dcm` shall process incoming OBD requests in parallel to a running UDS request. In this case the protocol priority check according to [SWS_Dcm_00015] is skipped and no protocol pre-emption is done.] (*RS_Diag_04163*)

With the container `DcmDslProtocolRow`, the `Dcm` configuration supports multiple protocols. Each protocol has a configured `DcmDemClientRef` defining the `Dem` client interacting with the `Dem`. This client Id allows the `Dem` to distinguish between concurrent calls of the `Dcm` of the same function or set of functions to process a certain request.

[SWS_Dcm_01369] [While processing a diagnostic request received from a given protocol, the `Dcm` shall determine the `DcmDemClientRef` of the `DcmDslProtocolRow` of the processed protocol. The `Dcm` shall use this value in all `Dem` API calls that have a `ClientId` as parameter.] (*RS_Diag_04162*)

[SWS_Dcm_01370] [The `Dcm` shall internally serialise all asynchronous C/S interface calls by the same port interface between the OBD and UDS protocol processors and return a pending to the re-entrant caller.] (*RS_Diag_04162*)

[SWS_Dcm_01371] [If the `Dcm` received an OBD request and the `Dcm` is processing a diagnostic service in a non-default session, the `Dcm` shall cancel the running UDS

request, make a transition into default session and process the OBD request.](RS_Diag_04162)

[SWS_Dcm_01372] [If the Dcm processes an OBD request and the Dcm is receiving an UDS diagnostic request to change in a non-default session, the Dcm shall delay the UDS request until the OBD service is finished according to [SWS_Dcm_01371]. After the OBD service is finished, the Dcm shall make a transition into the requested non-default session.](RS_Diag_04162)

7.3.4.16.5 Detection of protocol start

[SWS_Dcm_00036] [With first request of a diagnostic protocol, the DSL submodule shall call all configured Xxx_StartProtocol() functions (see configuration parameter DcmDslCallbackDCMRequestService).](SRS_BSW_00101)

Inside this function, the Application can examine the environment conditions and enable/disable further processing of the protocol.

[SWS_Dcm_00144] [After all Xxx_StartProtocol() functions have returned E_OK (meaning all components have allowed the start of the protocol), the default timing parameters are loaded from the default session configuration (see configuration parameter DcmDspSessionRow).](RS_Diag_04015)

[SWS_Dcm_CONSTR_6000] Harmonize the naming between interfaces and modes [The shortname of DcmDspSessionRow shall match names of Dcm_SesCtrlType and of the mode declarations of DcmDiagnosticSessionControl. The "DCM_" prefix is mandatory for all shortnames.]()

[SWS_Dcm_CONSTR_6001] Provide standardized names for ISO standardized diagnostic sessions [The following values of DcmDspSessionLevel which represent ISO defined diagnostic sessions shall be used for the shortname of DcmDspSessionRow:

- 1 DCM_DEFAULT_SESSION
 - 2 DCM_PROGRAMMING_SESSION
 - 3 DCM_EXTENDED_DIAGNOSTIC_SESSION
 - 4 DCM_SAFETY_SYSTEM_DIAGNOSTIC_SESSION
-]()

[SWS_Dcm_00145] [After all Xxx_StartProtocol() functions have returned E_OK (meaning all components have allowed the start of the protocol), the service table is set (see configuration parameter DcmDslProtocolSIDTable).]()

[SWS_Dcm_00146] [After all Xxx_StartProtocol() functions have returned E_OK (meaning all components have allowed the start of the protocol), the security state is reset.]()

[SWS_Dcm_00147] [After all Xxx_StartProtocol() functions have returned E_OK (meaning all components have allowed the start of the protocol), the session state is reset to default session. Furthermore the Dcm module shall

invoke the the mode switch of the ModeDeclarationGroupPrototype DcmDiagnosticSessionControl by calling SchM_Switch_<bsnp>_DcmDiagnosticSessionControl(RTE_MODE_DcmDiagnosticSessionControl_DEFAULT_SESSION) .]()

Note: <bsnp> is the BSW Scheduler Name Prefix

[SWS_Dcm_00674] [If Xxx_StartProtocol() doesn't return E_OK, the Dcm shall return NRC 0x22.]()

7.3.4.16.6 Protocol stop

A protocol stop can appear only in case of protocol preemption (see chapter 7.3.4.16.3 Preemption of protocol).

[SWS_Dcm_00624] [With the reception of Dcm_TpTxConfirmation connected to the response given by the DSL submodule, the Dcm shall not stop the current protocol (no call to xxx_StopProtocol).]()

Note: A protocol (e.g. OBD) will be active till reset or other protocol preempts.

[SWS_Dcm_01190] [If Xxx_StopProtocol() doesn't return E_OK, the Dcm shall return NRC 0x22.]()

7.3.4.17 Manage resources

Due to limited resources, the following points should be considered as hints for the design:

- It is allowed to use and allocate only one diagnostic buffer in the Dcm module. This buffer is then used for processing the diagnostic requests and responses.
- Output of NRC 0x78 (Response pending) responses is done with a separate buffer.
- paged-buffer handling (see [SWS_Dcm_00028]).

7.3.4.18 Communication Mode Handling

Communication Mode Handling is an interface between Dcm and ComM. The ComM informs the Dcm about the current communication state of a channel. The Dcm is calling the ComM about active Diagnostic which shall prevent an Ecu shutdown/sleep.

The status ActiveDiagnostic shows if diagnostic requests shall keep the ECU awake (ActiveDiagnostic == 'DCM_COMM_ACTIVE') or if diagnostic requests shall not prevent an Ecu shutdown/sleep (ActiveDiagnostic == 'DCM_COMM_NOT_ACTIVE'). Application can change the status ActiveDiagnostic regarding to system conditions.

[SWS_Dcm_CONSTR_6027] [The application will inform the `Dcm` by calling `Xxx_SetActiveDiagnostic()` about the `ActiveDiagnostic` status.]()

[SWS_Dcm_01069] [After `Dcm_Init`, the `Dcm` shall set `ActiveDiagnostic` to 'DCM_COMM_ACTIVE'.]()

[SWS_Dcm_01070] [If `Xxx_SetActiveDiagnostic()` is called with 'false' the `Dcm` set `ActiveDiagnostic` to 'DCM_COMM_NOT_ACTIVE'.]()

[SWS_Dcm_01071] [If `Xxx_SetActiveDiagnostic()` is called with 'true' the `Dcm` set `ActiveDiagnostic` to 'DCM_COMM_ACTIVE'.]()

[SWS_Dcm_01142] [The `Dcm` shall wait the Full Communication mode indication from the `ComM` (call to `Dcm_ComM_FullComModeEntered`) before initiating the transmission of the diagnostic answer. The time to wait should be no longer than the `P2ServerMax` calculated from the moment the request was received.]()

[SWS_Dcm_01143] [If the `Dcm` fails to confirm a response pending transmission (`DCM_E_FORCE_RCRRP`) due to **[SWS_Dcm_01142]**, the `Dcm` shall trigger the `Det` error `DCM_E_FORCE_RCRRP_IN_SILENT_COMM`.]()

Note : On the reception side a silent communication mode can lead to the lost of the request in case of segmented transmission.

7.3.4.18.1 No Communication

The `ComM` module will indicate the No Communication Mode to the `Dcm` module by calling `Dcm_ComM_NoComModeEntered`. In response, the `Dcm` will immediately disable all transmissions (see the definition of `Dcm_ComM_NoComModeEntered` for details).

[SWS_Dcm_00148] [`Dcm_ComM_NoComModeEntered` shall disable all kinds of transmissions (receive and transmit) of communication. This means that the message reception and also the message transmission shall be off.]()

[SWS_Dcm_00149] [`Dcm_ComM_NoComModeEntered` shall disable the `ResponseOnEvent` transmissions.]()

[SWS_Dcm_00150] [`Dcm_ComM_NoComModeEntered` shall disable the `periodicId` transmissions (`ReadDataByPeriodicIdentifier`).]()

[SWS_Dcm_00151] [`Dcm_ComM_NoComModeEntered` shall disable normal transmissions.]()

[SWS_Dcm_00152] [After `Dcm_ComM_NoComModeEntered` has been called, the `Dcm` module shall not call the function `PduR_DcmTransmit()`.]()

[SWS_Dcm_01324] [In case `Dcm_ComM_NoComModeEntered` is called with a `NetworkId` for a ComM channel not referenced within the `Dcm` (see configuration parameter `DcmDslProtocolComMChannelRef`), the `Dcm` shall return without performing any further action.]()

7.3.4.18.2 Silent Communication

The ComM module will indicate the Silent Communication Mode to the `Dcm` module by calling `Dcm_ComM_SilentComModeEntered`. In response, the `Dcm` will immediately disable all transmissions (see the definition of `Dcm_ComM_SilentComModeEntered` for details).

[SWS_Dcm_00153] [`Dcm_ComM_SilentComModeEntered` shall disable all transmission. This means that the message transmission shall be off.]()

[SWS_Dcm_00154] [`Dcm_ComM_SilentComModeEntered` shall disable the `ResponseOnEvent` transmissions.]()

[SWS_Dcm_00155] [`Dcm_ComM_SilentComModeEntered` shall disable the periodicId transmissions (`ReadDataByPeriodicIdentifier`) shall be disabled.]()

[SWS_Dcm_00156] [`Dcm_ComM_SilentComModeEntered` shall disable the normal transmissions.]()

[SWS_Dcm_01325] [In case `Dcm_ComM_SilentComModeEntered` is called with a `NetworkId` for a ComM channel not referenced within the `Dcm` (see configuration parameter `DcmDslProtocolComMChannelRef`), the `Dcm` shall return without performing any further action.]()

7.3.4.18.3 Full Communication

The ComM module will indicate the Full Communication Mode to the `Dcm` module by calling `Dcm_ComM_FullComModeEntered`. In response, the `Dcm` will enable all transmissions (see the definition of `Dcm_ComM_FullComModeEntered` for details).

[SWS_Dcm_00157] [`Dcm_ComM_FullComModeEntered` shall enable all kind of communication. This means that the message reception and also the message transmission shall be on.]()

[SWS_Dcm_00159] [`Dcm_ComM_FullComModeEntered` shall enable the `ResponseOnEvent` transmissions.]()

[SWS_Dcm_00160] [`Dcm_ComM_FullComModeEntered` shall enable the periodicId transmissions (`ReadDataByPeriodicIdentifier`).]()

[SWS_Dcm_00161] [`Dcm_ComM_FullComModeEntered` shall enable the normal transmissions.]()

[SWS_Dcm_00162] [After [Dcm_ComM_FullComModeEntered](#) has been called, the [Dcm](#) shall handle the functions [DsInternal_ResponseOnOneDataByPeriodicId\(\)](#) or [DsInternal_ResponseOnOneEvent\(\)](#) without restrictions.]()

[SWS_Dcm_01326] [In case [Dcm_ComM_FullComModeEntered](#) is called with a [NetworkId](#) for a [ComM](#) channel not referenced within the [Dcm](#) (see configuration parameter [DcmDslProtocolComMChannelRef](#)), the [Dcm](#) shall return without performing any further action.]()

7.3.4.18.4 Diagnostic Activation State

The [Dcm](#) notifies the [ComM](#) module about the internal diagnostic state for all networks. There are two options for the diagnostic state on a network. In 'active' diagnostic state, the [Dcm](#) is processing one or more diagnostic requests from this network or the [Dcm](#) is in a non-default session. In 'inactive' diagnostic state, the [Dcm](#) is in default session and is not processing a diagnostic request on that network.

When a network has no communication in progress, the [Dcm](#) will set the diagnostic activation state to 'inactive'. When there is a diagnostic communication on a network the [Dcm](#) sets the diagnostic state to 'active'. In any non-default session, the diagnostic state remains in state 'active'. The communication state can also be controlled by the API [Xxx_SetActiveDiagnostic](#) according to [\[SWS_Dcm_01070\]](#) and [\[SWS_Dcm_01071\]](#).

[SWS_Dcm_01373] [The [Dcm](#) shall go into 'active' diagnostic state on a network, if a diagnostic request is received on a network or the diagnostic session is changed to any non-default session.]([RS_Diag_04006](#))

[SWS_Dcm_01374] [The [Dcm](#) shall go into 'inactive' diagnostic state on a network when the current diagnostic request processing is finished and the [Dcm](#) is not processing a diagnostic request of another protocol on this network and if the [Dcm](#) is in default session.]([RS_Diag_04006](#))

[SWS_Dcm_01375] [The [Dcm](#) shall go into 'inactive' diagnostic state on all networks if a [S3Server](#) timeout occurs and the [Dcm](#) makes a transition into default session.]([RS_Diag_04006](#))

[SWS_Dcm_01376] [If [ActiveDiagnostic](#) is 'DCM_COMM_ACTIVE' and the [Dcm](#) is doing a transition into 'active' diagnostic state of a diagnostic protocol, the [Dcm](#) shall call [ComM_DCM_ActiveDiagnostic\(NetworkId\)](#), with the [networkId](#) associated to the received [Pdu](#) (see [DcmDslProtocolComMChannelRef](#)), with every request, to inform the [ComM](#) module about the need to stay in Full Communication Mode.]([RS_Diag_04006](#))

[SWS_Dcm_01377] [Upon a diagnostic state transition into 'inactive', the [Dcm](#) shall notify the [ComM](#) module about an inactive diagnostic state on a network by calling [ComM_DCM_InactiveDiagnostic\(NetworkId\)](#), with the [networkId](#) associated to the received [Pdu](#) (see [DcmDslProtocolComMChannelRef](#)).]([RS_Diag_04006](#))

[SWS_Dcm_01378] [The definition of a finished diagnostic request according to [\[SWS_Dcm_01374\]](#), shall be as follows:

- the *Dcm* has sent a positive or negative response unequal to NRC 0x78 by receiving the *Dcm_TpTxConfirmation* connected to the response given by the *DSL* submodule
- the *Dcm* has processed the service with *SPRMIB=true* and the positive response was suppressed
- in case of functional addressing, the *Dcm* has processed the service and the negative response was suppressed.

]([RS_Diag_04006](#))

7.4 Diagnostic Service Dispatcher (DSD)

7.4.1 Introduction

The *DSD* submodule is responsible to check the validity of an incoming diagnostic request (Verification of Diagnostic Session/Security Access levels/Application permission) and keeps track of the progress of a service request execution.

[SWS_Dcm_00178] [The *DSD* submodule shall only process valid requests and shall reject invalid ones.]()

7.4.2 Use cases

The following use cases are relevant and are described in detail in the following:

- Receive a request message and transmit a positive response message
- Receive a request message and suppress a positive response
- Receive a request message and suppress a negative response
- Receive a request message and transmit a negative response message
- Send a positive response message without corresponding request
- Segmented Responses

7.4.2.1 Receive a request message and transmit a positive response message

This is the standard use case of normal communication ("ping-pong"). The server receives a diagnostic request message. The *DSD* submodule ensures the validity of the request message. In this use case, the request is valid and the response will be

positive. The request will be forwarded to the appropriate data processor in the [DSP](#) submodule. When the data processor has finished all actions of data processing, it triggers the transmission of the response message by the [DSD](#) submodule.

If the data processor processes the service immediately as part of the request indication function, the data processor can trigger the transmission inside this indication function ("synchronous"). If the processing takes a longer time (e. g. waiting on EEPROM driver), the data processor defers some processing ("asynchronous"). The response pending mechanism is covered by the [DSL](#) submodule. The data processor triggers the transmission explicitly, but from within the data processor's context.

As soon as a request message is received, the corresponding DcmPduld is blocked by the [DSL](#) submodule (see [[SWS_Dcm_00241](#)]). During the processing of this request, no other request of the same protocol type (e.g. an enhanced session can be ended by a [OBD](#) session) can be received, until the corresponding response message is sent and the DcmPduld is released again.

7.4.2.2 Receive a request message and suppress a positive response

This is a sub-use-case of the previous one. Within the [UDS](#) protocol it is possible to suppress the positive response by setting a special bit in the request message (see [[SWS_Dcm_00200](#)]). This special suppression handling is completely performed within the [DSD](#) submodule.

7.4.2.3 Receive a request message and suppress a negative response

In case of functional addressing the [DSD](#) submodule shall suppress the negative response for [NRC](#) 0x11, 0x12, 0x31, 0x7E and 0x7F (see [[SWS_Dcm_00001](#)]).

7.4.2.4 Receive a request message and transmit a negative response message

There are a many different reasons why a request message is rejected and a negative response is to be sent. If a diagnostic request is not valid or if a request may not be executed in the current session, the [DSD](#) submodule will reject the processing and a negative response will be returned.

But there are even many reasons to reject the execution of a well-formed request message, e.g. if the ECU or system state does not allow the execution. In this case, the [DSP](#) submodule will trigger a negative response including an [NRC](#) supplying additional information why this request was rejected.

In case of a request composed of several parameters (e.g. a [UDS](#) Service Read-DataByIdentifier (0x22) request with more than one identifier to read), each parameter is treated separately. And each of these parameters can return an error. This kind of

request returns a positive response if at least one of the parameters was processed successfully.

[SWS_Dcm_00827] [The [DSD](#) sub-module shall check the received diagnostic request in the order given by ISO14229-1 [1]. If one of the computations failed the [Dcm](#) shall stop the execution of the [NRC](#) check sequence then stop or do not start the execution of the received diagnostic request and finally transmit the [NRC](#) for which the computation failed.]()

7.4.2.5 Send a positive response message without corresponding request

There are two services within the [UDS](#) protocol, where multiple responses are sent for only one request. In general, one service is used to enable (and disable) an event- or time-triggered transmission of another service, which again is sent by the ECU without a corresponding request (see ISO14229-1 [1]). These services are:

- [UDS](#) Service ReadDataByPeriodicIdentifier (0x2A). This service allows the client to request the periodic transmission of data record values from the server identified by one or more periodicDataIdentifiers.

Type 2 = UUDT message on a separate DcmTxPduld.

- ResponseOnEvent (0x86). This service requests a server to start or stop transmission of responses on a specified event.

Type 1 = USDT messages on the DcmTxPduld already used for normal diagnostic responses,

Type 2 = USDT messages on separate DcmTxPduld.

For Type 1, the outgoing messages must be synchronized with "normal outgoing messages", which have a higher priority.

This handling is especially controlled by the [DSL](#) submodule. However, the [DSD](#) submodule also provides the possibility to generate a response without a corresponding request.

7.4.2.6 Segmented Responses (paged-buffer)

Within the diagnostic protocol, some services allow to exchange a significant amount of data, e.g. [UDS](#) Service ReadDTCInformation (0x19) and [UDS](#) Service TransferData (0x36).

In the conventional approach, the ECU internal buffer must be large enough to keep the longest data message which is to be exchanged (worst-case) and the complete buffer is filled before the transmission is started.

RAM memory in an ECU often is a critical resource, especially in smaller micros. In a more memory-saving approach, the buffer is filled only partly, transmitted partly and then refilled partly - and so on. This paging mechanism requires only a significantly reduced amount of memory, but demands a well-defined reaction time for buffer refilling. The user can decide whether to use the "linear buffer" or paged-buffer for diagnostics.

7.4.3 Interaction of the DSD with other modules

The DSD submodule is called by the DSL submodule when receiving a diagnostic message and performs the following operations:

- delegates processing of request to the DSP submodule or external modules outside the Dcm
- keeps track of request processing (Return the status on <Module>_<DiagnosticService>() and <Module>_<DiagnosticService>_<SubService>() APIs call or "Service Interpreter calls")
- transmits the response of the Application to the DSL submodule (Transmit functionality)

7.4.3.1 Interaction of the DSD with the DSL main functionality

Direction	Explanation
Bidirectional	Exchange of the Diagnostic Messages (receive/transmit).
DSD submodule to DSL submodule	Obtain latest diagnostic session and latest security level.
DSL submodule to DSD submodule	Confirmation of transmission of Diagnostic Message.

Table 7.5: Interaction between the DSD submodule and the DSL submodule

7.4.3.2 Interaction of the DSD with the DSP

Direction	Explanation
DSD submodule to DSP submodule	- Delegate processing of request. - Confirmation of transmission of Diagnostic Message.
DSP submodule to DSD submodule	- Signal that processing is finished.

Table 7.6: Interaction of the DSD with the DSP

7.4.4 Functional Description of the DSD

7.4.4.1 Support checking the diagnostic service identifier and adapting the diagnostic message

The **DSD** submodule shall be triggered by the **DSL** submodule if a new diagnostic message is recognized. The **DSD** submodule will start processing by analyzing the diagnostic service identifier contained in the received diagnostic message.

[SWS_Dcm_00084] [If configured (configuration parameter `DcmRespondAllRequest=FALSE`), if the **Dcm** module receives a diagnostic request that contains a service **ID** that is in the range from 0x40 to 0x7F or in the range from 0xC0 to 0xFF, the **Dcm** shall not respond to such a request.]()

This range corresponds to the diagnostic response identifier.

[SWS_Dcm_00192] [The **DSD** submodule shall analyze the (incoming) diagnostic message for the diagnostic service identifier (based on first byte of the diagnostic message) and shall check the supported services with the newly received diagnostic service identifier.]()

[SWS_Dcm_00193] [During this check, the **DSD** submodule shall search the newly received diagnostic service identifier in the "Service Identifier Table".]()

For performance reasons it might be necessary that the support check is done with a "lookup table" functionality. In this "Service Identifier Table" all supported Service IDs of the ECU are predefined.

[SWS_Dcm_00195] [The **DSL** submodule shall provide the current "Service Identifier Table".]()

Rationale for **[SWS_Dcm_00195]**: The "Service Identifier Table" and the information about the supported services will be generated out of the configuration. More than one Service Identifier Table can be configured for selection. At one time only one Service Identifier Table can be active.

[SWS_Dcm_00196] [For the check, the **DSD** submodule shall scan the active "Service Identifier Table" for a newly received diagnostic service identifier. If this service identifier is supported and if the configuration parameter `DcmDsdSidTabFnc` (see `ECUC_Dcm_00777`) is not empty, the **DSD** submodule shall call the configured service interface (`<Module>_<DiagnosticService>`). If the configuration parameter is empty, the **Dcm** shall call the internally implemented service interface.]()

The diagnostic service identifier is not supported when it is not included in the "Service Identifier Table".

[SWS_Dcm_00197] [If the newly received diagnostic service identifier is not supported, the **DSD** submodule shall transmit a negative response with **NRC** 0x11 (Service not supported) to the **DSL** submodule.]()

[SWS_Dcm_00198] [The **DSD** submodule shall store the newly received diagnostic service identifier for later use.]()

For example: WriteDataByIdentifier (for writing VIN number):

1. A new diagnostic message is received by the **DSL** submodule. (Diagnostic Message WriteDataByIdentifier = 0x2E, 0xF1, 0x90, 0x57, 0x30, 0x4C, 0x30, 0x30, 0x30, 0x30, 0x34, 0x33, 0x4D, 0x42, 0x35, 0x34, 0x31, 0x33, 0x32, 0x36)
2. The **DSL** submodule indicates a new diagnostic message with the "Data Indication" functionality to the **DSD** submodule. In the diagnostic message buffer the diagnostic message is stored (buffer = 0x2E,0xF1,0x90,...).
3. The **DSD** submodule executes a check of the supported services with the determined service identifier (first byte of buffer 0x2E) on the incoming diagnostic message.
4. The incoming diagnostic message is stored in the **Dcm** variable **Dcm_MsgContextType**.

[SWS_Dcm_CONSTR_6047] [Id of the Service identifier configured in **DcmDsdSidTabServiceId** shall be unique within one **DcmDsdServiceTable**.]()

[SWS_Dcm_00732] [For the first call of <Module>_<DiagnosticService> the opStatus shall be set to DCM_INITIAL.]()

[SWS_Dcm_00733] [The **Dcm** shall not accept further requests (on same or lower priority) while <Module>_<DiagnosticService>() returns DCM_E_PENDING. Dcm-internal timeout handling (based on RCR-RP limitation) may lead to a cancellation of the external diagnostic service processing.]()

[SWS_Dcm_00735] [In case of cancellation the **API** <Module>_<DiagnosticService> is called again with the parameter opStatus set to DCM_CANCEL.]()

7.4.4.2 Handling of "suppressPosRspMsgIndicationBit"

The "suppressPosRspMsgIndicationBit" is part of the subfunction parameter structure (Bit 7 based on second byte of the diagnostic message, see ISO14229-1 [1] Section 6.5: Server response implementation rules).

[SWS_Dcm_00200] [If the "suppressPosRspMsgIndicationBit" is TRUE, the **DSD** submodule shall NOT send a positive response message.](*RS_Diag_04020*)

[SWS_Dcm_00201] [The **DSD** submodule shall remove the "suppressPosRspMsgIndicationBit" (by masking the Bit) from the diagnostic message.]()

[SWS_Dcm_00202] [The **Dcm** module shall transport the information on a suppression of a positive response being active (between the layers) via the parameter **Dcm_MsgContextType**.]()

[SWS_Dcm_00203] [In case of responsePending the `Dcm` module shall clear the "suppressPosRspMsgIndicationBit."] ()

Rationale for [\[SWS_Dcm_00203\]](#): In the described case the final response (negative/positive) is required.

[SWS_Dcm_00204] [The `Dcm` module shall only perform the "suppressPosRspMsgIndicationBit" handling when the configuration parameter `DcmDsdSidTabSubfuncAvail` is set for the newly received service identifier] ()

Note: The "suppressPosRspMsgIndicationBit" handling needs to be considered independent of the processing order in the request (like for RoutineControl service).

Rationale for [\[SWS_Dcm_00204\]](#): The "suppressPosRspMsgIndicationBit" is only available if a service has a subfunction.

7.4.4.3 Verification functionality

Prior of execution of a received diagnostic service, the DSD performs a set of verifications. The DSD will only accept a service, if all verifications are successfully passed.

[SWS_Dcm_01535] DSD verifications prior of service execution [The `Dcm` shall only accept a diagnostic request, if the following verifications have been passed in the following order:] ([RS_Diag_04230](#), [RS_Diag_04005](#), [RS_Diag_04006](#))

1. Verification of Manufacturer permission (Call of the manufacturer interface indication operation)
2. Verification of the SID
3. Verification of the service access control on the current authentication state
4. Verification of the Diagnostic Session
5. Verification of the Service Security Access levels
6. Verification of the Supplier permission (Call of the Supplier interface indication operation)
7. Verification of the Mode rules for service IDs.

[SWS_Dcm_01474] [In case the DSD generates a `NRC`, the `Dcm` shall only call `XXX_Confirmation`.] ([RS_Diag_04235](#))

This means that the `Dcm` will not call `DsplInternal_DcmConfirmation`().

7.4.4.3.1 Verification of the diagnostic service access rights

The UDS service Authentication (0x29) is used to change the authentication state of a diagnostic connection and to provide the access rights. Depending on the reached role

and provided white list a dynamic set of diagnostic services is available for the tester on that connection. The DSD submodule verifies on service ID (SID) and sub-function (SF) level, if a service can be executed or not.

[SWS_Dcm_01536] Authentication on UDS services only [The Dcm shall only verify the authentication for UDS services. A UDS service has a service ID within the range of 0x10 and 0xFF.] ([RS_Diag_04230](#))

OBD services are explicitly excluded from authentication checks. By legislation the OBD services need to be always available, independent from active authentication state. If WWH-OBD is used the system engineer must ensure that these services are always accessible.

[SWS_Dcm_01537] Verifying access rights [The Dcm shall only verify and check the configured access rights of a diagnostic service, if the container `DcmDspAuthentication` is configured.] ([RS_Diag_04230](#))

If no `DcmDspAuthentication` is configured, the Dcm will process all diagnostic services as if the current connection would grant access to execute the current processed service. Checking the access rights for diagnostic services is done at different levels of the service structure. The use of diagnostic service access rights introduces means to allow or to refuse a diagnostic service due to current roles and authentication states. Some services shall always be allowed to be executed, like the service 0x29 (Authentication) to set the current tester access rights. This service and other OEM or supplier specific services should have granted access independent from the authentication state. To realize this, the Dcm uses a default role that is used in all deauthenticated states. In that state, all role based verifications are done as in authenticated state. The active role is provided by the configuration.

[SWS_Dcm_01538] Access rights for services in deauthenticated state [If the current connection is in deauthenticated state, the Dcm shall use the role configured in `DcmDspAuthenticationDeauthenticatedRole` as current role for all role based access verification checks.] ([RS_Diag_04230](#))

[SWS_Dcm_01539] Definition of allowed service execution [The Dcm shall allow the service execution, if a role verification was successful or the service is allowed by the white list.] ([RS_Diag_04233](#))

[SWS_Dcm_01540] Diagnostic service execution rights verification [The Dcm shall check if a service execution is permitted in the current authentication check or not. The Dcm shall perform the following checks in the given order below. If a check grants access to a service, the remaining checks are skipped:

1. Checks on service ID level
2. Checks on service ID and sub-function level
3. Checks for services with one or multiple DIDs
4. Check on dynamically defined DIDs
5. Checks on service 0x31 per sub-function

6. Checks on service 0x19 parameter MemorySelection

](RS_Diag_04233)

[SWS_Dcm_01541] Service ID authentication check for UDS service requests

[Upon processing a diagnostic service, the Dcm shall grant access to the diagnostic service if:

1. a DcmDsdServiceRole is configured for that service and the verification according to [SWS_Dcm_01522] was successful or
2. the active white list on that connection has one entry for a SID (1-byte element) which matches that service.

](RS_Diag_04233)

[SWS_Dcm_01542] Service with sub-function authentication check for UDS service requests

[Upon processing a diagnostic service with sub-function, the Dcm shall grant access to the diagnostic service if:

1. a DcmDsdSubServiceRole is configured for that service and sub-function and the verification according to [SWS_Dcm_01522] was successful or
2. the active white list on that connection has one entry for a SID with sub-function (2-byte element) that matches that service and sub-function.

](RS_Diag_04230)

[SWS_Dcm_01562] White list verification for services with 3 and 4 bytes [For 3 and 4 bytes white list for services entries, the Dcm shall verify on the full length of the configured white list service element. The service is granted access if the first bytes of the received request match the entire white list entry.] (RS_Diag_04233)

Verification of byte 3 and 4 within the Dsd is beyond the scope of a typical Dsd operation. It provides means to extend the capabilities of white list service verifications and gives means to adapt to legacy authentication solutions.

[SWS_Dcm_01544] Response behavior of services without access rights [If the service execution verification fails due to a failed check in scope of [SWS_Dcm_01483], the Dcm shall send a NRC 0x34 authenticationRequired and stop the service processing.] (RS_Diag_04230)

7.4.4.3.2 Verification of the Diagnostic Session

The UDS Service DiagnosticSessionControl (0x10) is used to enable different diagnostic sessions in the ECU (e.g. Default session, Extended session). A diagnostic session enables a specific set of diagnostic services and/or functionality in the ECU. It furthermore enables a protocol-depending data set of timing parameters applicable to the started session.

On receiving a service request, the **DSD** module will obtain the current Diagnostic Session with `Dcm_GetSesCtrlType` and will verify whether the execution of the requested service (NOT the **UDS** Service DiagnosticSessionControl (0x10)) and sub-service is allowed in the current diagnostic session or not.

Note that the handling of the **UDS** Service DiagnosticSessionControl (0x10) itself is not part of the **DSD** submodule.

[SWS_Dcm_00211] [If the newly received diagnostic service is not allowed in the current Diagnostic Session (according to the configuration parameter `DcmDsdSidTabSessionLevelRef`), the **DSD** submodule shall transmit a negative response with **NRC** 0x7F (serviceNotSupportedInActiveSession) to the **DSL** submodule.]()

[SWS_Dcm_00616] [If the newly received diagnostic service is allowed in the current Diagnostic Session (see **[SWS_Dcm_00211]**), but the requested subservice is not allowed in the current Diagnostic Session (according to the configuration parameter `DcmDsdSubServiceSessionLevelRef`), the **DSD** submodule shall transmit a negative response with **NRC** 0x7E (subFunctionNotSupportedInActiveSession) to the **DSL** submodule.]()

7.4.4.3.3 Verification of the Service Security Access levels

The purpose of the Security Access level handling is to provide a possibility to access data and/or diagnostic services, which have restricted access for security, emissions, or safety reasons. The **DSD** submodule shall perform this handling with the **UDS** Service SecurityAccess (0x27). The **DSD** submodule will perform a verification whether the execution of the requested service (NOT the **UDS** Service SecurityAccess (0x27)) is allowed in the current Security level by asking for the current security level, using the **DSL** function `Dcm_GetSecurityLevel`.

The management of the security level is not part of the **DSD** submodule.

Note: For some use cases (e.g. **UDS** Service ReadDataByIdentifier (0x22), where some DataIdentifier can be secure) it will be necessary for the Application to call also the function `Dcm_GetSecurityLevel`.

[SWS_Dcm_00217] [If the newly received diagnostic service is not allowed in the current Security level (according to the configuration parameter `DcmDsdSidTabSecurityLevelRef`), the **DSD** submodule shall transmit a negative response with **NRC** 0x33 (Security access denied) to the **DSL** submodule.]()

[SWS_Dcm_00617] [If the newly received diagnostic service is allowed in the current Security level (see **[SWS_Dcm_00217]**), but the requested subservice is not allowed in the current Security level (according to the configuration parameter `DcmDsdSubServiceSecurityLevelRef`), the **DSD** submodule shall transmit a negative response with **NRC** 0x33 (Security access denied) to the **DSL** submodule.]()

7.4.4.3.4 Verification of the Service mode dependencies

[SWS_Dcm_00773] [If the newly received diagnostic service is not allowed in the current mode condition (according to the configuration parameter `DcmDsdSidTabModeRuleRef`), the `DSD` submodule shall transmit the calculated negative response of the referenced `DcmModeRule` to the `DSL` submodule.]()

[SWS_Dcm_00774] [If the newly received diagnostic service is allowed in the current mode condition [**[SWS_Dcm_00773]**], but the requested subservice is not allowed in the current mode condition (according to the configuration parameter `DcmDsdSubServiceModeRuleRef`), the `DSD` submodule shall transmit the calculated negative response of the referenced `DcmModeRule` to the `DSL` submodule.]()

7.4.4.4 Check format and subfunction support

The `DSD` submodule checks whether a specific subfunction is supported before executing the requested command.

[SWS_Dcm_00273] General sub-function supported NRC check [The `DSD` shall send the negative response NRC 0x12 (`sub-functionNotSupported`), if for the processed service no configured `DcmDsdSubService` exists with the `DcmDsdSubServiceId` of the processed service. This NRC check shall not be done for UDS Service 0x31 (`RoutineControl`).]()

The `DSD` submodule will check for the minimum message length before executing the requested command.

[SWS_Dcm_00696] [The `DSD` submodule shall trigger a negative response with NRC 0x13 (`Incorrect message length or invalid format`), if the length of the request is inferior to the minimum length of the request.]()

[SWS_Dcm_01411] [If `DcmDsdSubService` is configured for a `DcmDsdService`, the `Dcm` shall support the sub-function configured in `DcmDsdSubServiceId` with `SPRMIB` set to 0 or 1.]()

7.4.4.4.1 Verification of the Manufacturer Application environment/permission

The purpose of this functionality is that, just after receiving the diagnostic request, the Manufacturer Application is requested to check permission/environment.

E.g. in after-run ECU state, it might be not allowed to process `OBD` requests.

[SWS_Dcm_00218] [If container `DcmDsdServiceRequestManufacturerNotification` exists, the `DSD` submodule shall call the operation `Xxx_Indication` on all configured `ServiceRequestIndication` ports (see configuration parameter `DcmDsdServiceRequestManufacturerNotification`).]()

[SWS_Dcm_00462] [If at least a single `Xxx_Indication` function called according to [\[SWS_Dcm_00218\]](#) returns `E_REQUEST_NOT_ACCEPTED`, the `DSD` submodule shall give no response.]()

[SWS_Dcm_01172] [In case of [\[SWS_Dcm_00462\]](#), the `DSD` shall only call `Xxx_Confirmation` but not `DsplInternal_DcmConfirmation`.]()

[SWS_Dcm_00463] [If at least a single `Xxx_Indication` function called according to [\[SWS_Dcm_00218\]](#) has returned `E_NOT_OK` and no function has returned `E_REQUEST_NOT_ACCEPTED`, the `DSD` submodule shall trigger a negative response with `NRC` from the `ErrorCode` parameter.]()

[SWS_Dcm_01321] [If more than one `Xxx_Indication` function called, according to [\[SWS_Dcm_00218\]](#), has returned `E_NOT_OK` and no function has returned `E_REQUEST_NOT_ACCEPTED`, the `DSD` submodule shall trigger a negative response using the `ErrorCode` parameter from the first `Xxx_Indication` returning `E_NOT_OK`.]([RS_Diag_04011](#))

7.4.4.4.2 Verification of the Supplier Application environment/permission

The purpose of this functionality is that, right before processing the diagnostic message, the Supplier Application is requested to check permission/environment.

E.g. in after-run ECU state, it might be not allowed to process `OBD` requests.

[SWS_Dcm_00516] [If container `DcmDsdServiceRequestSupplierNotification` exists, the `DSD` submodule shall call the operation `Xxx_Indication` on all configured `ServiceRequestIndication` ports (see configuration parameter `DcmDsdServiceRequestSupplierNotification`).]()

[SWS_Dcm_00517] [If at least a single `Xxx_Indication` function called according to [\[SWS_Dcm_00516\]](#) returns `E_REQUEST_NOT_ACCEPTED`, the `DSD` submodule shall give no response.]()

[SWS_Dcm_00518] [If at least a single `Xxx_Indication` function called according to [\[SWS_Dcm_00516\]](#) has returned `E_NOT_OK` and no function has returned `E_REQUEST_NOT_ACCEPTED`, the `DSD` submodule shall trigger a negative response with `NRC` from the `ErrorCode` parameter.]()

[SWS_Dcm_01322] [If more than one `Xxx_Indication` function called, according to [\[SWS_Dcm_00516\]](#), has returned `E_NOT_OK` and no function has returned `E_REQUEST_NOT_ACCEPTED`, the `DSD` submodule shall trigger a negative response using the `ErrorCode` parameter from the first `Xxx_Indication` returning `E_NOT_OK`.]([RS_Diag_04011](#))

7.4.4.5 Distribution of diagnostic message to DSP submodule

[SWS_Dcm_00221] [The DSD submodule shall search for the executable functionality of the DSP submodule for newly received diagnostic service identifier and shall call the corresponding DSP service interpreter.]()

7.4.4.6 Assemble positive or negative response

[SWS_Dcm_00222] [When the DSP submodule has finished the execution of the requested Diagnostic Service the DSD submodule shall assemble the response.]()

The execution of the DSP service interpreter can have the results:

- positive Result or
- negative Result.

Following possible Responses can be assembled:

- positive Response,
- negative Response, or
- no Response (in the case of suppression of responses).

7.4.4.6.1 Positive Response

[SWS_Dcm_00223] [The DSD submodule shall add the response service identifier and the response data stream (returned by the Application) in the parameter "Dcm_MsgContextType".]()

[SWS_Dcm_00224] [The DSD submodule shall therefore transfer the Dcm_MsgContextType into a (response) buffer and shall add the service identifier at the first byte of the buffer.]()

[SWS_Dcm_00225] [The DSD submodule shall execute the "Initiate transmission" functionality in the next execution step.]()

7.4.4.6.2 Negative Response

The DSP submodule can trigger the transmission of a negative response with a specific NRC to the DSD submodule. For the allowed NRC of the executed Service ID please refer to the specification of the service in ISO14229-1 [1] (see Section 4.2.4 Response code parameter definition Table 12) and ISO15031-5 [2]. The DSP and the Application have to take care of the correct use of NRC of the executed Service ID.

[SWS_Dcm_00228] [The [DSD](#) submodule shall handle all NRCs supported from the Application and defined in `Dcm_NegativeResponseCodeType`.]()

7.4.4.6.3 Suppression of response

[SWS_Dcm_00231] [In the case that the "suppressPosRspMsgIndicationBit" is indicated in the functionality "Handling of suppressPosRspMsgIndicationBit" (stored in the Variable `Dcm_MsgContextType` (Element: `Dcm_MsgAddInfo`)), the [DSD](#) submodule shall activate the suppression of Positive Responses.]()

[SWS_Dcm_00001] [In the case of a Negative Result of the execution and active [Functional Addressing](#) the [DSD](#) submodule shall activate the suppression of the following Negative Responses:

- [NRC 0x11](#) (Service not supported),
- [NRC 0x12](#) (SubFunction not supported),
- [NRC 0x31](#) (Request out of range),
- [NRC 0x7E](#) (Subfunction not supported in active session),
- [NRC 0x7F](#) (Service not supported in active session)

] ([RS_Diag_04020](#))

7.4.4.7 Initiate transmission

[SWS_Dcm_00232] [The [DSD](#) submodule shall forward the diagnostic (response) message (positive or negative response) to the [DSL](#) submodule.]()

[SWS_Dcm_00237] [The [DSL](#) submodule shall forward the diagnostic (response) message (positive or negative response) further to the PduR module by executing a [DSL](#) transmit functionality.]()

The [DSL](#) submodule will receive a confirmation by the PduR module upon forwarding the data.

[SWS_Dcm_00235] [The [DSL](#) submodule shall forward the received confirmation from the PduR module to the [DSD](#) submodule.]()

[SWS_Dcm_00236] [The [DSD](#) submodule shall forward the confirmation via the internal function `DspInternal_DcmConfirmation()` to the [DSP](#) submodule.]()

[SWS_Dcm_00238] [In the case that no diagnostic (response) message shall be sent (Suppression of Responses) the [DSL](#) submodule shall not transmit any response.]()

In this case no Data Confirmation is sent from the [DSL](#) submodule to the [DSD](#) submodule but the [DSD](#) submodule will still call internal function `DspInternal_DcmConfirmation()`.

[SWS_Dcm_00240] [In case the request has been fully processed by the Dcm, The [DSD](#) submodule shall finish the processing of one Diagnostic Message of the Diagnostic Service Dispatcher by calling `DsplInternal_DcmConfirmation()`.]()

Rationale for [\[SWS_Dcm_00240\]](#): The [DSP](#) submodule is waiting for the execution of the `DsplInternal_DcmConfirmation()` functionality. So it has to be sent, also when no Data Confirmation is provided. Altogether this means that in any of the following cases:

- Positive Response,
- Negative Response,
- Suppressed Positive Response, and
- Suppressed Negative Response

The [DSD](#) submodule will finish by calling `DsplInternal_DcmConfirmation()` (refer to [8.10.3 DsplInternal_DcmConfirmation](#)).

[SWS_Dcm_00741] [The [DSD](#) submodule shall call the operation `Xxx_Confirmation()` on all ports using the `ServiceRequestNotification` interface (see configuration parameter `DcmDsdServiceRequestManufacturerNotification` and `DcmDsdServiceRequestSupplierNotification`)]()

[SWS_Dcm_00742] [The call of `Xxx_Confirmation()` shall be done right after the call of `DsplInternal_DcmConfirmation()`.]()

[SWS_Dcm_00677] [If the operation `Indication()` returns value `E_REQUEST_NOT_ACCEPTED`, the [Dcm](#) module shall not send any diagnostic response and shall end the current diagnostic request management.]()

[SWS_Dcm_00678] [If the operation `Indication()` returns value `E_NOT_OK`, the [Dcm](#) module shall send a negative response with `NRC` value equal to `ErrorCode` parameter value.]()

7.5 Diagnostic Service Processing (DSP)

7.5.1 General

When receiving a function call from the [DSD](#) submodule requiring the [DSP](#) submodule to process a diagnostic service request, the [DSP](#) always carries out following basic process steps:

- analyze the received request message,
- check format and whether the addressed subfunction is supported,
- acquire data or execute the required function call on the DEM, SW-Cs or other BSW modules
- assemble the response

The following sections are some general clarifications.

7.5.1.1 Check format and subfunction support

The `DSP` submodule will check for appropriate message length and structure before executing the requested command.

[SWS_Dcm_00272] [The `DSP` submodule shall trigger a negative response with `NRC` 0x13 (Incorrect message length or invalid format), when the analysis of the request message results in formatting or length failure.] ()

Note: It is up to the implementation in which detail the format check might be executed and depends on the level of detail the diagnostic data description provides at compile time.

7.5.1.2 Assemble response

[SWS_Dcm_00039] [The `DSP` submodule shall assemble the response message excluding response service identifier and determine the response message length.] ()

[SWS_Dcm_00038] [If the paged-buffer mechanism is used, the `DSP` submodule shall determine the overall response length before any data is passed to the `DSD` submodule or the `DSL` submodule respectively.] ()

Requirement **[SWS_Dcm_00038]** is needed because of segmented diagnostic data transmission on `CAN` using ISO15765-2 [11], which requires the provision of the overall length of the complete data stream in the very first `CAN` frame of the respective data transmission (please refer to Section 7.3.4.9 for details about the paged-buffer mechanism).

7.5.1.3 Negative Response Codes handling

[SWS_Dcm_00271] [Unless another particular `NRC` is specified, the `DSP` submodule shall trigger a negative response with `NRC` 0x10 (generalReject), when the `API` calls made to execute the service do not return OK.] ()

[SWS_Dcm_01414] Accepted range of Dcm_NegativeResponseType for negative responses [If the `Dcm` calls an external application by any of the `APIs` having the out parameter `Dcm_NegativeResponseType` `ErrorCode`, the `Dcm` shall accept only values in the range 0x01-0xFF in case the return value is `E_NOT_OK`.] ()

[SWS_Dcm_01415] Behavior on application returning unexpected return code [If the `Dcm` calls an `API` with the out parameter `Dcm_NegativeResponseType` `ErrorCode` and the application sets this parameter to `DCM_POS_RESP` and `E_NOT_OK` is returned, the `Dcm` shall report the runtime error `DCM_E_INVALID_VALUE`.] ()

[SWS_Dcm_00275] [The `DSP` submodule shall trigger a negative response with `NRC 0x31` (Request out of range), when the analysis of the request message results in other unsupported message parameters.]()

7.5.1.4 Diagnostic mode declaration groups

[SWS_Dcm_00775] [The `Dcm` shall act as a mode manager for the diagnostic modes:

1. `DcmDiagnosticSessionControl` (service `0x10`)
2. `DcmEcuReset` (partly service `0x11`)
3. `DcmSecurityAccess` (service `0x27`)
4. `DcmModeRapidPowerShutDown` (partly service `0x11`)
5. `DcmCommunicationControl_<symbolic name of ComMChannelId>`. (service `0x28`)
6. `DcmControlDTCSetting` (service `0x85`)
7. `DcmResponseOnEvent_<RoeEventID>` (service `0x86`)
8. `DcmAuthenticationState_<Symbolic Name of DcmDsIMainConnection>`

]()

Note: The RTE/SchM will prefix the names with "MODE_", wherefore the names do not include the MODE keyword.

[SWS_Dcm_01327] [The `Dcm` shall define the `ModeDeclarationGroupPrototype DcmSecurityAccess` as provided-ModeGroup based on the following `ModeDeclarationGroup`:

```

1 ModeDeclarationGroup DcmSecurityAccess {
2     {
3         DCM_SEC_LEV_LOCKED
4         DCM_SEC_LEV_1
5         ...
6         DCM_SEC_LEV_63
7     }
8     initialMode = DCM_SEC_LEV_LOCKED
9 };

```

]()

[SWS_Dcm_01328] [

```

1 ModeSwitchInterface SchM_Switch_<bsnp>_DcmSecurityAccess {
2     isService = true;
3     SecLevel currentMode;
4 };

```

]()

[SWS_Dcm_00806] [The `Dcm` shall define the `ModeDeclarationGroupPrototype DcmDiagnosticSessionControl` as provided-ModeGroup based on the `ModeDeclarationGroup DcmDiagnosticSessionControl`.]()

[SWS_Dcm_00777] [The `Dcm` shall define the `ModeDeclarationGroupPrototype DcmEcuReset` as provided-ModeGroup in its Basic Software Module instance based on the `ModeDeclarationGroup DcmEcuReset`.]()

[SWS_Dcm_00807] [The `Dcm` shall define the `ModeDeclarationGroupPrototype DcmModeRapidPowerShutDown` as provided-ModeGroup in its Basic Software Module instance based on the `ModeDeclarationGroup DcmModeRapidPowerShutDown`.]()

[SWS_Dcm_00780] [The `Dcm` shall define for each network which is considered in the `CommunicationControl` service a separate `ModeDeclarationGroupPrototype DcmCommunicationControl_<symbolic name of ComMChannelId>` as provided-ModeGroup in its Basic Software Module instance based on the `ModeDeclarationGroup DcmCommunicationControl`.]()

[SWS_Dcm_00781] [The `Dcm` shall define the `ModeDeclarationGroupPrototype DcmControlDTCSetting` as provided-ModeGroup in its Basic Software Module instance based on the `ModeDeclarationGroup DcmControlDTCSetting`.]()

[SWS_Dcm_00933] [The `Dcm` shall define for each `RoeEvent` a separate `ModeDeclarationGroupPrototype DcmResponseOnEvent_<Symbolic name of RoeEventId>` as provided-ModeGroup in its Basic Software Module instance based on the `ModeDeclarationGroup DcmResponseOnEvent`.]()

The `Dcm` provides a state machine for each `RoeEvent` (see Figure 7.4). The state for a `RoeEvent` is needed by SWC to activate event reporting or report the `Roe` status to a `Did`. Therefore the `Dcm` provides for each state of each `RoeEvent` a `ModeDeclarationGroupPrototype` which reports the current state of the state machine as `mode`.

[SWS_Dcm_00934] [The `ModeDeclarationGroupPrototype` shall represent the current state of the ROE state machine for this `RoeEvent`.]()

7.5.1.5 Environmental condition dependent execution

The execution of a diagnostic service or the acceptance of certificates can be restricted to a mode condition. This enables the `Dcm` to formalize environmental checks. For diagnostic service processing, a further check (see [SWS_Dcm_00773] and [SWS_Dcm_00774]) can be configured to the `Dcm`. This is like session and security checks. The referenced mode rule is arbitrating on to several mode declarations of a mode declaration groups in which the request can be processed. Otherwise a configurable NRC (see [SWS_Dcm_00812]) is responded. The same mode rule checks can be applied on certificate validation. Certificates can be restricted to certain vehicle properties, such as VIN or a certain version number. Only if all the conditions are valid, the certificate is accepted by the `Dcm`.

[SWS_Dcm_00808] [The `DcmModeRule` shall evaluate all referenced `DcmModeConditions` and/or nested `DcmModeRules` either by a logical AND in case `DcmLogicalOperator` is set to `DCM_AND` or by a logical OR in case the `DcmLogicalOperator` is set to `DCM_OR`. In case only a single `DcmModeCondition` or `DcmModeRule` is referenced the `DcmLogicalOperator` shall not be present and therefore not be used.]
()

[SWS_Dcm_CONSTR_6028] [`DcmModeCondition` shall either have a `DcmBswModeRef` or a `DcmSwcModeRef` or a `DcmSwcSRDataElementRef` as external reference.]
()

[SWS_Dcm_00810] [The `DcmSwcModeRef` and `DcmBswModeRef` of `DcmModeConditions` shall evaluate if the referenced Mode-Declaration is set in case of `DcmConditionType` is set to `DCM_EQUALS` or is not set in case of `DcmConditionType` is set to `DCM_EQUALS_NOT`.]
()

[SWS_Dcm_01119] Mode condition evaluation [For each mode condition, the Dcm shall compare a compare value with a S/R data element. The compare value is provided by `DcmSwcSRDataElementValueRef` or `DcmModeConditionConnectionCertificateCompareElementRef` and the S/R Element is by `DcmSwcSRDataElementRef`. The mode condition is evaluated to true if the S/R data element value is:

- equal to the compare value in case of `DcmConditionType` is set to `DCM_EQUALS`
- unequal to the compare value in case of `DcmConditionType` is set to `DCM_EQUALS_NOT`
- greater than the compare value in case of `DcmConditionType` is set to `DCM_GREATER_THAN`
- greater or equal than the compare value in case of `DcmConditionType` is set to `DCM_GREATER_OR_EQUAL`
- less than the compare value in case of `DcmConditionType` is set to `DCM_LESS_THAN`
- less or equal than the compare value in case of `DcmConditionType` is set to `DCM_LESS_OR_EQUAL`.

]()

[SWS_Dcm_CONSTR_6029] [The values `DCM_GREATER_THAN`, `DCM_GREATER_OR_EQUAL`, `DCM_LESS_OR_EQUAL` and `DCM_LESS_THAN` shall not used with a Mode reference (`DcmBswModeRef` or `DcmSwcModeRef`).]
()

Note: The current mode of the referenced `ModeDeclarationGroupPrototypes` could be read by either the `API SchM_Mode` (in case of `DcmBswModeRef`) or by the `API Rte_Mode` (in case of `DcmSwcModeRef`).

[SWS_Dcm_00811] [In case multiple `DcmModeConditions` are referenced within a `DcmModeRule` they shall be evaluated in order of the index attributes of the `EcucReferenceValues` for `DcmArgumentRef`.]()

Note: This implies the priority of NRCs

[SWS_Dcm_00782] [If a `DcmModeRule` is not referenced from the `DcmDspAuthenticationConnection`, the Dcm shall use the optional parameter `DcmModeRuleNrcValue` as `NegativeResponseCode` in case the mode rule is evaluated to false.]()

Mode rules for `DcmDspAuthenticationConnection` are not part of the NRC evaluation.

[SWS_Dcm_00812] [In case a nested `DcmModeRule` contains also a `DcmModeRuleNrcValue` parameter, this NRC shall be used prior the higher-level NRC.]()

[SWS_Dcm_00813] [In case `DcmLogicalOperator` is set to `DCM_AND`, the first failed `DcmModeRule` with an explicit configured NRC (`DcmModeRuleNrcValue`) shall be used to define the NRC for the response message.]()

[SWS_Dcm_00814] [In case `DcmLogicalOperator` is set to `DCM_OR`, the last failed `DcmModeRule` with an explicit configured NRC (`DcmModeRuleNrcValue`) shall be used to define the NRC for the response message.]()

Note: The difference in the AND and OR logical operation is to allow an optimized implementation.

[SWS_Dcm_00815] [In case the complete evaluation result in no specific NRC the NRC 0x22 (`ConditionsNotCorrect`) shall be used.]()

[SWS_Dcm_00942] [The Dcm shall create for commonly used `ModeDeclarationGroupPrototype` of each `DcmSwcModeRef` of `DcmModeConditions` a required mode switch port referencing this `ModeDeclarationGroupPrototype`. The name pattern of this port prototype shall be `DcmModeUser_<ModeDeclarationGroupPrototype>` in case the `ModeDeclarationGroupPrototype` shortname is unique. Otherwise the name pattern is implementation specific, except the required prefix "DcmModeUser_".]()

Note: `ModeDeclarationGroupPrototypes` are not necessarily unique, wherefore the exception is required to avoid name clashes in the Dcm Service-SWC.

Examples on using mode dependent request execution:

General assumptions:

1. `DcmModeRule1` consists of `DcmModeCondition1`, `DcmModeRule2` and `DcmModeRule3`
2. `DcmModeRule1` defines NRC 0x22
3. `DcmModeRule2` and `DcmModeRule3` do not have any sub-rules
4. `DcmModeRule2` defines NRC 0x72
5. `DcmModeRule3` does not define a NRC value

Example 1:

- 1) DcmModeRule1 uses an OR combination (DcmModeCondition1 OR DcmModeRule2 OR DcmModeRule3)
 - a) DcmModeCondition1 is failing
→ NRC 0x22 is returned
 - b) DcmModeRule2 is failing
→ NRC 0x72 is returned
 - c) DcmModeRule3 is failing
→ NRC 0x22 is returned
 - d) DcmModeCondition1, DcmModeRule2 and DcmModeRule3 are failing
→ NRC 0x72 is returned
 - e) DcmModeCondition1 and DcmModeRule3 are failing
→ NRC 0x22 is returned

Example 2:

- 1) DcmModeRule1 uses an AND combination (DcmModeCondition1 AND DcmModeRule2 AND DcmModeRule3)
 - a) DcmModeCondition1 is failing
→ NRC 0x22 is returned
 - b) DcmModeRule2 is failing
→ NRC 0x72 is returned
 - c) DcmModeRule3 is failing
→ NRC 0x22 is returned
 - d) DcmModeCondition1, DcmModeRule2 and DcmModeRule3 are failing
→ NRC 0x22 is returned
 - e) DcmModeCondition1 and DcmModeRule3 are failing
→ NRC 0x22 is returned
 - e) DcmModeRule2 and DcmModeRule3 are failing
→ NRC 0x72 is returned

[SWS_Dcm_CONSTR_6089] Only one compare element [In one [DcmModeCondition](#) only one of the elements [DcmSwcSRDataElementRef](#) or [DcmModeConditionCertificateCompareElementRef](#) shall be configured.] ([RS_Diag_04232](#))

[SWS_Dcm_CONSTR_6090] Use of certificate compare elements [The [DcmModeConditionCertificateCompareElementRef](#) is only allowed, if the parent [DcmModeRule](#) is referenced from a [DcmDspAuthenticationConnection](#).] ()

7.5.1.6 Sender/Receiver Communication

[SWS_Dcm_00964] [If [DcmDspDiagnosisScaling](#) is present, the [Dcm](#) shall derive the [CompuMethod](#) from the [DcmDspDiagnosisScaling](#) container and add it to the [DataType](#) in their respective port interface for S/R port of [DataServices_Data](#) [\[SWS_Dcm_01035\]](#).] ()

7.5.1.7 Passing SwDataDefProps properties from DEXT file to the Dcm Service SW-C

UseCase: Pass the SwDataDefProps details like CompuMethod, DataContraints and Units to the Dcm Service SW-C and make them there available per DID DataElement / per RoutineControl signal. Two alternative work flows are available.

7.5.1.7.1 DcmDspDiagnosticDataElementRef workflow

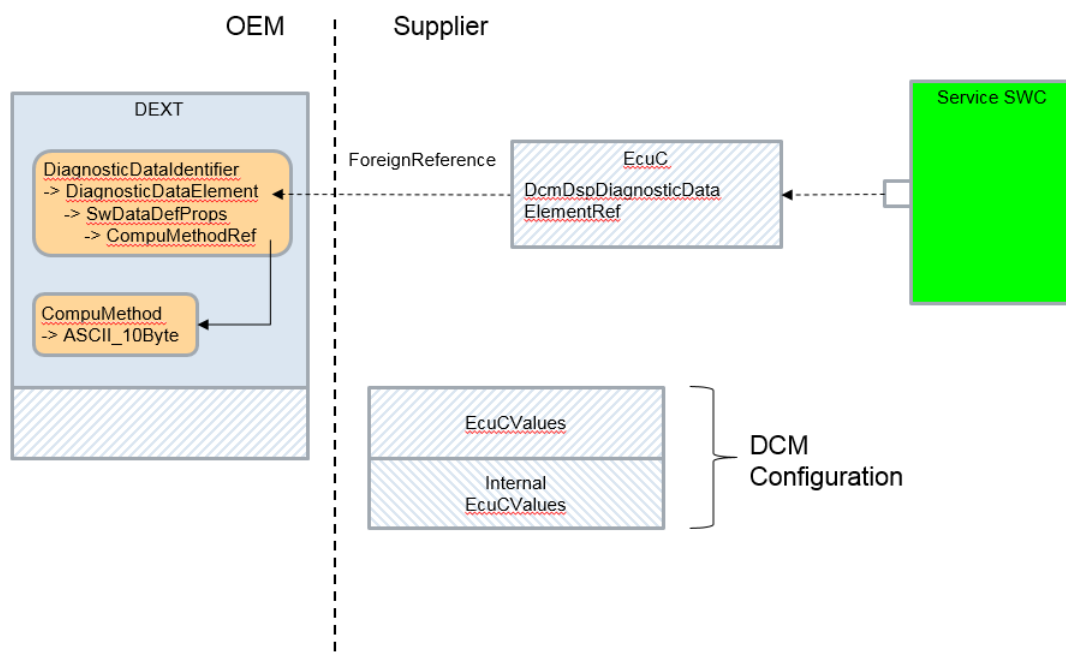


Figure 7.6: DcmDspDiagnosticDataElementRef Workflow

The feature of the `DcmDspDiagnosticDataElementRef` workflow is the use of a `EcucForeignReference` inside the generated `EcuC` values. While importing the DEXT information, a dedicated `EcuC` parameter is generated, which holds a `EcucForeignReference` named `DcmDspDiagnosticDataElementRef` to a `DiagnosticDataElement` in the DEXT file. This `EcucForeignReference` enables the access to all `SwDataDefProps` (`BaseType`, `CompuMethod`, `DataConstr`, etc.) of the corresponding `DiagnosticDataElement`. The container `DcmDspAlternativeDiagnosticDataElement` aggregates this `EcucForeignReference`. In the process step of generating the corresponding `Service SWC` all needed content will be copied directly based on the `EcucForeignReference` from DEXT to the `Service SW-C`. In this work flow the existence of the DEXT file while the generation of the `Service SW-C` is required.

[SWS_Dcm_CONSTR_6053] [The aggregation of `DcmDspTextTableMapping` at `DcmDspAlternativeDataType` is only valid if the category of the `CompuMethod`

of the DataType referenced by `DcmDspAlternativeDataType.DcmApplicationDataType` has category set to `TEXTTABLE` or `SCALE_LINEAR_AND_TEXTTABLE`.]()

7.5.1.7.2 `DcmDspAlternativeDataType.DcmApplicationDataType` workflow

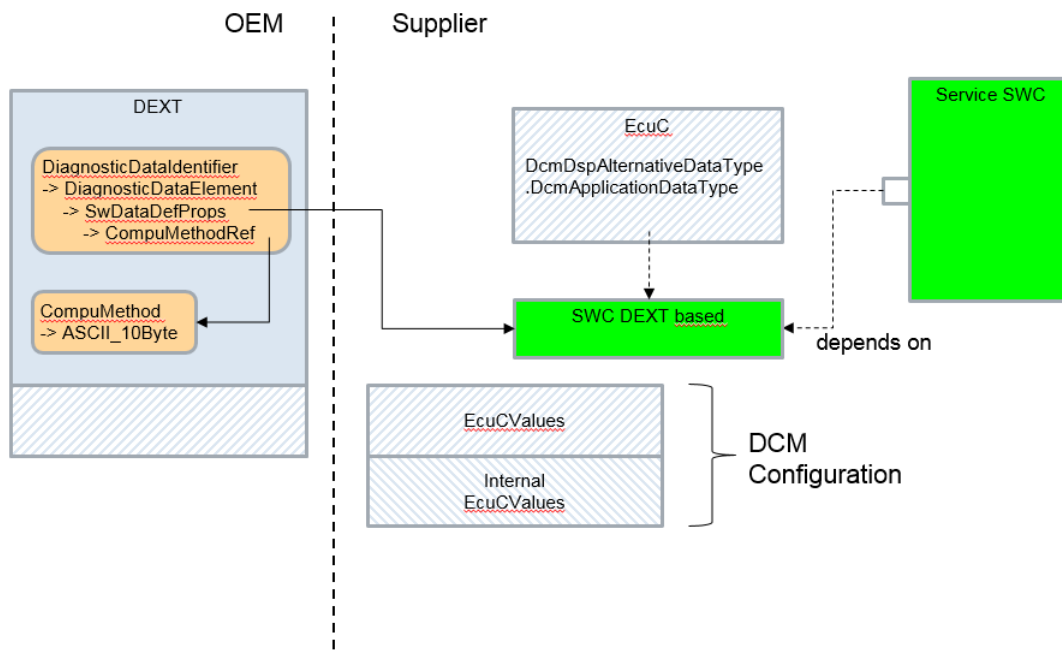


Figure 7.7: `DcmDspAlternativeDataType.DcmApplicationDataType` Workflow

The feature of the `DcmDspAlternativeDataType.DcmApplicationDataType` workflow is that while importing the DEXT information beside the EcuC values also a `SW-C` fragment is generated. In this `SW-C` fragment all needed `SwDataDefProps` are directly copied from the DEXT file. Inside the generated EcuC values the EcuC parameter `DcmDspAlternativeDataType.DcmApplicationDataType` refers to the SWC fragment and enables the access to all `SwDataDefProps` (`BaseType`, `CompuMethod`, `DataConstr`, etc.). In the process step of generating the corresponding Service `SW-C`, all needed content will be included based on the reference from `DcmDspAlternativeDataType.DcmApplicationDataType` to the `SW-C` fragment. In this work flow the existence of the DEXT file while the generation of the Service `SW-C` is not required.

7.5.1.8 Asynchronous call behavior

[SWS_Dcm_01412] [If a `Dem` function returns `DEM_PENDING`, the `Dcm` shall call this function again at a later point in time as long as `DEM_PENDING` is returned.]()

[SWS_Dcm_00120] [If the number of negative responses for a requested diagnostic tasks (see [SWS_Dcm_00024]) reaches the value defined in the configuration parameter `DcmDslDiagRespMaxNumRespPend`, the `Dcm` module shall stop processing the active diagnostic request, inform the application or BSW (if this diagnostic task implies the call to a `SW-C` interface or a BSW interface) by setting `OpStatus` parameter, of active port interface, to `DCM_CANCEL`, report the runtime error `DCM_E_INTERFACE_TIMEOUT` and shall send a negative response with `NRC 0x10` (General reject).] ()

[SWS_Dcm_01184] [The `Dcm_SetProgConditions` API shall be called again in the next `Dcm` main function cycle if previous return status was `E_PENDING`.] ()

[SWS_Dcm_00760] [The return of `DCM_E_PENDING` shall do a re-triggering (e.g. in the next `MainFunction` cycle).] ()

[SWS_Dcm_01413] [The return values of interfaces called with an `OpStatus` equal to `DCM_CANCEL` shall be ignored] ()

7.5.2 UDS Services

[SWS_Dcm_00442] [The `Dcm` module shall implement the services of `UDS` according to Table 7.7.] ()

SID	Service	Subfunction	Supported
0x10	DiagnosticSessionControl		Supported
0x11	ECUReset		Supported
0x14	ClearDiagnosticInformation		Supported
0x19	ReadDTCInformation		Supported
0x22	ReadDataByIdentifier		Supported
0x23	ReadMemoryByAddress		Supported (callout)
0x24	ReadScalingDataByIdentifier		Supported
0x27	SecurityAccess		Supported
0x28	CommunicationControl		Supported
0x29	Authentication		Supported
0x2A	ReadDataByPeriodicIdentifier		Supported
0x2C	DynamicallyDefineDataIdentifier		Supported
0x2E	WriteDataByIdentifier		Supported
0x2F	InputOutputControlByIdentifier		Supported
0x31	RoutineControl		Supported
0x34	RequestDownload		Supported (callout)
0x35	RequestUpload		Supported (callout)
0x36	TransferData		Supported
0x37	RequestTransferExit		Supported
0x38	RequestFileTransfer		Supported (callout)
0x3D	WriteMemoryByAddress		Supported (callout)
0x3E	TesterPresent		Supported
0x83	AccessTimingParameter		<code>NRC</code> "ServiceNotSupported"
0x84	SecuredDataTransmission		<code>NRC</code> "ServiceNotSupported"
0x85	ControlDTCSetting	On, off	Supported

SID	Service	Subfunction	Supported
0x86	ResponseOnEvent	All excepted onComparisionOfValues and OnTimerInterrupt	Supported
0x87	LinkControl		User optional

Table 7.7: Support of UDS Services

7.5.2.1 General behavior using DEM interfaces

[SWS_Dcm_00007] [The `Dcm` module shall retrieve the `DTCStatusAvailabilityMask` by using the function `Dem_GetDTCStatusAvailabilityMask()`.]()

The mask `DTCStatusAvailabilityMask` reflects the status bits supported by the ECU.

Note : Masking is performed in the module `Dem` and does not need to be done on `Dcm` side (see `SWS_Dem_00657` in [12]).

[SWS_Dcm_00371] [To ensure consistent event related data during the reading sequence, the `Dcm` module shall lock the update of event related data before reading freeze frames or extended data records. The `Dcm` shall lock the update using the `Dem` API `Dem_DisableDTCRecordUpdate()`. After the locking the `Dcm` shall read the event related data by calls to:

- `Dem_SelectExtendedDataRecord()`
- `Dem_GetSizeOfExtendedDataRecordSelection()`
- `Dem_GetNextExtendedDataRecord()`
- `Dem_SelectFreezeFrameData()`
- `Dem_GetSizeOfFreezeFrameSelection()` and
- `Dem_GetNextFreezeFrameData()` After the event related data is read, the `Dcm` shall re-enable updates by calling `Dem_EnableDTCRecordUpdate()`.

]()

[SWS_Dcm_00702] [If function `Dem_DisableDTCRecordUpdate()` returns `DEM_PENDING`, the `Dcm` shall retry to get the lock in the next `Dcm_MainFunction()`.]()

[SWS_Dcm_00700] [When the `Dcm` module receives a request with the `DTCStatusMask` set to `0x00`, it shall send positive response and shall not use the `Dem` interface `Dem_SetDTCFilter()`.]()

Note: The parameter `DTCFormat` of the functions `Dem_ClearDTC()`, `Dem_SetDTCFilter()`, `Dem_SetFreezeFrameRecordFilter()` and `Dem_GetNextFilteredDTCAndFDC()` defines the output-format of the requested `DTC` values for the sub-sequent `API` calls. For the 2-byte ISO15031-6 [13] `DTC` format, the `DTCFormat` parameter shall be equal to `DEM_DTC_FORMAT_OBD`. For

the 2-byte ISO14229-1 **DTC** format, the **DTCFormat** parameter shall be equal to **DEM_DTC_FORMAT_UDS**.

[SWS_Dcm_01160] [When the **Dcm** module receives a request with the **DTCSeverityMask** set to 0x00, it shall send a positive response as specified in ISO14229-1 [1] and shall not use the **Dem** interface **Dem_SetDTCFilter()**]()

[SWS_Dcm_00835] [The **Dcm** shall call **Dem_SetDTCFilter** prior to **Dem_GetNumberOfFilteredDTC**, any sequence of **Dem_GetNextFilteredDTC**, any sequence of **Dem_GetNextFilteredDTCAndFDC**, as well as any sequence of **Dem_GetNextFilteredDTCAndSeverity**.]()

[SWS_Dcm_00836] [The **Dcm** shall call **Dem_SetFreezeFrameRecordFilter** prior to any sequence of **Dem_GetNextFilteredRecord**.]()

[SWS_Dcm_01127] [The **Dcm** module shall retrieve the **DTCSeverityAvailabilityMask** by using the function **Dem_GetDTCSeverityAvailabilityMask()**]()

Note: The mask **DTCSeverityAvailabilityMask** reflects the severity bits supported by the ECU.

[SWS_Dcm_01212] [If **Dem_DisableDTCRecordUpdate()** returns **DEM_WRONG_DTC**, the **Dcm** shall send a **NRC 0x31 (RequestOutOfRange)**.]()

[SWS_Dcm_01213] [If **Dem_DisableDTCRecordUpdate()** returns **DEM_WRONG_DTCORIGIN**, the **Dcm** shall send a **NRC 0x31 (RequestOutOfRange)**.]()

[SWS_Dcm_01234] [If **Dem_GetNextFilteredDTCAndSeverity()** returns **DEM_NO_SUCH_ELEMENT** and at least one matching element could be retrieved before, the **Dcm** shall send a positive response including these data elements.]()

[SWS_Dcm_01235] [If **Dem_GetNextFilteredDTCAndSeverity()** returns **DEM_NO_SUCH_ELEMENT** and no matching element could be retrieved before, the **Dcm** shall send a positive response only for service, subservice and mandatory data specified in ISO 14229-1 [1].]()

[SWS_Dcm_01242] [If **Dem_GetSizeOfExtendedDataRecordSelection()** returns **DEM_WRONG_DTC**, **DEM_WRONG_DTCORIGIN** or **DEM_NO_SUCH_ELEMENT**, the **Dcm** shall send a **NRC 0x31 (RequestOutOfRange)**]()

[SWS_Dcm_01250] [If **Dem_GetStatusOfDTC()** returns **DEM_WRONG_DTC** or **DEM_WRONG_DTCORIGIN**, the **Dcm** shall send a **NRC 0x31 (RequestOutOfRange)**.]()

[SWS_Dcm_01409] [If **Dem_GetStatusOfDTC()** returns **DEM_NO_SUCH_ELEMENT**, the **Dcm** shall send a positive response only for service and subservice.]()

[SWS_Dcm_01255] [If Dem_SetDTCTFilter() returns E_NOT_OK, the Dcm shall send a NRC 0x31 (RequestOutOfRange).] ()

7.5.2.2 Service 0x10 - Diagnostic Session Control

UDS Service 0x10 allows an external tester to enable different diagnostic sessions in the server. A diagnostic session enables a specific set of diagnostic services and/or functionality in the server. The service request contains the parameter:

- diagnosticSessionType

[SWS_Dcm_00250] [The Dcm module shall implement the UDS Service 0x10.] (*RS_Diag_04006*)

[SWS_Dcm_00307] [When responding to UDS Service 0x10, if the requested subfunction value is not configured in the ECU (configuration parameter DcmDspSessionLevel), the DSP submodule shall trigger a negative response with NRC 0x12 (SubFunction not supported).] ()

If the requested subfunction value is configured, the following steps are processed even if the requested session type is equal to the already running session type (see ISO14229-1 [1] Section 9.2).

[SWS_Dcm_00311] [The send confirmation function shall set the new diagnostic session type with DslInternal_SetSesCtrlType() and shall set the new timing parameters (P2ServerMax, P2ServerMax*) (see configuration parameters DcmDspSessionP2ServerMax and DcmDspSessionP2StarServerMax) and do the mode switch of the ModeDeclarationGroupPrototype DcmDiagnosticSessionControl by calling SchM_Switch_<bsnp>_DcmDiagnosticSessionControl() with the new diagnostic session type (see [SWS_Dcm_91019]).] (*RS_Diag_04015*)

[SWS_Dcm_00085] [The DSP submodule shall manage internally a read access for the dataIdentifier 0xF186 (ActiveDiagnosticSessionDataIdentifier) defined in ISO14229-1 [1].] ()

7.5.2.3 Service 0x11 - ECUReset

UDS Service ECUReset (0x11) allows an external tester to request a server reset. The service request contains parameter:

- resetType

[SWS_Dcm_00260] [The Dcm module shall implement the UDS Service ECUReset (0x11).] ()

[SWS_Dcm_00373] [On reception of a request for UDS Service 0x11 with the sub functions other than enableRapidPowerShutDown (0x04) or disableRapidPowerShutDown (0x05), the Dcm module shall trigger the mode switch of ModeDeclarationGroupPrototype DcmEcuReset equal to the received resetType. After the mode switch is requested the Dcm shall trigger the start of the positive response message transmission. Sub function hardReset (0x01) to HARD Sub function keyOffOnReset (0x02) to KEYONOFF Sub function softReset (0x03) to SOFT]()

Note: By this mode switch the Dcm informs the BswM to carry out necessary actions for the handling of this individual reset type. These actions can be configured within the BswM action list corresponding to the requested reset type. Here the integrator can also define if an ECU reset will finally be performed or not.

[SWS_Dcm_00594] [On the transmit confirmation (call to Dcm_TpTxConfirmation) of the positive response, the Dcm module shall trigger the mode switch of ModeDeclarationGroupPrototype DcmEcuReset to the mode EXECUTE (via SchM_Switch_<bsnp>_DcmEcuReset(RTE_MODE_DcmEcuReset_EXECUTE)).]()

Note: By this mode switch the Dcm requests the BswM to perform the final processing on the reset type according to the configured action list.

[SWS_Dcm_00818] [On reception of a request for UDS Service 0x11 with the sub functions enableRapidPowerShutdown (0x04) or disableRapidPowerShutdown (0x05), the Dcm module shall trigger the mode switch of ModeDeclarationGroupPrototype DcmRapidPowerShutDown: Sub function enableRapidPowerShutDown (0x04) to ENABLE_RAPIDPOWERSHUTDOWN, Sub function disableRapidPowerShutDown (0x05) to DISABLE_RAPIDPOWERSHUTDOWN]()

Note: If EnableRapidPowerShutdown is enabled, the ECU should shorten its power-down time.

[SWS_Dcm_00589] [In case the parameter DcmDspPowerDownTime is present, the Dcm shall set the powerDownTime in positive response to sub-service enableRapidPowerShutDown with value set in DcmDspPowerDownTime.]()

[SWS_Dcm_00834] [After sending the positive response of EcuReset (call of Dcm_TpTxConfirmation) the Dcm shall ignore all further requests during reset-processing.]()

[SWS_Dcm_CONSTR_6080] DcmDspEcuResetRow container configuration [One container DcmDspEcuResetRow shall be configured for each DcmDsdSubService (DcmDspEcuResetId matching to the DcmDsdSubServiceId) configured for the UDS service ECUReset (0x11) which does not have the corresponding DcmDsdSubServiceFnc parameter configured.](RS_Diag_04098)

7.5.2.4 Service 0x14 - Clear Diagnostic Information

UDS Service ClearDiagnosticInformation (0x14) requests an ECU to clear the error memory. The service request contains the parameter:

- groupOfDTC.

[SWS_Dcm_00247] [The *Dcm* module shall implement UDS Service 0x14.]()

[SWS_Dcm_01263] [Upon reception of a UDS Service ClearDiagnosticInformation (0x14) request with parameter groupOfDTC, the *Dcm* module shall call the API Dem_SelectDTC() with the following parameter values:

- ClientId: Client Id for this *Dcm* instance (see *DcmDemClientRef*)
- DTC: groupOfDTC from the service request
- DTCFormat: DEM_DTC_FORMAT_UDS
- DTCOrigin: DEM_DTC_ORIGIN_PRIMARY_MEMORY

] (*RS_Diag_04058*)

[SWS_Dcm_01400] [After call of Dem_SelectDTC() the *Dcm* shall call Dem_GetDTCSelectionResultForClearDTC() with the following parameter value:

- ClientId: Client Id for this *Dcm* instance (see *DcmDemClientRef*).

]()

[SWS_Dcm_01265] [In case Dem_GetDTCSelectionResultForClearDTC() returns DEM_WRONG_DTC, the *Dcm* shall send a NRC 0x31 (RequestOutOfRange).]()

[SWS_Dcm_01268] [In case Dem_GetDTCSelectionResultForClearDTC() returns E_OK, the *Dcm* module shall check if application allows to clear the DTC (according to the configuration parameter *DcmDspClearDTCCheckFnc*). If not, the *Dcm* module shall send a negative response with NRC set to value from the parameter "ErrorCode".]()

[SWS_Dcm_01269] [In case application allows to clear the DTC, the *Dcm* module shall check if the DTC can be cleared in the current mode condition (according to the configuration parameter *DcmDspClearDTCModeRuleRef*). If not, the *Dcm* module shall send the calculated negative response code of the referenced *DcmModeRule*.]()

[SWS_Dcm_00005] [If the condition checks are successfully done, the *Dcm* module shall call Dem_ClearDTC with the following parameter values:

- ClientId = Client Id for this *Dcm* instance (see *DcmDemClientRef*)

] (*RS_Diag_04058*)

[SWS_Dcm_00705] [In case Dem_ClearDTC() returns E_OK, the *Dcm* module shall send a positive response.]()

[SWS_Dcm_00707] [In case Dem_ClearDTC() returns DEM_CLEAR_FAILED, the *Dcm* shall send a negative response 0x22 (conditionsNotCorrect).]()

[SWS_Dcm_00708] [In case Dem_ClearDTC() returns DEM_WRONG_DTC, the *Dcm* shall send a negative response 0x31 (requestOutOfRange).]()

[SWS_Dcm_00966] [In case Dem_ClearDTC() returns DEM_CLEAR_BUSY, the Dcm shall send a negative response 0x22 (conditionsNotCorrect).]()

[SWS_Dcm_01060] [In case Dem_ClearDTC() returns DEM_CLEAR_MEMORY_ERROR, the Dcm shall trigger a negative response with NRC 0x72 (generalProgrammingFailure).]()

[SWS_Dcm_01408] [In case Dem_ClearDTC() returns DEM_WRONG_DTCORIGIN, the Dcm shall trigger a negative response 0x31 (requestOutOfRange).]()

7.5.2.5 Service 0x19 - Read DTC Information

Service 0x19 allows a client to read the status of server resident Diagnostic Trouble Code (DTC) information.

[SWS_Dcm_00248] [The Dcm module shall implement the UDS Service 0x19.]()

To setup the retrieval of specific data from the Dem module, the Dcm will call different filter APIs (Dem_SetDTCFilter(), Dem_SetFreezeFrameRecordFilter(), Dem_SelectFreezeFrameData() and Dem_SelectExtendedDataRecord()).

[SWS_Dcm_01043] [In case E_NOT_OK is returned by Dem_SetDTCFilter(), the Dcm module shall send a negative response with NRC 0x31 (requestOutOfRange).]()

[SWS_Dcm_01334] [For all sub-functions addressing user defined fault memory, before calling the appropriate Dem API, the Dcm shall add the value 0x0100 to the received selection request parameter MemorySelection in order to match the Dem_DTCOriginType.]()

7.5.2.5.1 Subfunctions 0x01, 0x07, 0x11 and 0x12

UDS Service 0x19 with subfunctions 0x01, 0x11 or 0x12 requests the ECU to report the number of DTCs matching tester-defined criteria. The service request contains the parameter:

- DTCStatusMask

UDS Service 0x19 with subfunction 0x07 requests the ECU to report the number of DTCs matching tester-defined criteria. The service request contains the parameters:

- DTCSeverityMask
- DTCStatusMask

[SWS_Dcm_00376] [When sending a positive response to UDS Service 0x19 with subfunction 0x01, 0x07, 0x11 or 0x12, the Dcm module shall use the data in the response message according to Table 7.8]()

Parameter name	Value
DTCStatusAvailabilityMask	DTCStatusAvailabilityMask (see [SWS_Dcm_00007]).
DTCFormatIdentifier	Value returned by Dem_GetTranslationType()
DTCCount	Value calculated according to [SWS_Dcm_00293]

Table 7.8: Subfunction 0x01, 0x07, 0x11 and 0x12 response values

[SWS_Dcm_00293] [When responding to UDS Service 0x19 with subfunction 0x01, 0x07, 0x11 or 0x12, the Dcm module shall calculate the number of DTCs using Dem_GetNumberOfFilteredDTC() after having set the DEM-filter with Dem_SetDTCFilter() using the parameter values according to Table 7.9.] (RS_Diag_04058, RS_Diag_04067)

	reportNumber OfDTC ByStatusMask	reportNumber OfDTCBySeveri- tyMaskRecord	reportNumberOf MirrorMemory DTCByStatus- Mask	reportNumberOf EmissionsRe- lated OBDDTCBySta- tusMask
	0x01	0x07	0x11	0x12
ClientId	Client Id for this Dcm instance (see DcmDem-ClientRef)	Client Id for this Dcm instance (see DcmDem-ClientRef)	Client Id for this Dcm instance (see DcmDem-ClientRef)	Client Id for this Dcm instance (see DcmDem-ClientRef)
DTCStatusMask	DTCStatusMask from request (see [SWS_Dcm_00700])	DTCStatusMask from request (see [SWS_Dcm_00700])	DTCStatusMask from request (see [SWS_Dcm_00700])	DTCStatusMask from request (see [SWS_Dcm_00700])
DTCFormat	DEM_DTC_FORMAT_UDS	DEM_DTC_FORMAT_UDS	DEM_DTC_FORMAT_UDS	DEM_DTC_FORMAT_UDS
DTCOrigin	PRIMARY_MEMORY	PRIMARY_MEMORY	MIRROR_MEMORY	DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY
FilterWithSeverity	NO	YES	NO	NO
DTCSeverityMask	Not relevant	DTCSeverityMask from request	Not relevant	Not relevant
FilterForFaultDetectionCounter	NO	NO	NO	NO

Table 7.9: Dem_SetDTCFilter() parameters values for subfunctions 0x01, 0x07, 0x11 and 0x12

7.5.2.5.2 Subfunctions 0x02, 0x0A, 0x0F, 0x13, 0x15 and 0x17

UDS Service 0x19 with subfunctions 0x02, 0x0F or 0x13 requests the DTCs (and their associated status) that match certain conditions. The service request contains the parameter:

- DTCStatusMask

UDS Service 0x19 with subfunction 0x0A requests all supported DTCs and their associated status. UDS Service 0x19 with subfunction 0x15 requests all DTCs with permanent status.

[SWS_Dcm_00377] [When sending a positive response to UDS Service 0x19 with subfunction 0x02, 0x0A, 0x0F, 0x13, 0x15 or 0x17, the Dcm module shall use the data in the response message according to Table 7.10.] ()

Parameter name	Value
DTCStatusAvailabilityMask	DTCStatusAvailabilityMask (see [SWS_Dcm_00007])
DTCAndStatusRecord	As defined in [SWS_Dcm_00008] and [SWS_Dcm_00378]
MemorySelection (subservice 0x17 only)	From request

Table 7.10: Subfunction 0x02, 0x0A, 0x0F, 0x13, 0x15 and 0x17 response values

[SWS_Dcm_01545] Read user defined memory by status mask authentication check [On reception of the UDS Service ReadDTCInformation (0x19) with subfunction reportUserDefMemoryDTCByStatusMask (0x17), the Dcm shall check if the access to the selected user defined memory in parameter MemorySelection is authenticated and read the DTC information only if:

- a `DcmDspReadDTCInformationUserDefinedFaultMemoryRole` is configured with `DcmDspReadDTCInformationUserDefinedFaultMemoryRoleId` matching the `MemorySelection` and the verification according to [SWS_Dcm_01479] was successful or
- the active white list on that connection has for that requested user defined memory selection one entry.

] ([RS_Diag_04233](#))

According to [SWS_Dcm_01545] the authentication checks are only executed if `DcmDspAuthentication` is configured. In case of a failed authentication the NRC handling is according to [SWS_Dcm_01544] and [SWS_Dcm_01551] applies.

[SWS_Dcm_00008] [On reception of a UDS Service 0x19 request with subfunction 0x02, 0x0F and 0x13 and if the result of the bitwise AND operation between the DTC-StatusMask received within the request message and the DTCStatusAvailabilityMask reported by the DEM is equal to 0, the Dcm module shall answer positively with 0 DTC.] ()

[SWS_Dcm_00378] [When responding to UDS Service 0x19 with subfunctions 0x02, 0x0A, 0x0F, 0x13, 0x15 or 0x17, the Dcm module shall obtain the records with DTCs (and their associated status) by repeatedly calling `Dem_GetNextFilteredDTC()` after having configured the filter with `Dem_SetDTCFilter()` using the parameter values according to Table 7.11.] ([RS_Diag_04058](#), [RS_Diag_04067](#))

	reportDTC ByStatus Mask	report Supported DTCs	report Mirror Memory DTCBy StatusMask	report Emissions Related OBDDTC ByStatus Mask	report DTCWith Permanent Status	report UserDef Memory DTCBy StatusMask
	0x02	0x0A	0x0F	0x13	0x15	0x17
ClientId	Client Id for this <i>Dcm</i> instance (see <i>DcmDem-ClientRef</i>)	Client Id for this <i>Dcm</i> instance (see <i>DcmDem-ClientRef</i>)	Client Id for this <i>Dcm</i> instance (see <i>DcmDem-ClientRef</i>)	Client Id for this <i>Dcm</i> instance (see <i>DcmDem-ClientRef</i>)	Client Id for this <i>Dcm</i> instance (see <i>DcmDem-ClientRef</i>)	Client Id for this <i>Dcm</i> instance (see <i>DcmDem-ClientRef</i>)
DTCStatus Mask	DTCStatus Mask from request (see <i>SWS_Dcm_00700</i>)	0x00	DTCStatus Mask from request (see <i>SWS_Dcm_00700</i>)	DTCStatus Mask from request (see <i>SWS_Dcm_00700</i>)	0x00	DTCStatus Mask from request (see <i>SWS_Dcm_00700</i>)
DTCFormat	DEM_DTC_FORMAT_UDS	DEM_DTC_FORMAT_UDS	DEM_DTC_FORMAT_UDS	DEM_DTC_FORMAT_UDS	DEM_DTC_FORMAT_UDS	DEM_DTC_FORMAT_UDS
DTCOrigin	PRIMARY_MEMORY	PRIMARY_MEMORY	MIRROR_MEMORY	DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY	PERMANENT_MEMORY	Memory Selection from request + 0x0100
FilterWith Severity	NO	NO	NO	NO	NO	NO
DTCSeverity Mask	Not relevant	Not relevant	Not relevant	Not relevant	Not relevant	Not relevant
FilterFor FaultDetectionCounter	NO	NO	NO	NO	NO	NO

Table 7.11: Dem_SetDTCFilter() parameters values for subfunctions 0x02, 0x0A, 0x0F, 0x13,0x15 and0x17

Note:

- The *Dcm* module can get an indication of the number of records that will be found using *Dem_GetNextFilteredDTC()* by using *Dem_GetNumberOfFilteredDTC()*. This allows the implementation to calculate the total size of the response before cycling through the DTCs.
- The value 0x00 used as DTCStatusMask for the subfunctions 0x0A and 0x15 disables the status byte filtering in *Dem_SetDTCFilter()*.

[SWS_Dcm_00828] [In case of paged buffer support is disabled, the *Dcm* module shall not insert zero-padded DTCs to the response of *UDS* Service 0x19 with subfunctions 0x02, 0x0A, 0x0F, 0x13, 0x15 or 0x17.]()

When using paged buffer mechanism, in some case, it's possible that the number of *DTC* matching the filter change between the calculation of the total size, needed for

the first page transmission, and the sending of the further pages. For this reason, the following requirements apply :

[SWS_Dcm_00587] [In case of paged buffer support is enabled, The *Dcm* shall limit the response size to the size calculated when sending the first page. If more DTCs match the filter after this sending, the additional DTCs shall not be considered.] ()

[SWS_Dcm_00588] [In case of paged buffer support is enabled, The *Dcm* shall pad the response with the size calculated when sending the first page. If less DTC match the filter after this sending, the missing DTCs shall be padded with 0 value as defined in 15031-6 [13].] ()

[SWS_Dcm_01229] [If *Dem_GetNextFilteredDTC()* returns DEM_NO_SUCH_ELEMENT and at least one matching element could be retrieved before, the *Dcm* shall send a positive response including these data elements.] ()

[SWS_Dcm_01230] [If *Dem_GetNextFilteredDTC()* returns DEM_NO_SUCH_ELEMENT and at no matching element could be retrieved before, the *Dcm* shall send a positive response only for service and subservice and additional parameters required within a positive response.] ()

7.5.2.5.3 Subfunction 0x08

UDS Service 0x19 with subfunction 0x08 requests the DTCs and the associated status that match a tester-defined severity mask record. The service request contains the following parameters:

- DTCSeverityMask
- DTCStatusMask

[SWS_Dcm_00379] [When sending a positive response to UDS Service 0x19 with subfunction 0x08, the *Dcm* module shall use the data in the response message according to Table 7.12.] ()

Parameter name	Value
DTCStatusAvailabilityMask	DTCStatusAvailabilityMask (see [SWS_Dcm_00007])
DTCAndSeverityRecord	As defined in [SWS_Dcm_00380]

Table 7.12: Subfunction 0x08 response values

[SWS_Dcm_00380] [When responding to UDS Service 0x19 with subfunction 0x08, the *Dcm* module shall obtain the DTCAndSeverityRecords by repeatedly calling *Dem_GetNextFilteredDTCAndSeverity()* after having configured the filter with *Dem_SetDTCFilter()* using the parameter values according to Table 7.13.] ()

	reportDTCBySeverityMaskRecord
ClientId	Client Id for this <i>Dcm</i> instance (see <i>DcmDemClientRef</i>)

	reportDTCBySeverityMaskRecord
DTCStatusMask	DTCStatusMask from request (see [SWS_Dcm_00700])
DTCFormat	DEM_DTC_FORMAT_UDS
DTCOrigin	PRIMARY_MEMORY
FilterWithSeverity	YES
DTCSeverityMask	DTCSeverityMask from request
FilterForFaultDetectionCounter	NO

Table 7.13: Dem_SetDTCFilter() parameters values for Subfunction 0x08

Note: The `Dcm` module can get an indication of the number of records that will be found using `Dem_GetNextFilteredDTCAndSeverity()` by using `Dem_GetNumberOfFilteredDTC()`.

7.5.2.5.4 Subfunction 0x09

`UDS` Service 0x19 with subfunction 0x09 requests the severity information of a DTC. The service request contains the parameter:

- DTCMaskRecord

[SWS_Dcm_00381] [When sending a positive response to `UDS` Service 0x19 with subfunction 0x09, the `Dcm` module shall use the data in the response message according to Table 7.14.] ()

Parameter name	Value
DTCStatusAvailabilityMask	DTCStatusAvailabilityMask (see [SWS_Dcm_00007])
DTCAndSeverityRecord	DTCSeverityMask: see [SWS_Dcm_01402] DTCFunctionalUnit: see [SWS_Dcm_01403] DTC: the given DTC of the request statusOfDTC : see [SWS_Dcm_01404]

Table 7.14: Subfunction 0x09 response values

[SWS_Dcm_01402] [To select the DTC, the `Dcm` module shall call the API `Dem_SelectDTC()` with the following parameter values:

- ClientId: Client Id for this `Dcm` instance (see `DcmDemClientRef`)
- DTC: DTC from the service request
- DTCFormat: DEM_DTC_FORMAT_UDS
- DTCOrigin: DEM_DTC_ORIGIN_PRIMARY_MEMORY

]()

[SWS_Dcm_01403] [To retrieve the DTCSeverityMask of the selected DTC, the `Dcm` shall call `Dem_GetSeverityOfDTC()` with the following parameter value:

- ClientId: Client Id for this `Dcm` instance (see `DcmDemClientRef`)

]()

[SWS_Dcm_01404] [To retrieve the DTCFunctionalUnit of the selected DTC, the Dcm shall call Dem_GetFunctionalUnitOfDTC() with the following parameter value:

- ClientId: Client Id for this Dcm instance (see DcmDemClientRef)

]()

[SWS_Dcm_01405] [To retrieve the statusOfDTC of the selected DTC, the Dcm shall call Dem_GetStatusOfDTC() with the following parameter value:

- ClientId: Client Id for this Dcm instance (see DcmDemClientRef)

]()

[SWS_Dcm_01226] [If Dem_GetFunctionalUnitOfDTC() returns DEM_WRONG_DTC or DEM_WRONG_DTCORIGIN, the Dcm shall send a NRC 0x31 (requestOutOfRange).]()

[SWS_Dcm_01240] [If Dem_GetSeverityOfDTC() returns DEM_WRONG_DTC, the Dcm shall send a NRC 0x31 (requestOutOfRange).]()

[SWS_Dcm_01406] [If Dem_GetStatusOfDTC() returns DEM_WRONG_DTC or DEM_WRONG_DTCORIGIN, the Dcm shall send a NRC 0x31 (requestOutOfRange).]()

7.5.2.5.5 Subfunctions 0x06/0x10/0x19

The UDS Service 0x19 with subfunction 0x06, 0x10 or 0x19 requests a specific Extended Data Records for a specific DTC. The service request contains the parameters:

- DTCMaskRecord
- DTCExtendedDataRecordNumber

[SWS_Dcm_01547] Read user defined memory extended data record authentication check [On reception of the UDS Service ReadDTCInformation (0x19) with subfunction reportUserDefMemoryDTCExtDataRecordByDTCNumber (0x19), the Dcm shall check if the access to the selected user defined memory in parameter MemorySelection is authenticated and read the DTC information only if:

- a DcmDspReadDTCInformationUserDefinedFaultMemoryRole is configured with DcmDspReadDTCInformationUserDefinedFaultMemoryRoleId matching the MemorySelection and the verification according to [SWS_Dcm_01522] was successful or
- the active white list on that connection has for that requested user defined memory selection one entry.

](RS_Diag_04233)

According to [SWS_Dcm_01537] the authentication checks are only executed if DcmDspAuthentication is configured. In case of a failed authentication the NRC handling is according to [SWS_Dcm_01485] and [SWS_Dcm_01544] applies.

[SWS_Dcm_00386] [Upon reception of UDS Service 0x019 with subfunction 0x06 or 0x10 or 0x19, the Dcm shall retrieve from the Dem the stored extended data records for the requested DTC and origin.] ()

[SWS_Dcm_00295] [When responding to UDS Service 0x19 with subfunction 0x06, 0x10 or 0x19, the Dcm module shall calculate the statusOfDTC by first calling Dem_SelectDTC() with the parameters values set according to Table 7.15 and then Dem_GetStatusOfDTC() with ClientId = Client Id for this Dcm instance (see DcmDemClientRef).] (RS_Diag_04058)

	reportDTC ExtendedData Record ByDTCNumber	report MirrorMemoryDTC ExtendedData Record ByDTCNumber	reportUserDef MemoryDTC ExtDataRecord ByDTCNumber
	0x06	0x10	0x19
ClientId	Client Id for this Dcm instance (see DcmDemClientRef)	Client Id for this Dcm instance (see DcmDemClientRef)	Client Id for this Dcm instance (see DcmDemClientRef)
DTC	DTCMaskRecord from request	DTCMaskRecord from request	DTCMaskRecord from request
DTCOrigin	PRIMARY_MEMORY	MIRROR_MEMORY	MemorySelection from request + 0x0100

Table 7.15: Dem_SelectDTC() parameters values for subfunctions 0x06, 0x10 and 0x19

[SWS_Dcm_00841] [If Dem_GetNextExtendedDataRecord() returns E_OK and Buf-Size 0 (empty buffer), the Dcm module shall omit the DTCExtendedDataRecordNumber for the related record in the response of service 0x19 0x06/0x10/0x19.] ()

[SWS_Dcm_00382] [When responding to UDS Service 0x19 with subfunction 0x06, 0x10 or 0x19, the Dcm module shall calculate the DTCExtendedDataRecord by first calling Dem_SelectExtendedDataRecord() with the parameter values set according to Table 7.16 and then call Dem_GetNextExtendedDataRecord() repeatedly until DEM_NO_SUCH_ELEMENT is returned.] ()

	reportDTCExtended- DataRecord ByDTCNumber	reportMirrorMemoryD Extended- DataRecord ByDTCNumber	reportUserDefMemory DTCExtDataRecord ByDTCNumber
	0x06	0x10	0x19
ClientId	Client Id for this Dcm instance (see DcmDemClientRef)	Client Id for this Dcm instance (see DcmDemClientRef)	Client Id for this Dcm instance (see DcmDemClientRef)
DTC	DTCMaskRecord from request	DTCMaskRecord from request	DTCMaskRecord from request

DTCOrigin	PRIMARY_MEMORY	MIRROR_MEMORY	Memory Selection from request + 0x0100
ExtendedDataNumber	DTCExtendedData RecordNumber from request	DTCExtendedData RecordNumber from request	DTCExtendedData RecordNumber from request

Table 7.16: Dem_SelectExtendedDataRecord() parameters values for subfunctions 0x06, 0x10 and 0x19

As required in [SWS_Dcm_00371], the `Dcm` module shall obtain the size of the extended data record by using `Dem_GetSizeOfExtendedDataRecordSelection()`.

7.5.2.5.6 Subfunction 0x03

UDS Service 0x19 with subfunction 0x03 allows an external tester to request the corresponding DTCs for all FreezeFrame records present in an ECU.

[SWS_Dcm_00300] [When sending a positive response to UDS Service 0x19 with subfunction 0x03, the `Dcm` module shall use the data in the response message according to Table 7.17.]()

Parameter name	Value
DTCRecord / DTCSnapshotRecord-Number	As defined in [SWS_Dcm_00299]

Table 7.17: Subfunction 0x03 response values

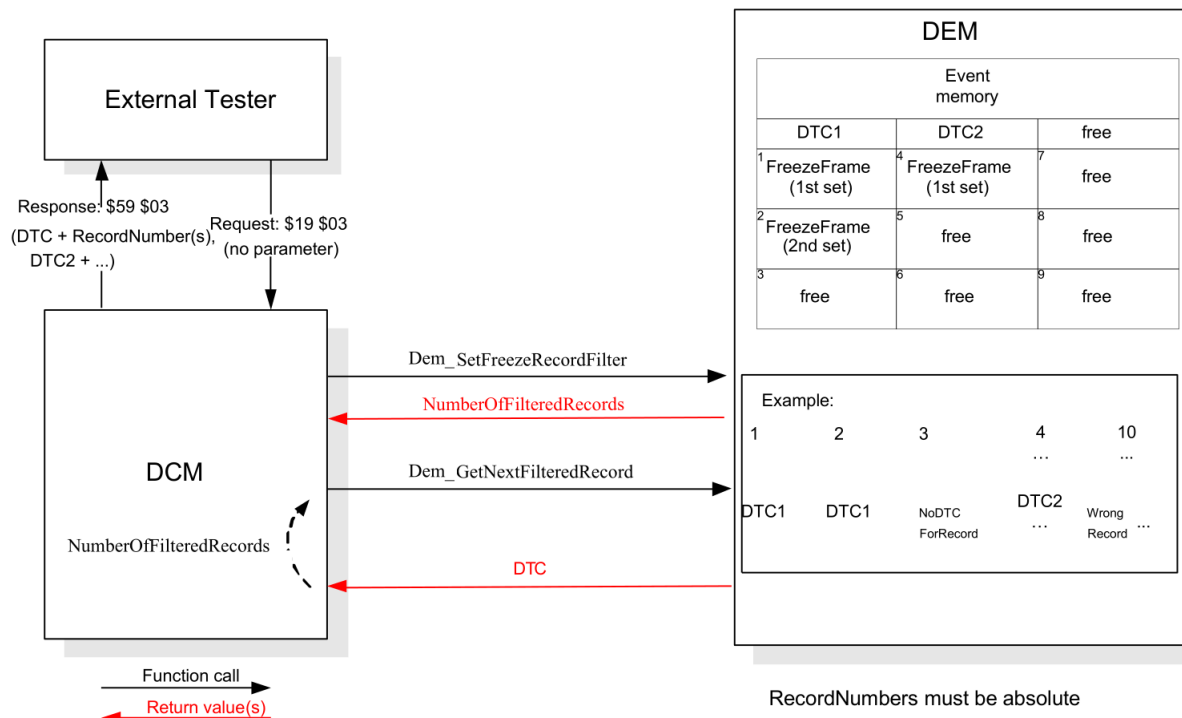


Figure 7.8: Request DTC Snapshot Record Identification

[SWS_Dcm_00298] [When UDS Service 0x19 with subfunction 0x03 is requested, the DSP submodule shall retrieve the number of stored freeze frame records by calling Dem_SetFreezeFrameRecordFilter() with DTCFormat equal to DEM_DTC_FORMAT_UDS and Dem_GetNumberOfFreezeFrameRecords.]()

[SWS_Dcm_00299] [When responding to UDS Service 0x19 with subfunction 0x03, the Dcm module shall obtain the consecutive DTCs and DTCSnapshotRecordNumbers by repeatedly calling Dem_GetNextFilteredRecord().]()

When using paged buffer mechanism, in some case, it's possible that the number of DTC matching the filter change between the calculation of the total size, needed for the first page transmission, and the sending of the further pages. For this reason, the requirement [SWS_Dcm_00587] and [SWS_Dcm_00588] shall be considered for the implementation of this subservice.

[SWS_Dcm_01237] [If Dem_GetNextFilteredRecord() returns DEM_NO_SUCH_ELEMENT and at least one matching element could be retrieved before, the Dcm shall send a positive response including these data elements.]()

[SWS_Dcm_01238] [If Dem_GetNextFilteredRecord() returns DEM_NO_SUCH_ELEMENT and no matching element could be retrieved before, the Dcm shall send a positive response only for service and subservice.]()

[SWS_Dcm_01256] [If Dem_SetFreezeFrameRecordFilter() returns E_NOT_OK, the Dcm shall send a NRC 0x31 (RequestOutOfRange).]()

7.5.2.5.7 Subfunctions 0x04 and 0x18

Using UDS Service 0x19 with subfunction 0x04 or 0x18, an external tester can request FreezeFrame information for one or all FreezeFrames of a specific DTC. The service request contains parameters:

- DTCMaskRecord
- DTCSnapshotRecordNumber

The subfunction 0x18 has an additional MemorySelection.

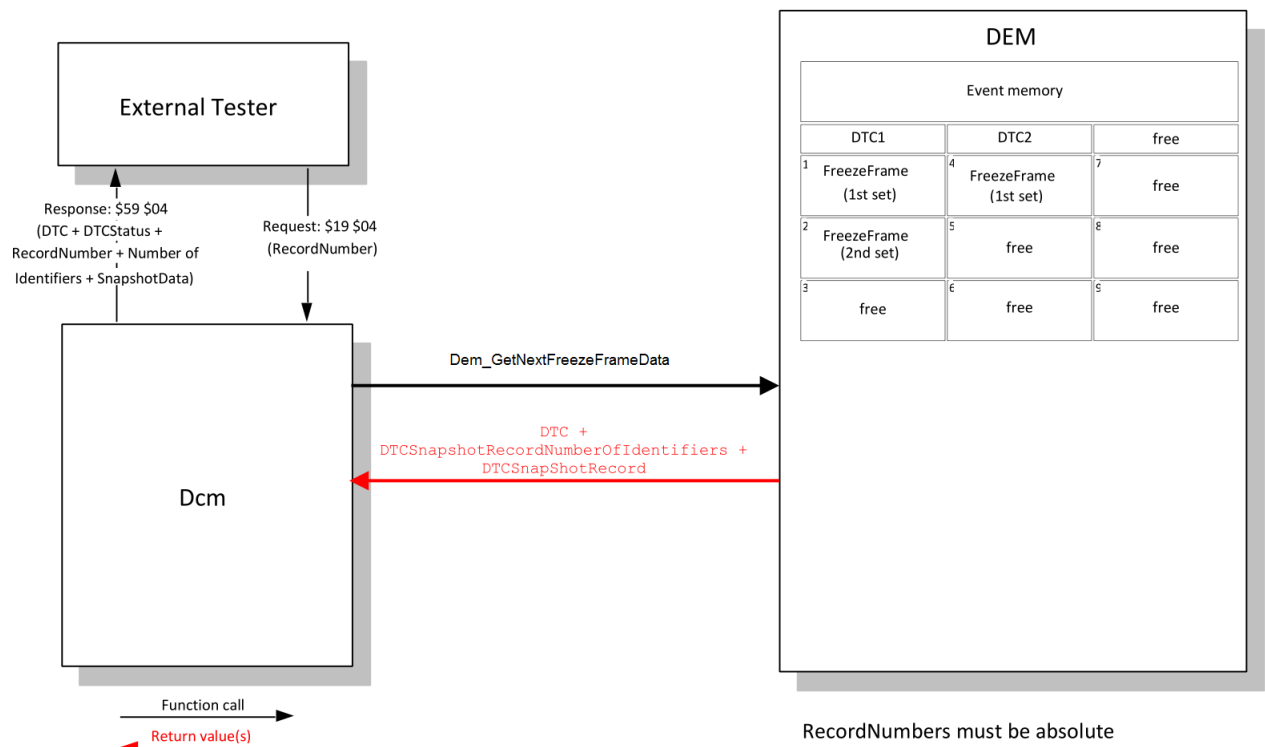


Figure 7.9: Request DTC Snapshot Record by Snapshot Record Number

[SWS_Dcm_01546] Read user defined memory snapshot record authentication check [On reception of the UDS Service ReadDTCInformation (0x19) with subfunction reportUserDefMemoryDTCSnapshotRecordByDTCNumber (0x18), the Dcm shall check if the access to the selected user defined memory in parameter MemorySelection is authenticated and read the DTC information only if:

- a DcmDspReadDTCInformationUserDefinedFaultMemoryRole is configured with DcmDspReadDTCInformationUserDefinedFaultMemoryRoleId matching the MemorySelection and the verification according to [SWS_Dcm_01522] was successful or
- the active white list on that connection has for that requested user defined memory selection one entry.

](RS_Diag_04233)

According to [SWS_Dcm_01546] the authentication checks are only executed if DcmDspAuthentication is configured. In case of a failed authentication the NRC handling is according to [SWS_Dcm_01485] and [SWS_Dcm_01551] applies.

[SWS_Dcm_00302] [When sending a positive response to UDS Service 0x19 with subfunction 0x04 or 0x18, the Dcm module shall use the data in the response message according to Table 7.18.]()

Parameter name	Value in Subservice 0x04	Value in Subservice 0x18
DTCAndStatusRecord	DTC from the request, statusOfDTC according to [SWS_Dcm_00383]	DTC from the request, statusOfDTC according to [SWS_Dcm_01147]
DTCSnapshotRecordNumber	The DTCSnapshotRecordNumber is contained in the output buffer from the Dem_GetNextFreezeFrame() call. see [SWS_Dcm_00384]	The DTCSnapshotRecordNumber is contained in the output buffer from the Dem_GetNextFreezeFrame() call. see [SWS_Dcm_00384].
DTCSnapshotRecordNumberOfIdentifiers / DTCSnapshotRecord	As defined in [SWS_Dcm_00384]	As defined in [SWS_Dcm_00384]
MemorySelection	n/a	From request

Table 7.18: Subfunction 0x04 and 0x18 response values

[SWS_Dcm_00383] [When responding to UDS Service 0x19 with subfunction 0x04, the Dcm module shall obtain the status of the DTC by first calling Dem_SelectDTC() with the following parameters:

- ClientId: Client Id for this Dcm instance (see DcmDemClientRef)
- DTC: DTC from the request
- DTCTOrigin: DEM_DTC_ORIGIN_PRIMARY_MEMORY

and then Dem_GetStatusOfDTC() with ClientId = Client Id for this Dcm instance (see DcmDemClientRef)](RS_Diag_04058)

[SWS_Dcm_01147] [When responding to UDS Service 0x19 with subfunction 0x18, the Dcm module shall obtain the status of the DTC by first calling Dem_GetStatusOfDTC() with the following parameters:

- ClientId: Client Id for this Dcm instance (see DcmDemClientRef)
- DTC: DTC from the request
- DTCTOrigin: Memory Selection from request + 0x0100

and then Dem_GetStatusOfDTC() with ClientId = DcmDemClientRef)](RS_Diag_04058)

[SWS_Dcm_00384] [Upon reception of UDS Service 0x019 with subfunction 0x04 or 0x18, the Dcm shall retrieve from the Dem the stored snapshot records for the requested DTC and origin.](RS_Diag_04058)

[SWS_Dcm_00441] [The `Dcm` module shall obtain the size of the data returned by `Dem` in `Dem_GetNextFreezeFrameData()` call by using `Dem_GetSizeOfFreezeFrameSelection()`.]()

To get the size of all `FreezeFrame` data, the `Dcm` module calls `Dem_SelectFreezeFrameData()` with `RecordNumber` set to `0xFF`.

[SWS_Dcm_01220] [If `Dem_GetNextFreezeFrameData()` returns `DEM_WRONG_DTC` or `DEM_WRONG_DTCORIGIN` the `Dcm` shall send a `NRC 0x31 (RequestOutOfRange)`.]()

[SWS_Dcm_01430] [When responding to `UDS Service 0x19` with subfunction `0x04`, or `0x18`, the `Dcm` shall collect the freeze frame data by first calling `Dem_SelectFreezeFrameData()` and then call `Dem_GetNextFreezeFrameData()` repeatedly until `DEM_NO_SUCH_ELEMENT` is returned.]()

[SWS_Dcm_01224] [If at least one of the requested freeze frame data is supported, the `Dcm` shall send a positive response. Otherwise the `Dcm` shall send a `NRC 0x31 (RequestOutOfRange)`.]()

[SWS_Dcm_01246] [If `Dem_GetSizeOfFreezeFrameSelection()` returns `DEM_WRONG_DTC`, `DEM_WRONG_DTCORIGIN` or `DEM_NO_SUCH_ELEMENT` the `Dcm` shall send a `NRC 0x31 (RequestOutOfRange)`.]()

7.5.2.5.8 Subfunction 0x05

`UDS Service 0x19` with subfunction `0x05` allows an external tester to request `FreezeFrame` information for a specific `FreezeFrame` record number. The service request contains parameter:

- `DTCStoredDataRecordNumber`

Due to `Dem` limitation, the diagnostic service `$19 05` is limited to the `OBD` legislative freeze frame.

[SWS_Dcm_00632] [On reception of service `0x19` with subfunction `0x05`, if the record number of the diagnostic request is different from `0x00`, the `Dcm` module shall send a negative response with `NRC 0x31 (request out of range)`.]()

[SWS_Dcm_00574] [When sending a positive response to `UDS Service 0x19` with subfunction `0x05` and `DTCStoredDataRecordNumber` is `0x00`, the `Dcm` module shall use the data in the response message according to Table 7.19.]()

Parameter name	Value
<code>DTCStoredDataRecordNumber</code>	<code>DTCStoredDataRecordNumber</code> from request (<code>0x00</code>)
<code>DTCAndStatusRecord</code>	<code>DTC</code> according to [SWS_Dcm_01193], <code>statusOfDTC</code> according to [SWS_Dcm_00389]
<code>DTCStoredDataRecordNumberOfIdentifiers / DTCStoredDataRecord</code>	As defined in [SWS_Dcm_00388]

Table 7.19: Subfunction 0x05 response values

[SWS_Dcm_00388] [When responding to UDS Service 0x19 with subfunction 0x05 and DTCStoredDataRecordNumber is 0x00, the Dcm shall compose the OBD Freeze-frame by looping all DcmDspPid and collecting all DcmDspPidData which are configured for service 0x02 by calling Dem_DcmReadDataOfOBDFreezeFrame() for the Data Element. The Dcm shall compose the DidId by adding 0xF400 to the Pid, and calculate padding and supported informations.] (RS_Diag_04058)

[SWS_Dcm_01193] [When responding to UDS Service 0x19 with subfunction 0x05 and DTCStoredDataRecordNumber is 0x00, the Dcm shall call Dem_DcmGetDTCOfOBDFreezeFrame() with FrameNumber 0x00 and DTCFormat DEM_DTC_FORMAT_UDS to retrieve the DTC of the provided FreezeFrame.] ()

[SWS_Dcm_00389] [When responding to UDS Service 0x19 with subfunction 0x05 and DTCStoredDataRecordNumber is 0x00, the Dcm module shall obtain the status of the DTC by first calling Dem_SelectDTC() with the following parameters:

- ClientId: Client Id for this Dcm instance (see DcmDemClientRef)
- DTC: DTC as defined in [SWS_Dcm_00388]
- DTCOrigin: DEM_DTC_ORIGIN_PRIMARY_MEMORY

and then Dem_GetStatusOfDTC() with the following parameter:

- ClientId: Client Id for this Dcm instance (see DcmDemClientRef)

] (RS_Diag_04058)

7.5.2.5.9 Subfunctions 0x0B, 0x0C, 0x0D and 0x0E

An external test tool can request the first occurred or most recent failed or confirmed DTC and associated status, by sending the UDS Service request 0x19 including one of the following sub-functions 0x0B, 0x0C, 0x0D, 0x0E

[SWS_Dcm_00392] [When sending a positive response to UDS Service 0x19 with subfunction 0x0B, 0x0C, 0x0D or 0x0E, the Dcm module shall use the data in the response message according to Table 7.20.] ()

Parameter name	Value
DTCStatusAvailabilityMask	DTCStatusAvailabilityMask (see [SWS_Dcm_00007])
DTCAndStatusRecord	The DTC is obtained according to [SWS_Dcm_00466], the StatusOfDtc is obtained according to [SWS_Dcm_00393]

Table 7.20: Subfunctions 0x0B, 0x0C, 0x0D and 0x0E response values

[SWS_Dcm_00393] [For the purpose of responding to UDS Service 0x19 with subfunctions 0x0B, 0x0C, 0x0D or 0x0E, the Dcm module shall obtain the StatusOfDtc by calling Dem_GetStatusOfDTC() with the following parameter values:

- ClientId :Client Id for this Dcm instance (see DcmDemClientRef)

- **DTC**: the **DTC** value as defined in [SWS_Dcm_00466]
- **DTCOrigin**: DEM_DTC_ORIGIN_PRIMARY_MEMORY

] (RS_Diag_04058)

[SWS_Dcm_00466] [For the purpose of responding to **UDS** Service 0x19 with subfunctions 0x0B, 0x0C, 0x0D or 0x0E, the **Dcm** shall obtain the **DTC** with Dem_GetDTCByOccurrenceTime() using the parameter values according to Table 7.21.]()

	reportFirstTestFailedDTC	reportFirstConfirmedDTC	reportMostRecentTestFailedDTC	reportMostRecentConfirmedDTC
	0x0B	0x0C	0x0D	0x0E
ClientId	Client Id for this Dcm instance (see DcmDem-ClientRef)	Client Id for this Dcm instance (see DcmDem-ClientRef)	Client Id for this Dcm instance (see DcmDem-ClientRef)	Client Id for this Dcm instance (see DcmDem-ClientRef)
DTCRequest	DEM_FIRST_FAILED_DTC	DEM_FIRST_DET_CONFIRMED_DTC	DEM_MOST_RECENT_FAILED_DTC	DEM_MOST_REC_DET_CONFIRMED_DTC

Table 7.21: Dem_GetDTCByOccurrenceTime() parameters values for subfunctions 0x0B, 0x0C, 0x0D and 0x0E

[SWS_Dcm_00766] [If the **Dcm** received DEM_NO_SUCH_ELEMENT by calling Dem_GetDTCByOccurrenceTime it shall reply with a positive response and empty DT-CAndStatusRecord.]()

7.5.2.5.10 Subfunction 0x14

An external test tool may request an ECU to report the FaultDetectionCounter for all **DTCs** with a "Prefailed" status, by sending a **UDS** Service request 0x19 with subfunction 0x14.

[SWS_Dcm_00464] [When sending a positive response to **UDS** Service 0x19 with subfunction 0x14, the **Dcm** module shall use the data in the response message according to Table 7.22.]()

Parameter name	Value
DTC	The DTC is obtained according from the call to Dem_GetNextFilteredDTCAndFDC()
DTCFaultDetectionCounter	The DTCFaultDetectionCounter is obtained according from the call to Dem_GetNextFilteredDTCAndFDC()

Table 7.22: Subfunction 0x14 response values

[SWS_Dcm_00465] [When responding to UDS Service 0x19 with subfunctions 0x14, the Dcm module shall obtain the DTCFaultCounter of every DTCs with status "prefailed" by repeatedly calling Dem_GetNextFilteredDTCAndFDC() after having configured the filter with Dem_SetDTCFilter() using the parameter values according to Table 7.23.] (RS_Diag_04058)

Parameter name	Value
ClientId	Client Id for this Dcm instance (see DcmDemClientRef)
DTCStatusMask	0x00
DTCFormat	DEM_DTC_FORMAT_UDS
DTCOrigin	PRIMARY_MEMORY
FilterWithSeverity	NO
DTCSeverityMask	Not relevant
FilterForFaultDetectionCounter	YES

Table 7.23: Dem_GetNextFilteredDTCAndFDC() parameters values for subfunctions 0x14

[SWS_Dcm_00681] [The Dcm module shall obtain the number of records that will be found using Dem_GetNextFilteredDTCAndFDC() by using Dem_GetNumberOfFilteredDTC().] ()

[SWS_Dcm_00519] [The calls to Dem_SetDTCFilter() with parameter FilterForFaultDetectionCounter set to YES shall be done in the context of the Dcm_MainFunction] ()

This allows the implementation to calculate the total size of the response before cycling through the DTCs.

When using paged buffer mechanism, in some case, it's possible that the number of DTC matching the filter change between the calculation of the total size, needed for the first page transmission, and the sending of the further pages. For this reason, the requirement [SWS_Dcm_00587] and [SWS_Dcm_00588] shall be considered for the implementation of this subservice.

[SWS_Dcm_01232] [If Dem_GetNextFilteredDTCAndFDC() returns DEM_NO_SUCH_ELEMENT and at least one matching element could be retrieved before, the Dcm shall send a positive response including these data elements.] ()

[SWS_Dcm_01233] [If Dem_GetNextFilteredDTCAndFDC() returns DEM_NO_SUCH_ELEMENT and no matching element could be retrieved before, the Dcm shall send a positive response only for service and subservice.] ()

7.5.2.5.11 Subfunction 0x42

UDS Service 0x19 with subfunction 0x42 requests WWH OBD DTCs matching a DTC status mask a severity mask record. The service request contains the following parameters:

- FunctionalGroupIdentifier
- DTCSeverityMask
- DTCStatusMask

[SWS_Dcm_01128] [The *Dcm* shall reject request messages for subFunction 0x42 with FunctionalGroupIdentifier unequal to 0x33 by returning *NRC* 0x31 (requestOutOfRange)]()

[SWS_Dcm_01129] [When sending a positive response to *UDS* Service 0x19 with subfunction 0x42, the *Dcm* module shall use the data in the response message according to Table 7.24.]()

Parameter name	Value
FunctionalGroupIdentifier	0x33
DTCStatusAvailabilityMask	Dem_GetDTCStatusAvailabilityMask (see [SWS_Dcm_00007])
DTCSeverityAvailabilityMask	Dem_GetDTCSeverityAvailabilityMask (see [SWS_Dcm_01127])
DTCFormatIdentifier	Dem_GetTranslationType (limited to values 0x04 and 0x02)
DTCAndSeverityRecord	As defined in [SWS_Dcm_01130]

Table 7.24: Subfunction 0x42 response values

[SWS_Dcm_01130] [When responding to *UDS* Service 0x19 with subfunction 0x42, the *Dcm* module shall obtain the DTCAndSeverityRecords by repeatedly calling Dem_GetNextFilteredDTCAndSeverity() after having configured the filter with Dem_SetDTCFilter() using the parameter values according to Table 7.25.]()

Parameter name	Value
ClientId	Client Id for this <i>Dcm</i> instance (see <i>DcmDemClientRef</i>)
DTCStatusMask	DTCStatusMask from request (see [SWS_Dcm_00700])
DTCFormat	DEM_DTC_FORMAT_UDS
DTCOrigin	DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY
FilterWithSeverity	YES
DTCSeverityMask	DTCSeverityMask from request
FilterForFaultDetectionCounter	NO

Table 7.25: Dem_GetNextFilteredDTCAndSeverity() parameters values for subfunctions 0x42

[SWS_Dcm_01131] [The return values of Dem_GetNextFilteredDTCAndSeverity shall be filled according to Table 7.26.]()

Parameter name	Value
DTCSeverity	DTCSeverity
DTCHighByte (MSB)	<i>DTC</i> (high byte)
DTCMiddleByte	<i>DTC</i> (middle byte)
DTCLowByte	<i>DTC</i> (low byte)
statusOfDTC	DTCStatus

Parameter name	Value
----------------	-------

Table 7.26: Dem_GetNextFilteredDTCAndSeverity return values

Note: The `Dcm` module can get an indication of the number of records that will be found using `Dem_GetNextFilteredDTCAndSeverity()` by using `Dem_GetNumberOfFilteredDTC()`.

7.5.2.5.12 Subfunction 0x55

With `UDS` Service 0x19 with sub-function 0x55 a client can retrieve a list of WWH-OBD `DTCs` with the "permanent `DTC`" status. The service request contains the following parameter:

- `FunctionalGroupIdentifier`

[SWS_Dcm_01343] [The `Dcm` shall only process request messages for sub-function 0x55 with `FunctionalGroupIdentifier` equal to 0x33.]()

[SWS_Dcm_01344] [The `Dcm` shall reject request messages for sub-function 0x55 with `FunctionalGroupIdentifier` unequal to 0x33 by returning `NRC` 0x31 (RequestOutOfRange).]()

[SWS_Dcm_01345] [When sending a positive response to `UDS` Service 0x19 with sub-function 0x55, the `Dcm` module shall use the following data in the response message according to Table 7.27.]()

Parameter name	Value
<code>FunctionalGroupIdentifier</code>	0x33
<code>DTCStatusAvailabilityMask</code>	<code>Dem_GetDTCStatusAvailabilityMask</code> (see [SWS_Dcm_00007])
<code>DTCFormatIdentifier</code>	<code>Dem_GetTranslationType</code> (limited to values 0x04 and 0x02)
<code>DTCAndStatusRecord</code>	As returned by <code>Dem_GetNextFilteredDTC()</code>

Table 7.27: Subfunction 0x55 response values

Note : When responding to `UDS` Service 0x19 with sub-function 0x55, the `Dcm` module could obtain the `DTCAndStatusRecords` by repeatedly calling `Dem_GetNextFilteredDTC()` after having configured the filter with `Dem_SetDTCFilter()` using the parameter values according to Table 7.28.

Parameter name	Value
<code>ClientId</code>	See <code>DcmDemClientRef</code>
<code>DTCStatusMask</code>	0x00
<code>DTCFormat</code>	<code>DEM_DTC_FORMAT_UDS</code>
<code>DTCOrigin</code>	<code>DEM_DTC_ORIGIN_PERMANENT_MEMORY</code>
<code>FilterWithSeverity</code>	NO
<code>DTCSeverityMask</code>	Not relevant
<code>FilterForFaultDetectionCounter</code>	NO

Parameter name	Value
----------------	-------

Table 7.28: Dem_GetNextFilteredDTCAndSeverity() parameters values for subfunctions 0x42

The `Dcm` module can get an indication of the number of records that will be found using `Dem_GetNextFilteredDTC()` by using `Dem_GetNumberOfFilteredDTC()`.

[SWS_Dcm_01346] [When responding to `UDS` Service 0x19 with sub-function 0x55 and `Dem_GetTranslationType` returns a `Dem_DTCTranslationFormatType` different to 0x02 (`DEM_DTC_TRANSLATION_SAEJ1939_73`) or 0x04 (`DEM_DTC_TRANSLATION_J2012DA_FORMAT_04`), the `Dcm` module shall return `NRC` 0x10 (generalReject).] ()

7.5.2.6 Service 0x22 - ReadDataByIdentifier

[SWS_Dcm_00253] [The `Dcm` module shall implement the `UDS` Service `ReadDataByIdentifier` (0x22)] ()

[SWS_Dcm_01335] [On reception of the `UDS` Service `ReadDataByIdentifier` (0x22), if the number of requested `DID` exceeds the configured maximum number of data identifiers (refer to configuration parameter `DcmDspMaxDidToRead`), the `Dcm` module shall send `NRC` 0x13 (Incorrect message length or invalid format)] ()

With `UDS` Service 0x22, the tester can request the value of one or more `DIDs`.

[SWS_Dcm_01548] Read UDS DID authentication check [On reception of the `UDS` Service `ReadDataByIdentifier` (0x22), the `Dcm` shall check if the access to all requested `DIDs` outside the range 0xF200-0xF8FF is authenticated and read the data identifiers only if:

- a `DcmDspDidReadRole` is configured for that `DID` and the verification according to [\[SWS_Dcm_01522\]](#) was successful or
- the active white list on that connection has for each requested `DID` one entry with read access that matches that `DID`.

] ([RS_Diag_04233](#))

[SWS_Dcm_01549] Read Dynamically defined DID authentication check [On reception of the `UDS` Service `ReadDataByIdentifier` (0x22), the `Dcm` shall check for all requested `DIDs` inside the range 0xF200-0xF3FF if the content is based of `DIDs` or parts of `DIDs` that are authenticated and read the data identifiers only if:

- a `DcmDspDidReadRole` is configured for those `DIDs` and the verification according to [\[SWS_Dcm_01522\]](#) was successful or
- the active white list on that connection has for each requested `DID` one entry with read access that matches those `DIDs`.

](RS_Diag_04233)

According to [SWS_Dcm_01537] the authentication checks are only executed if DcmDspAuthentication is configured. In case of a failed authentication the NRC handling is according to [SWS_Dcm_01544] and [SWS_Dcm_01551] applies.

[SWS_Dcm_00438] [On reception of the UDS Service ReadDataByIdentifier (0x22), for every requested DID the Dcm module shall check if the DID is supported (see configuration parameter DcmDspDid and DcmDspDidRange) If none of the requested DIDs is supported, the Dcm module shall send NRC 0x31 (Request out of range).]()

[SWS_Dcm_00651] [On reception of the UDS Service ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF, the Dcm module shall check if the DID can be dynamically defined (the DcmDspDidInfo it references has the DcmDspDidDynamicallyDefined set to true). If yes, if this DID has not been dynamically defined yet by calls to the DynamicallyDefineDataIdentifier (0x2C) service, i.e. it has no data sources defined, the Dcm module shall send NRC 0x31 (Request out of range).]()

[SWS_Dcm_00652] [On reception of the UDS Service ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF, if verification has been successfully done (see [SWS_Dcm_00651]) and the dynamic DID has been defined with a DID source (see [SWS_Dcm_00646]), the Dcm module shall use the configuration of this DID source to read the data.]()

[SWS_Dcm_00864] [On reception of the UDS Service ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF, if verification has been successfully done (see [SWS_Dcm_00651]) and the dynamic DID has been defined with a DID source (see [SWS_Dcm_00646]), the Dcm module shall do the session, security and mode dependencies checks for all source DIDs in case the configuration parameter DcmDspDDDIDcheckPerSourceDID is set to TRUE.]()

[SWS_Dcm_00865] [In case the configuration parameter DcmDspDDDIDcheckPerSourceDID is set to FALSE, there is no session, security or mode dependencies check for the source DIDs.]()

Note: In case there is a need to validate the session or security dependencies always, the DDDID should be cleared by any security and session transitions.

[SWS_Dcm_00653] [On reception of the UDS Service ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF, if verification has been successfully done (see [SWS_Dcm_00651]) and the dynamic DID has been defined with a memory address (see [SWS_Dcm_00646]), the Dcm module shall use the callout Dcm_ReadMemory to read the data.]()

[SWS_Dcm_00561] [If a DID is set as unused (DcmDspDidUsed set to FALSE), the Dcm shall consider the DID as not supported (according to [SWS_Dcm_00438]).]()

[SWS_Dcm_00433] [On reception of the UDS Service ReadDataByIdentifier (0x22), for every requested DID the Dcm module shall check if the DID has a Read access configured (see configuration parameter DcmDspDidRead in DcmDspDidInfo). If none

of the `DID` has a Read access, the `Dcm` module shall send `NRC 0x31` (Request out of range).]()

[SWS_Dcm_00434] [On reception of the `UDS Service ReadDataByIdentifier (0x22)`, for every requested `DID` the `Dcm` module shall check if the `DID` can be read in the current session (see configuration parameter `DcmDspDidReadSessionRef`). If none of the `DID` can be rendered in the current session, the `Dcm` module shall send a `NRC 0x31` (RequestOutOfRange).]()

[SWS_Dcm_00435] [On reception of the `UDS Service ReadDataByIdentifier (0x22)`, for every requested `DID` the `Dcm` module shall check if the `DID` can be read in the current security level (see configuration parameter `DcmDspDidReadSecurityLevelRef`). If not, the `Dcm` module shall send `NRC 0x33` (Security access denied).]()

[SWS_Dcm_00819] [On reception of the `UDS Service ReadDataByIdentifier (0x22)`, for every requested `DID` the `Dcm` module shall check if the `DID` can be read in the current mode condition (according to the configuration parameter `DcmDspDidReadModeRuleRef`). If not, the `Dcm` module shall send the calculated negative response of the referenced `DcmModeRule`.]()

[SWS_Dcm_00440] [If the requested `DID` references other `DID` using `DcmDspDidRef`, the `Dcm` module shall process the verification and the reading of every referenced `DID` and concatenate the response data without any gaps based on the sequence in the configuration.]()

[SWS_Dcm_CONSTR_6023] **`DcmDspDidRef` shall not reference the same `DID` reference twice** [`DcmDspDid` container shall not include the same `DcmDspDidRef` parameters more than once.]()

[SWS_Dcm_CONSTR_6057] **Dependency for `DcmDspDataEcuSignal`** [`DcmDspDataEcuSignal` shall be only present if `DcmDspDataUsePort` is set to `USE_ECU_SIGNAL`.]()

[SWS_Dcm_CONSTR_6058] **Dependency for `DcmDspDataEndianness`** [In case `DcmDspDataEndianness` is not configured, the `DcmDspDataDefaultEndianness` shall be used instead.]()

[SWS_Dcm_CONSTR_6061] **Dependency for `DcmDspDataReadDataLengthFnc`** [`DcmDspDataReadDataLengthFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_FNC_ERROR`

]()

[SWS_Dcm_CONSTR_6062] **Dependency for `DcmDspDataReadFnc`** [`DcmDspDataReadFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or

- `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_FNC_ERROR`

]()

[SWS_Dcm_01385] [If the `DID` `dataRecord` has bytes, not defined by any data element, the `Dcm` shall fill these bytes with the value `0x00`.]()

[SWS_Dcm_01431] [If the configuration parameter `DcmDspDidSize` is configured, the `Dcm` shall enforce the `DID` data to have the configured size.]()

[SWS_Dcm_01587]{DRAFT} Atomic read BndM read operation [After all verification (see [\[SWS_Dcm_00433\]](#), [\[SWS_Dcm_00434\]](#), [\[SWS_Dcm_00435\]](#), [\[SWS_Dcm_00436\]](#)) the `Dcm` module shall get for every requested `DID` with `DcmDspDidUsePort` set to `USE_ATOMIC_BNDM` and outside the `OBD` range (`F400-F8FF`) the data values by reading the data from the associated `BlockId` from the `BndM` (`DcmDspDidBndMBlockIdRef`) using the block specific reading function `BndM_GetBlockPtr_<BlockId.Shortname>`.] ([RS_Diag_04243](#))

7.5.2.6.1 UDS DID

[SWS_Dcm_00578] [On reception of the `UDS` Service `ReadDataByIdentifier` (`0x22`), for every requested `DID` outside the `OBD` range (`F400-F8FF`), after all verification (see [\[SWS_Dcm_00433\]](#), [\[SWS_Dcm_00434\]](#) and [\[SWS_Dcm_00435\]](#)), If the data is configured as a "ECU signal" of the `IoHwAb` (parameter `DcmDspDataUsePort`), the `Dcm` shall call the `Api IoHwAb_Dcm_Read<EcuSignalName >()` (parameter `DcmDspDataReadEcuSignal`) to get the Data. In this case, the requirements [\[SWS_Dcm_00439\]](#), [\[SWS_Dcm_00436\]](#) and [\[SWS_Dcm_00437\]](#) shall not apply.]()

[SWS_Dcm_00439] [On reception of the `UDS` Service `ReadDataByIdentifier` (`0x22`), for every requested `DID` outside the `OBD` range (`F400-F8FF`), the `Dcm` module shall request the application if the `DID` can be read by calling the configured function (if parameter `DcmDspDataUsePort` set to `USE_DATA_SYNC_FNC` or `USE_DATA_ASYNC_FNC` or `USE_DATA_ASYNC_FNC_ERROR`; see configuration parameter `DcmDspDataConditionCheckReadFnc`) on each data of the `DID` or call the associated `ConditionCheckRead` operation (if parameter `DcmDspDataUsePort` set to `USE_DATA_SYNC_CLIENT_SERVER` or `USE_DATA_ASYNC_CLIENT_SERVER` or `USE_DATA_ASYNC_CLIENT_SERVER_ERROR`). If not (one function returns `E_NOT_OK`), the `Dcm` module shall send a negative response with `NRC` set to value from the parameter "ErrorCode" of `DcmDspDataConditionCheckReadFnc` function or `ConditionCheckRead` operation.]()

[SWS_Dcm_00436] [On reception of the `UDS` Service `ReadDataByIdentifier` (`0x22`), for every requested `DID` outside the `OBD` range (`F400-F8FF`), the `Dcm` module shall for each signal (`DcmDspDidSignal`) with a dynamic data length (`DcmDspDataType` is set to `UINT8_DYN`): call either the configured function `DcmDspDataReadDataLengthFnc` (if parameter `DcmDspDataUsePort` set to `USE_DATA_SYNC_FNC` or

USE_DATA_ASYNC_FNC or USE_DATA_ASYNC_FNC_ERROR) or the associated ReadDataLength operation (if parameter DcmDspDataUsePort set to USE_DATA_ASYNC_CLIENT_SERVER or USE_DATA_ASYNC_CLIENT_SERVER_ERROR or USE_DATA_ASYNC_CLIENT_SERVER_ERROR) to get the data length in byte.]()

[SWS_Dcm_00437] [After all verification (see [SWS_Dcm_00433], [SWS_Dcm_00434], [SWS_Dcm_00435] and [SWS_Dcm_00436]) the Dcm module shall get for every requested DID outside the OBD range (F400-F8FF), all the data values by calling all the configured function (if parameter DcmDspDataUsePort set to USE_DATA_SYNC_FNC or USE_DATA_ASYNC_FNC or USE_DATA_ASYNC_FNC_ERROR; see configuration parameter DcmDspDataReadFnc) or call all the associated ReadData operations (if parameter DcmDspDataUsePort set to USE_DATA_SYNC_CLIENT_SERVER or USE_DATA_ASYNC_CLIENT_SERVER or USE_DATA_ASYNC_CLIENT_SERVER_ERROR) or read all the associated SenderReceiver interfaces (if parameter DcmDspDataUsePort set to USE_DATA_SENDER_RECEIVER or to USE_DATA_SENDER_RECEIVER_AS_SERVICE).]()

[SWS_Dcm_01432] [After all verification (see [SWS_Dcm_00433], [SWS_Dcm_00434], [SWS_Dcm_00435] and [SWS_Dcm_00436]) the Dcm module shall get for every requested DID with DcmDspDidUsePort set to USE_ATOMIC_SENDER_RECEIVER_INTERFACE, USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE or USE_ATOMIC_NV_DATA_INTERFACE and outside the OBD range (F400-F8FF) the data values by reading the associated sender-receiver or NvDataInterface DataServices_{DID}.]()

[SWS_Dcm_00560] [If the data is configured as a BlockId of the NvRam (parameter DcmDspDataUsePort), the Dcm shall call the Api NvM_ReadBlock() with the BlockId (parameter DcmDspDataBlockIdRef)]()

Note : For more information, refer to [14, SWS-NVRAMManager].

[SWS_Dcm_00638] [To serialize the required AUTOSAR data types (signed- and unsigned integer) into the response message of ReadDataByIdentifier responses, the target endianness configured in DcmDspDataEndianness shall be considered for DcmDspData elements having DcmDspDataUsePort set to USE_DATA_SENDER_RECEIVER, USE_DATA_SENDER_RECEIVER_AS_SERVICE, USE_ECU_SIGNAL.]()

[SWS_Dcm_CONSTR_6070] Dependency for DcmDspDataEndianness [In case DcmDspDataEndianness is not present, the DcmDspDataDefaultEndianness shall be used instead.]()

7.5.2.6.2 OBD DID

[SWS_Dcm_00481] [On reception of the UDS Service ReadDataByIdentifier (0x22), for every requested DID inside the OBD range (F400-F4FF), the Dcm module shall get the DID value as defined for OBD Service \$01 (see [SWS_Dcm_00407], [SWS_Dcm_00408], [SWS_Dcm_00943], [SWS_Dcm_00621], [SWS_Dcm_00622],

[SWS_Dcm_00623], [SWS_Dcm_00944] and [SWS_Dcm_00718]), if `DcmDspEnableObdMirror` is set to true.]()

[SWS_Dcm_00482] [On reception of the UDS Service ReadDataByIdentifier (0x22), for every requested `DID` inside the OBD Monitor range (F600-F6FF), the `Dcm` module shall get the `DID` value as defined for OBD Service \$06 (see [SWS_Dcm_00957], [SWS_Dcm_00958], [SWS_Dcm_00945] and [SWS_Dcm_00956])]()

[SWS_Dcm_00483] [On reception of the UDS Service ReadDataByIdentifier (0x22), for every requested `DID` inside the OBD InfoType range (F800-F8FF), the `Dcm` module shall get the `DID` value as defined for OBD Service \$09 (see [SWS_Dcm_00422], [SWS_Dcm_00423] and [SWS_Dcm_00949] without including the number of data items within the response, if `DcmDspEnableObdMirror` is set to true.]()

[SWS_Dcm_01195] [If `DcmDspEnableObdMirror` is set to true, an explicitly configured `DID` inside the OBD range (F400-F4FF) and the OBD InfoType range (F800-F8FF) shall use the UDS interface.]()

[SWS_Dcm_01197] [If `DcmDspEnableObdMirror` is set to FALSE, all requests within the OBD `DID` range shall use the UDS interface.]()

If `DcmDspEnableObdMirror` is set to FALSE ([SWS_Dcm_01197]) or the `DID` is explicitly configured inside the OBD `PID` range (F400-F4FF) ([SWS_Dcm_01195]), the access to the OBD data shall be given in the following way:

[SWS_Dcm_01379] [On reception of an UDS Service ReadDataByIdentifier (0x22) request with only "availability OBDDataIdentifier" as parameter, the `Dcm` shall respond with the corresponding supported (=configured) `DIDs` in the OBD range (F400-F4FF).]()

[SWS_Dcm_01380] [On reception of an UDS Service ReadDataByIdentifier (0x22) request with only OBDDataIdentifier that are not "availability OBDDataIdentifier", the `Dcm` shall obtain the current value of these OBDDataIdentifier by invoking the configured `Xxx_ReadData()` functions for every data of the OBDDataIdentifier and shall return these values as response to Service 0x22.]()

[SWS_Dcm_01381] [On reception of an UDS Service ReadDataByIdentifier (0x22) request with a mixture of "availability OBDDataIdentifier" and not "availability OBDDataIdentifier", this request shall be ignored by the `Dcm`.]()

[SWS_Dcm_01382] [If an OBDDataIdentifier contains support information (presence of `DcmDspDidDataSupportInfo` container), the `Dcm` shall add the support information in the diagnostic response.]()

[SWS_Dcm_01383] [If an OBDDataIdentifier contains support information (presence of `DcmDspDidDataSupportInfo` container), the `Dcm` shall calculate the support information value according to the available data for this `DID`: for every `DcmDspData` container existing for this `DID`, the associated support information bits, referenced in `DcmDspDidDataSupportInfo`, shall be set to one.]()

[SWS_Dcm_01384] [When responding to UDS Service ReadDataByIdentifier (0x22) with OBDDataIdentifier, the Dcm shall put fill-bytes between DcmDspData in the OBD-DataIdentifier whenever content bytes are missing, in order to fit to the DID size (see configuration parameter DcmDspDidSize).]()

[SWS_Dcm_01386] [To serialize the required AUTOSAR data types (signed and unsigned integer) into the response message of ReadDataByIdentifier (0x22) OBD-DataIdentifier responses the target endianness configured in DcmDspDataEndianness shall be considered for DcmDspData elements having DcmDspDataUsePort set to {USE_DATA_SENDER_RECEIVER, USE_DATA_SENDER_RECEIVER_AS_SERVISE, USE_ECU_SIGNAL}. In case DcmDspDataEndianness is not present, the DcmDspDataDefaultEndianness shall be used instead.]()

If DcmDspEnableObdMirror is set to FALSE or the DID is explicitly configured inside the OBD InfoType range (F800-F8FF), the access to the OBD data shall be given in the following way:

[SWS_Dcm_01387] [On reception of an UDS Service ReadDataByIdentifier (0x22) request with one or more "availability OBDInfoTypeDataIdentifier" as parameter, the Dcm module shall respond with the corresponding supported (=configured) DIDs in the OBD range (F800-F8FF).]()

[SWS_Dcm_01388] [On reception of an UDS Service ReadDataByIdentifier (0x22) request with "availability OBDInfoTypeDataIdentifier" together with other OBDInfoTypeDataIdentifier as parameter, the Dcm module shall ignore the request.]()

[SWS_Dcm_01389] [On reception of an UDS Service ReadDataByIdentifier (0x22) request with an OBDInfoTypeDataIdentifier that is not an "availability OBDInfoTypeDataIdentifier", the Dcm module shall obtain the value of this OBDInfoTypeDataIdentifier by invoking all the configured Xxx_ReadData() function for every data of this OBD-InfoTypeDataIdentifier and shall return the value as response to Service 0x22.]()

7.5.2.7 Service 0x24 - ReadScalingDataByIdentifier

[SWS_Dcm_00258] [The Dcm module shall implement the UDS Service ReadScalingDataByIdentifier (0x24).]()

To obtain scaling information, the tester can invoke UDS Service 0x24 with the 2-byte DID as parameter. The configuration of the Dcm contains for each configured DID:

- The 2-byte DID (see configuration parameter DcmDspDidIdentifier)
- For every data of the DID :
- The function GetScalingInformation (see configuration parameters DcmDspDataGetScalingInfoFnc and DcmDspDataUsePort)
- The length of the ScalingInfo returned by the GetScalingInformation function (see configuration parameter DcmDspDataScalingInfoSize)

[SWS_Dcm_00394] [On reception of a request for UDS Service ReadScalingByIdentifier, the Dcm module shall call every function Xxx_GetScalingInformation() configured for every data of the DID received in the request and return the data received in the response.]()

[SWS_Dcm_CONSTR_6060] Dependency for DcmDspDataGetScalingInfoFnc [DcmDspDataGetScalingInfoFnc shall be only present if:

- DcmDspDataUsePort is set to USE_DATA_SYNCH_FNC or
- DcmDspDataUsePort is set to USE_DATA_ASYNC_FNC or
- DcmDspDataUsePort is set to USE_DATA_ASYNC_FNC_ERROR

]()

7.5.2.8 Service 0x27 - SecurityAccess

[SWS_Dcm_00252] [The Dcm module shall implement the UDS Service SecurityAccess (0x27)](*RS_Diag_04005*)

The purpose of this service is to provide a means to access data and/or diagnostic services, which have restricted access for security, emissions, or safety reasons.

[SWS_Dcm_00321] [If the request length is correct, the DSP submodule shall check if the requested subfunction value (access type) is configured in the ECU (see configuration parameter DcmDspSecurityLevel). If the requested subfunction value is not configured, the DSP submodule shall trigger a negative response with NRC 0x12 (SubFunction not supported).]()

[SWS_Dcm_00323] [If the requested subfunction value is configured and a service with subfunction type "requestSeed" (= odd value) has been received and if the requested access type is already active (see Dcm_GetSecurityLevel), the DSP submodule shall set the seed content to 0x00.]()

[SWS_Dcm_00324] [In the other case than the one described in [SWS_Dcm_00323] (access type is not active or "send key" request), if DcmDspSecurityUsePort is set to USE_ASYNC_CLIENT_SERVER, the DSP submodule shall call the configured operation Xxx_GetSeed() (in case "request seed" is received) or Xxx_CompareKey() (in case "send key" is received).]()

[SWS_Dcm_00862] [On reception of the UDS Service SecurityAccess (0x27) with subfunction type "requestSeed" and if the requested access type is not already active, the Dcm module shall request a seed by calling the configured Xxx_GetSeed() function (if the configuration parameter DcmDspSecurityUsePort is set to USE_ASYNC_FNC, refer to configuration parameter DcmDspSecurityGetSeedFnc).]()

Note : If the static seed mechanism is used, the processing needs to be done by the application implementing the Xxx_GetSeed() and Xxx_CompareKey() functions.

[SWS_Dcm_CONSTR_6077] Dependency for DcmDspSecurityGetSeedFnc [DcmDspSecurityGetSeedFnc shall be present only if DcmDspSecurityUsePort is set to USE_ASYNCH_FNC.]()

[SWS_Dcm_00863] [On reception of the UDS Service SecurityAccess (0x27) with subfunction type "sendKey", if the requested access type is not already active and if the "request seed" for the related access type was executed successfully, the Dcm module shall request the result of a key comparison by calling the configured Xxx_CompareKey() function (if the configuration parameter DcmDspSecurityUsePort is set to USE_ASYNCH_FNC, refer to configuration parameter DcmDspSecurityCompareKeyFnc).]()

[SWS_Dcm_CONSTR_6075] Dependency for DcmDspSecurityCompareKeyFnc [DcmDspSecurityCompareKeyFnc shall be configured only if DcmDspSecurityUsePort is set to USE_ASYNCH_FNC.]()

The following list gives as an example, which errors can be detected by the security access service and stored in the error code information:

- RequestSequenceError (NRC 0x24), when invalid access type is send at "send key",
- RequiredTimeDelayNotExpired (NRC 0x37), when time delay is active (see configuration parameter DcmDspSecurityDelayTime),
- ExceededNumberOfAttempts (NRC 0x36), when number of attempts to get security access reaches or exceeds the configured limit (see configuration parameter DcmDspSecurityNumAttDelay), and
- InvalidKey (NRC 0x35), when invalid key is send at "send key".

[SWS_Dcm_00325] [If the operation CompareKey() returns E_OK, the DSP submodule shall set the new access type with DslInternal_SetSecurityLevel()(see the conversion formula given in ECUC_Dcm_00754 DcmDspSecurityLevel).]()

[SWS_Dcm_01397] [If Xxx_CompareKey() returns value DCM_E_COMPARE_KEY_FAILED and DcmDspSecurityNumAttDelay is configured, the Dcm shall increment the attempt counter of the security level for which the sendKey subfunction request failed.](RS_Diag_04005)

[SWS_Dcm_00660] [If Xxx_CompareKey() returns value DCM_E_COMPARE_KEY_FAILED and if the number of failed attempts to enter the requested security level (AttemptCounter) is less than the value configured for the DcmDspSecurityNumAttDelay parameter of the requested security level, the Dcm module shall send a negative response with NRC 0x35 (InvalidKey) and shall not change the Dcm internal security level. Note: if DcmDspSecurityNumAttDelay parameter is not configured, the number of failed attempts to enter the requested security level (AttemptCounter) is not considered.](/)

[SWS_Dcm_01349] [If Xxx_CompareKey() returns value DCM_E_COMPARE_KEY_FAILED and if the number of failed attempts to enter

the requested security level (AttemptCounter) is greater or equal than the value configured for the `DcmDspSecurityNumAttDelay` parameter of the requested security level, the `Dcm` module shall start the `SecurityDelayTimer` with the value configured in `DcmDspSecurityDelayTime` for the `SecurityLevel` which was requested in the failed request, send a negative response with NRC 0x36 (exceededNumberOfAttempts) and shall not change the `Dcm` internal security level.]()

[SWS_Dcm_01150] [If `Xxx_CompareKey()` returns value `E_NOT_OK`, the `Dcm` module shall send a negative response with NRC code equal to the `ErrorCode` parameter value and shall not increment the attempt counter or change the `Dcm` internal security level.]()

[SWS_Dcm_01350] [While the `SecurityDelayTimer` of `SecurityLevel` is not yet elapsed, the `Dcm` module shall send a negative response with NRC 0x37 (requiredTimeDelayNotExpired) on a `SecurityAccess` (0x27) requestSeed subfunction request for that `SecurityLevel`.]()

[SWS_Dcm_00659] [If `Xxx_GetSeed()` returns value `E_NOT_OK`, the `Dcm` module shall send a negative response with NRC code equal to the `ErrorCode` parameter value.]()

7.5.2.9 Service 0x28 - CommunicationControl

[SWS_Dcm_00511] [The `Dcm` module shall implement the `CommunicationControl` (service 0x28) of the Unified Diagnostic Services.]()

[SWS_Dcm_00512] [On invocation of the sent confirmation function of the `UDS` Service `CommunicationControl` (0x28) from `DSD` with the subnet parameter of the request equal to 0x00, the `Dcm` shall do for each `NetworkHandle` (see `DcmDspAllComMChannelRef`) which is configured in `DcmDspComControlAllChannel`:

1. trigger the mode switch `Dcm_CommunicationControl_<Network> ModeDeclarationGroupPrototype` to the mode corresponding the `communicationType` and `controlType` parameter from the `CommunicationControl` request.
2. call the Api `BswM_Dcm_CommunicationMode_CurrentState` with the parameters `NetworkHandleType` and `Dcm_CommunicationModeType` corresponding to the `communicationType` and `controlType` parameter from the `CommunicationControl` request (see `Dcm_CommunicationModeType` definition).

]()

[SWS_Dcm_00785] [On invocation of the sent confirmation function of the `UDS` Service `CommunicationControl` (0x28) from `DSD` with the subnet parameter of the request equal to 0x0F (`CommunicationControl` on the network which request is received on), the `Dcm` shall do for the `NetworkHandle` (see `DcmDslProtocolComMChannelRef`) of the current received `DcmDslProtocolRxPduRef`:

1. trigger the mode switch `Dcm_CommunicationControl_<Network> ModeDeclarationGroupPrototype` to the mode corresponding to the `communicationType` and `controlType` parameter from the `CommunicationControl` request.
2. call the Api `BswM_Dcm_CommunicationMode_CurrentState` with the parameters `NetworkHandleType` and `Dcm_CommunicationModeType` corresponding to the `communicationType` and `controlType` parameter from the `CommunicationControl` request (see `Dcm_CommunicationModeType` definition)

}|()

[SWS_Dcm_00786] [On invocation of the sent confirmation function of the `UDS Service CommunicationControl (0x28)` from `DSD` with the subnet parameter of the request between `0x01` and `0x0E`, the `Dcm` shall check if the received subnet parameter (see `DcmDspSubnetNumber`) is supported. In case it is not supported a `NegativeResponse code 0x31` shall be sent. In case it is supported the `Dcm` shall do for the corresponding `NetworkHandle` (see `DcmDspSpecificComMChannelRef`) of the received subnet parameter (see `DcmDspSubnetNumber`):

1. trigger the mode switch `Dcm_CommunicationControl_<Network> ModeDeclarationGroupPrototype` to the mode corresponding the `communicationType` and `controlType` parameter from the `CommunicationControl` request.
2. call the Api `BswM_Dcm_CommunicationMode_CurrentState` the parameters `NetworkHandleType` and with `Dcm_CommunicationModeType` corresponding the `communicationType` and `controlType` parameter from the `CommunicationControl` request (see `Dcm_CommunicationModeType` definition)

}|()

For some use-cases the `Dcm` may re-enable the `CommunicationControl` due to external changed mode conditions:

[SWS_Dcm_00753] [In case that the referenced `ModeRule` (see `ECUC_Dcm_00943`) is not fulfilled anymore for a `NetworkHandle` which is currently in a state other than `DCM_ENABLE_RX_TX_NORM_NM`, the `Dcm` shall:

1. switch the mode group `Dcm_CommunicationControl_<Network> ModeDeclarationGroupPrototype` to `DCM_ENABLE_RX_TX_NORM_NM`
2. call `BswM_Dcm_CommunicationMode_CurrentState` with the parameters `NetworkHandleType` set to the corresponding `NetworkHandle` of the network and `RequestedCommunicationMode` set to `DCM_ENABLE_RX_TX_NORM_NM`

}|()

[SWS_Dcm_00860] [For a `NetworkHandle` which is currently in a state other than `DCM_ENABLE_RX_TX_NORM_NM` if the `Dcm` is transitioning to default session or upon any diagnostic session change where the new session does not support `UDS Service CommunicationControl` anymore, the `Dcm` shall:

1. switch the mode group `Dcm_CommunicationControl_<Network> ModeDeclarationGroupPrototype` to `DCM_ENABLE_RX_TX_NORM_NM`
2. call `BswM_Dcm_CommunicationMode_CurrentState` with the parameters `NetworkHandleType` set to the corresponding `NetworkHandle` of the network and `RequestedCommunicationMode` set to `DCM_ENABLE_RX_TX_NORM_NM`

]()

Note: the `NetworkHandles` to be considered are all `ComM` channels which are referenced from either `DcmDspSpecificComMChannelRef`, `DcmDspAllComMChannelRef` or `DcmDspComControlSubNodeComMChannelRef`.

[SWS_Dcm_01077] [If a `CommunicationControl` Request with the sub-function "enableRxAndDisableTxWithEnhancedAddressInformation" is received, the `Dcm` shall check the "nodeIdentification-Number" listed as `DcmDspComControlSubNodeId` and for the referenced network (see `DcmDspComControlSubNodeComMChannelRef`), it shall do the followings:

1. trigger the mode switch `Dcm_CommunicationControl_<Network> ModeDeclarationGroupPrototype` to the mode corresponding the `communicationType` and `controlType` parameter from the `CommunicationControl` request.
2. call the Api `BswM_Dcm_CommunicationMode_CurrentState` with the parameters `NetworkHandleType` and `Dcm_CommunicationModeType` corresponding to the `communicationType` and `controlType` parameter from the `CommunicationControl` request (see `Dcm_CommunicationModeType` definition).

The analogue `controlType` `enableRxAndDisableTx` shall be used with the the following existing `Dcm_CommunicationModeType` values:

- `DCM_ENABLE_RX_DISABLE_TX_NORM`
- `DCM_ENABLE_RX_DISABLE_TX_NM`
- `DCM_ENABLE_RX_DISABLE_TX_NORM_NM`.

]()

[SWS_Dcm_01078] [The `Dcm` shall trigger a negative response with `NRC` 0x31 (`RequestOutOfRange`), if a `CommunicationControl` Request with the sub-function "enableRxAndDisableTxWithEnhancedAddressInformation" and a "nodeIdentification-Number" which is not listed as `DcmDspComControlSubNodeId` is received.]()

[SWS_Dcm_01079] [If a `CommunicationControl` Request with the sub-function "enableRxAndTxWithEnhancedAddressInformation" is received, the `Dcm` shall check the "nodeIdentification-Number" listed as `DcmDspComControlSubNodeId` and for the referenced network (see `DcmDspComControlSubNodeComMChannelRef`) it shall do the followings:

1. trigger the mode switch `Dcm_CommunicationControl_<Network> ModeDeclarationGroupPrototype` to the mode corresponding the `communicationType` and `controlType` parameter from the `CommunicationControl` request.
2. call the Api `BswM_Dcm_CommunicationMode_CurrentState` with the parameters `NetworkHandleType` and `Dcm_CommunicationModeType` corresponding to the `communicationType` and `controlType` parameter from the `CommunicationControl` request (see `Dcm_CommunicationModeType` definition).

The analogue `controlType enableRxAndTx` shall be used with this the following existing `Dcm_CommunicationType` values :

- `DCM_ENABLE_RX_TX_NORM`
- `DCM_ENABLE_RX_TX_NM`
- `DCM_ENABLE_RX_TX_NORM_NM`.

]()

[SWS_Dcm_01080] [The `Dcm` shall trigger a negative response with `NRC 0x31` (`RequestOutOfRange`), if a `CommunicationControl Request` with the sub-function "`enableRxAndTxWithEnhancedAddressInformation`" and a "`nodeIdentification-Number`" which is not listed as `DcmDspComControlSubNodeId` is received.]()

[SWS_Dcm_01081] [If `DcmDspComControlSubNodeUsed` is set to `FALSE` the sub-system (`DcmDspComControlSubNode`) is not available in this configuration.]()

[SWS_Dcm_01082] [If `DcmDspComControlSubNodeUsed` is set to `TRUE` the sub-system (`DcmDspComControlSubNode`) is available in this configuration.]()

Note : Condition checks (i.e. `NRC 22` checks) on `CommunicationType` and `NetworkType` as well as check of `CommunicationType` support (i.e. `NRC 0x31` check for `CommunicationType`) are not directly supported by the `Dcm`. `Supplier/manufacture`r notifications can be used.

7.5.2.10 Service 0x29 - Authentication

The UDS service `Authentication (0x29)` is used to provide dynamic means to control the access to diagnostic services. Execution of certain diagnostic services might have impact on safety, emissions or the vehicle. Controlling the access to diagnostic services via certificates provide means to control the access to diagnostic services after production. The access to these resources can be limited in time or bound to certain vehicles or ECUs only. AUTOSAR `Dcm` provides an out of the box solution for authenticated diagnostics with a given semantics of the UDS service `Authentication`. Even ISO 14229-1 [x] leaves more freedom, the `Dcm` will use only the functionality specified in this chapter. Further interpretations, certificate types or certificate content are out of scope of the AUTOSAR `Dcm` module. At the point of time this specification was developed, only the DIS (Draft International Standard) of ISO 14299-1:2018 is available.

Unless the official international standard ISO 14299-1:2018 is released, AUTOSAR will implement a solution based on this not public available draft.

[SWS_Dcm_01559] Support of UDS service authentication [The Dcm shall implement the UDS service Authentication (0x29) for sub-functions:

- deAuthenticate
- verifyCertificateUnidirectional
- verifyCertificateBidirectional
- proofOfOwnership
- authenticationConfiguration

]([RS_Diag_04230](#))

Note: AUTOSAR Dcm only implements the authentication via PKI certificated exchange. Authentication with challenge-response (ACR) is out of scope of AUTOSAR. If it is required it needs a full custom implementation using existing Dcm callouts for custom service processing.

[SWS_Dcm_01551] NRC Handling for UDS service authentication [The Dcm shall follow the NRC handling as defined for ISO 14229-1:2018 for UDS service authentication. This includes the NRC codes as well as the order of individual NRC checks.] ([RS_Diag_04230](#))

[SWS_Dcm_CONSTR_6091] Authentication configuration [The presence of a DcmDsdService with DcmDsdSidTabServiceId set to 0x29, requires a configured container DcmDspAuthentication on DcmDsp.] ()

[SWS_Dcm_CONSTR_6092] Authentication per connection [The presence of a DcmDsdService with DcmDsdSidTabServiceId set to 0x29, requires a configured DcmDspAuthenticationConnection per configured connection DcmDslConnection.] ()

[SWS_Dcm_CONSTR_6093] One authentication configuration per connection [Each DcmDspAuthenticationConnection shall refer a different DcmDslMainConnection by the reference in DcmDspAuthenticationConnectionMainConnectionRef.] ()

The Dcm is using the Csm and KeyM for certificate management. Parsing the certificate is a potential time-consuming operation and needs to be executed asynchronous. There are opinions that security shall not reveal any cause of failed authentication and thus have no dedicated NRCs for different certificate failures. To respect this use case the Dcm provides a general security NRC handling.

[SWS_Dcm_01560] Use of generic NRC for invalid certificate or content [If the mode referenced by DcmDspAuthenticationGeneralNRCModeRuleRef evaluates to true, the Dcm shall send the NRC code given in DcmDspAuthenticationGeneralNRC instead of the specific NRC codes in all situation where a Certificate verification failed - NRC is returned.] ([RS_Diag_04230](#))

[SWS_Dcm_CONSTR_6094] Generic NRC configuration requirements [If `DcmDspAuthenticationGeneralNRCModeRuleRef` is configured the parameter `DcmDspAuthenticationGeneralNRC` shall also be configured.] ([RS_Diag_04230](#))

7.5.2.10.1 De-authentication

The de-authenticate sub-function shall be always available if service 0x29 is used. This service set the authentication state to deauthenticated for the diagnostic connection the request was received on.

[SWS_Dcm_CONSTR_6095] [The presence of a `DcmDsdService` with `DcmDsdSidTabServiceId` set to 0x29, requires a `DcmDsdSubService` on this `DcmDsdService` with `DcmDsdSubServiceId` set to `deAuthenticate`.] ()

[SWS_Dcm_01561] Deauthentication by diagnostic service request [On reception of an Authentication (0x29) service with sub-function equal to de-authenticate, the `Dcm` shall set the authentication state to deauthenticated on the connection the `Dcm` has received that request.] ([RS_Diag_04230](#))

[SWS_Dcm_01565] [On reception of an Authentication (0x29) service with sub-function equal to de-Authenticate, the `Dcm` shall reply with a positive response having the `authenticationReturnParameter` set to `DeAuthentication successful`.] ([RS_Diag_04230](#))

7.5.2.10.2 Verify Certificates

[SWS_Dcm_01459] Supported communicationConfiguration [On reception of an Authentication (0x29) service with sub-function equal to `verifyCertificateUnidirectional` or `verifyCertificateBidirectional`, the `Dcm` shall reply with an `NRC` 0x31 (`requestOutOfRange`) if the `communicationConfiguration` (COCO) parameter has a value different than 0x00.] ([RS_Diag_04230](#))

The support of a COCO with constant value of 0x00 implies that no session key is supported by the `Dcm`. Upon receiving an authentication request with sub-function `verifyCertificateUniDirectional` and the `communicationConfiguration` (COCO) set to 0x00, the `Dcm` starts to verify the certificate. This service is implemented on BSW level by intention, to reduce the possible dialects of service 0x29 to what is natively supported by AUTOSAR. The `Csm` [15] and `KeyM` [16] modules allow to use different cryptographic methods. It is part of the `Dcm` strategy to explicitly require that certificates and all kind of access to information inside are handled by the `Csm` and its configuration. This allows to map different kind of certificates with different levels of security to the `Dcm`, thus abstracting the certificate complexity from `Dcm`.

[SWS_Dcm_01460] Verify verifyCertificateUnidirectional message length check [On reception of an Authentication (0x29) service with sub-function equal to `verifyCertificateUnidirectional`, the `Dcm` shall verify that the message length fits to the size given

in the parameters `lengthOfCertificateClient` and return a `NRC 0x13` if the size does not match. [\(RS_Diag_04230\)](#)

[SWS_Dcm_01461] Verify verifyCertificateBidirectional message length check [On reception of an Authentication (0x29) service with sub-function equal to `verifyCertificateBidirectional`, the `Dcm` shall verify that the message length fits to the size given in the parameters `lengthOfCertificateClient` and `lengthOfChallengeClient` and return a `NRC 0x13` if the size does not match. The `Dcm` uses the `lengthOfCertificateClient` as offset to calculate the position of `lengthOfChallengeClient`.] [\(RS_Diag_04230\)](#)

[SWS_Dcm_01462] Required configuration for bidirectional authentication [If the sub-function `verifyCertificateBidirectional` is configured in the `DsdServiceSubFunction` for 0x29, the configuration parameters `DcmDspAuthenticationECUCertificateRef` and `DcmDspAuthenticationECUCertificateKeyElementRef` are required.] [\(\)](#)

[SWS_Dcm_01579] Mandatory certificate data [On reception of an Authentication (0x29) service with sub-function equal to `verifyCertificateUnidirectional` or `verifyCertificateBidirectional` and a `lengthOfCertificateClient` equal to zero, the `Dcm` shall respond with `NRC 0x13` (`incorrectMessageLengthOrInvalidFormat`).] [\(RS_Diag_04230\)](#)

Store certificate in Csm

[SWS_Dcm_01463] Certificate validation [On reception of an Authentication (0x29) service with sub-function equal to `verifyCertificateUnidirectional` or `verifyCertificateBidirectional`, the `Dcm` shall use the KeyM API `KeyM_SetCertificate` to store the client certificate from the service request within the KeyM module. The following parameter values shall be used:

- `CertificateId` = `DcmDspAuthenticationConnectionCertificateRef` -> `KeyMCertificate.KeyMCertificateId`
- `certificateDataPtr.certData` = Pointer to `certificateClient` from Request
- `certificateDataPtr.certDataLength` = `lengthOfCertificateClient` from Request

[\(RS_Diag_04230\)](#)

Diagnostic certificate configuration is a task that is mainly executed in the `Csm` and `KeyM` modules. The `Dcm` provides an abstraction from these modules and only keeps symbolic references to containers that keep the information. Please take attention while configuring the `Csm` and `KeyM` 'in spirit of `Dcm` certificates'.

[SWS_Dcm_01464] Invalid certificate size failure [If the API `KeyM_SetCertificate` returns `KEYM_E_SIZE_MISMATCH`, the `Dcm` shall return the `NRC 0x13` (`incorrectMessageLengthOrInvalidFormat`).] [\(RS_Diag_04230\)](#)

[SWS_Dcm_01465] Behavior on busy crypto operation [If any of the `Csm` or `KeyM` APIs called by the `Dcm` is returning `CRYPTO_E_BUSY` or `KEYM_E_BUSY`, the `Dcm` shall return the `NRC 0x21` (`busyRepeat`).] [\(RS_Diag_04230\)](#)

[SWS_Dcm_01466] Csm APIs returning failure code behavior [Throughout this chapter the Csm or KeyM are used to process the authentication requests. These APIs have return values for failures. Unless there is no dedicated requirement for a given return value and if the return value is different to E_OK, the Dcm shall return a NRC 0x10 'GeneralReject'.] ([RS_Diag_04230](#))

The cryptographic operations have potential long execution times and are called asynchronously.

[SWS_Dcm_01467] Asynchronous certificate operation handling [For all asynchronous Csm or KeyM operations, the Dcm shall wait until the callback has been called, indicating a successful job termination. If the result (e.g. KeyM_ResultType) is E_OK, the Dcm shall continue to process the 0x29 request, if the result is different to E_OK, the Dcm shall skip the further processing and return a negative response with NRC 'GeneralReject'.] ([RS_Diag_04230](#))

Parse and Verify certificate in Csm

[SWS_Dcm_01468] Verifying client certificate [After the Dcm has set the certificate according to [\[SWS_Dcm_01463\]](#), the Dcm shall verify the certificate by calling KeyM_VerifyCertificate with the following parameters:

- CertificateId = DcmDspAuthenticationConnectionCertificateRef -> KeyMMCertificate.KeyMCertificateId

] ([RS_Diag_04230](#))

[SWS_Dcm_01469] Behavior on failed certificate verification [After the Dcm has verified a certificate and KeyM_ReturnType is set to KEYM_E_SIGNATURE_FAIL, the Dcm shall send a negative response with NRC set to 'Certificate verification failed - Invalid Signature'.] ([RS_Diag_04230](#))

[SWS_Dcm_01470] Check certificate chain of trust [If the operation started in [\[SWS_Dcm_01468\]](#) returns a result of KEYM_E_INVALID_CHAIN_OF_TRUST, the Dcm shall refuse the client certificate and return a negative response with NRC 'Certificate verification failed - Invalid Chain of Trust'.] ([RS_Diag_04235](#))

[SWS_Dcm_01471] Check certificate type [If the operation started in [\[SWS_Dcm_01468\]](#) returns a result of KEYM_E_INVALID_TYPE, the Dcm shall refuse the client certificate and return a negative response with NRC 'Certificate verification failed - Invalid Type'.] ([RS_Diag_04235](#))

[SWS_Dcm_01472] Check certificate format [If the operation started in [\[SWS_Dcm_01468\]](#) returns a result of KEYM_E_INVALID_FORMAT, the Dcm shall refuse the client certificate and return a negative response with NRC 'Certificate verification failed - Invalid Format'.] ([RS_Diag_04235](#))

[SWS_Dcm_01473] Check certificate format [If the operation started in [\[SWS_Dcm_01468\]](#) returns a result of KEYM_E_INVALID_CONTENT, the Dcm shall refuse the client certificate and return a negative response with NRC 'Certificate verification failed - Invalid Content'.] ([RS_Diag_04235](#))

[SWS_Dcm_01563] Check certificate format [If the operation started in [SWS_Dcm_01468] returns a result of KEYM_E_INVALID_SCOPE, the Dcm shall refuse the client certificate and return a negative response with NRC 'Certificate verification failed - Invalid Scope'.] (RS_Diag_04235)

[SWS_Dcm_01475] Check certificate format [If the operation started in [SWS_Dcm_01468] returns a result of KEYM_E_CERTIFICATE_REVOKED, the Dcm shall refuse the client certificate and return a negative response with NRC 'Certificate verification failed - Invalid Certificate (revoked)'.] (RS_Diag_04235)

[SWS_Dcm_01476] Check certificate valid until [If the operation started in [SWS_Dcm_01468] returns a result of KEYM_E_VALIDITY_PERIOD_FAIL, the Dcm shall refuse the client certificate and return a negative response with NRC 'Certificate verification failed - Invalid Time Period'.] (RS_Diag_04235)

Verify target identification

Certificates can have target constraints, such as limiting a certificate to a certain VIN or ECU version number. This is realized by target identification blocks as elements of the certificate. A target identification block is evaluated at runtime, only if all items are fulfilled, the certificate is accepted by the Dcm. Each target identification element is mapped to a compare value of a DcmModeCondition. A DcmModeRule combines these conditions to one term that evaluates to true or false.

[SWS_Dcm_01490] Certificate acceptance by target identification [The Dcm shall only accept certificates if the configured mode rule referenced by DcmDspAuthenticationTargetIdentificationModeRuleRef is evaluated to true.] (RS_Diag_04236)

[SWS_Dcm_01491] Behavior on failed target identification [If the configured mode rule referenced by DcmDspAuthenticationTargetIdentificationModeRuleRef is evaluated to fail, the Dcm shall refuse the certificate and send a negative response with NRC 'Certificate verification failed - Invalid Scope'.] (RS_Diag_04236)

[SWS_Dcm_01492] Invalid target identification block data [If the Dcm fails to match the target identification block data from DcmModeConditionConnectionCertificateCompareElementRef as valid (e.g. number of required bytes do not match), the Dcm shall send a negative response with NRC 'Certificate verification failed - Invalid Format'.] (RS_Diag_04236)

Providing the server challenge

[SWS_Dcm_01493] Creating the server challenge [After successfully validating the client certificate in [SWS_Dcm_01463], the Dcm shall create a server challenge by using the Csm in the following sequence:

- 1) Call of Csm_RandomGenerate with parameters
 - a. jobld : DcmDspAuthenticationRandomJobRef -> CsmJobld
 - 2) Wait until the asynchronous job has terminated according to [SWS_Dcm_01467].
-] (RS_Diag_04230)

The API `Csm_RandomGenerate` requires an initialised random generator. The initialization of the random generator is a task of the system (integrator).

[SWS_Dcm_01503] Sending positive response on `verifyCertificateUniDirectional`

[If the `Dcm` has successfully calculated server challenge the `Dcm` shall send a positive response for the `verifyCertificateUniDirectional` request with the following parameter values:

`authenticationReturnParameter` : 'Certificate verified, Ownership verification necessary'

`lengthOfChallengeServer`: length of the challenge generated in [SWS_Dcm_01493]

`challengeServer`: challenge generated in [SWS_Dcm_01493]

`lengthOfEphemeralPublicKeyServer`: 0x0000

](RS_Diag_04230)

Processing a diagnostic request `verifyCertificateBidirectional` will furthermore require the `Dcm` to send a server certificate and signing the client challenge. Therefore, the following steps are done additionally for `verifyCertificateBidirectional`.

[SWS_Dcm_01504] Signing client challenge [On reception of an Authentication (0x29) service with sub-function equal to `verifyCertificateBidirectional`, the `Dcm` shall sign the received client challenge by calling `Csm_SignatureGenerate` with the following parameter values:

`jobId`: `DcmDspAuthenticationClientChallengeSignJobRef` -> `CsmJobId`

`mode`: `CRYPTO_OPERATIONMODE_SINGLECALL`

`dataPtr`: Pointer to `challengeClient` in request

`dataLength`: `lengthOfChallengeClient` from request

`resultPtr`: `Dcm` sided buffer to store the `proofOfOwnershipServer`

`resultLengthPtr`: Response data for `lengthOfProofOfOwnershipServer`

](RS_Diag_04230)

[SWS_Dcm_CONSTR_6096] Require asynchronous client challenge signing [`DcmDspAuthenticationClientChallengeSignJobRef` shall be only accepted if the referenced `CsmJob` itself:

- has a `CsmProcessingMode` set to `CRYPTO_PROCESSING_ASYNC`
- references a `CsmPrimitive` with an aggregated `CsmSignatureGenerate`.

](RS_Diag_04230)

[SWS_Dcm_01506] Providing the server certificate [On reception of an Authentication (0x29) service with sub-function equal to `verifyCertificateBidirectional`, the `Dcm` shall provide the server certificate in the response. The `Dcm` shall call `KeyM_GetCertificate` with the following parameters:

`certId`: `DcmDspAuthenticationECUCertificateRef/KeyMCertificateId`

`certificateDataPtr`: `Dcm` implementation specific

](RS_Diag_04230)

[SWS_Dcm_01507] Server certificate size check [If the API Csm_GetKeyElement returns CRYPTO_E_SMALL_BUFFER, the Dcm shall return NRC 0x14 (responseTooLong).] ([RS_Diag_04230](#))

[SWS_Dcm_01508] Sending positive response on verifyCertificateBidirectional [If the Dcm has successfully calculated server challenge, the server challenge and the server certificate, the Dcm shall send a positive response for the verifyCertificateBidirectional request with the following parameter values:
authenticationReturnParameter: 'Certificate verified, Ownership verification necessary'
lengthOfChallengeServer: length of the challenge generated in [\[SWS_Dcm_01493\]](#)
challengeServer: challenge generated in [\[SWS_Dcm_01493\]](#)
lengthOfCertificateServer: length of the certificated provided in [\[SWS_Dcm_01506\]](#)
certificateServer: certificated provided in [\[SWS_Dcm_01506\]](#)
lengthOfProofOfOwnershipServer: length of proof-of-ownership server created in [\[SWS_Dcm_01504\]](#)
proofOfOwnershipServer: proof-of-ownership server created in [\[SWS_Dcm_01504\]](#)
lengthOfEphemeralPublicKeyServer: 0x0000
] ([RS_Diag_04230](#))

7.5.2.10.3 Proof of ownership client

[SWS_Dcm_01509] Sequence check [On reception of an Authentication (0x29) service with sub-function equal to proofOfOwnership and if on this connection the most recent processed verifyCertificateUnidirectional or verifyCertificateBidirectional service failed or no such sub-function was processed yet, the Dcm shall send the negative response 'requestSequenceError'. The connection shall remain in de-authenticated state.] ([RS_Diag_04230](#))

[SWS_Dcm_01510] Proof of ownership message length check [On reception of an Authentication (0x29) service with sub-function equal to proofOfOwnership, the Dcm shall verify that the message length fits to the size given in the parameters lengthOfProofOfOwnershipClient and lengthOfEphemeralPublicKeyClient and return a NRC 0x13 if the size does not match.] ([RS_Diag_04230](#))

[SWS_Dcm_01511] Client proof of ownership verification [On reception of an Authentication (0x29) service with sub-function equal to proofOfOwnership, the Dcm shall verify the client's proof of ownership data provide in the request by calling Csm_SignatureVerify with the following in-parameters:

jobId: [DcmDspAuthenticationVerifyProofOfOwnershipClientJobRef](#) -> CsmJob.CsmJobId
mode: set to CRYPTO_OPERATIONMODE_SINGLECALL
dataPtr: challenge generated [\[SWS_Dcm_01493\]](#)
dataLength: length of the challenge generated in [\[SWS_Dcm_01493\]](#)
signaturePtr: proofOfOwnershipClient from request

signatureLength: lengthOfProofOfOwnershipClient from request
](RS_Diag_04230)

[SWS_Dcm_CONSTR_6098] Require asynchronous signature verification [DcmDspAuthenticationVerifyProofOfOwnershipClientJobRef is only valid if the referenced CsmJob itself:

- has a CsmProcessingMode set to CRYPTO_PROCESSING_ASYNC
- references a CsmPrimitive with an aggregated CsmSignatureVerify.

]()

[SWS_Dcm_01512] Failed ownership verification [If the result of Csm_SignatureVerify from [SWS_Dcm_01511] is Crypto_VerifyResultType equal to CRYPTO_E_VER_NOT_OK, the Dcm shall send a negative response with NRC 'Ownership verification failed'.](RS_Diag_04230)

[SWS_Dcm_01513] SessionKey support [Upon sending a positive response for an authentication request with sub-function equal to proofOfOwnership, the Dcm shall:

- set all bytes of lengthOfSessionKeyInfo to 0x00
- omit the sessionKeyInfo

](RS_Diag_04230)

Set current role

[SWS_Dcm_01514] Update the current role [If the proof of ownership verification in [SWS_Dcm_01511] was successful and resulted in CRYPTO_E_VER_OK, the Dcm shall use the value read from the certificate extension DcmDspAuthenticationRoleElementRef as new role for the current authenticated state.](RS_Diag_04233)

[SWS_Dcm_01515] Role size check [If the size of the read role information in [SWS_Dcm_01514] is different than the size in DcmDspAuthenticationRoleSize, the Dcm shall send a negative response with NRC 'Certificate verification failed - Invalid Content'.](RS_Diag_04230)

Set white list

[SWS_Dcm_01516] Update the current whitelist [If the proof of ownership verification in [SWS_Dcm_01511] was successful and resulted in CRYPTO_E_VER_OK, the Dcm shall use the white list read from the certificate according to [SWS_Dcm_01517] as new white list for the current authenticated state.](RS_Diag_04232)

[SWS_Dcm_01517] White list access [To read the white list from a certificate, the Dcm shall read all child elements from the referenced DcmDspAuthenticationWhiteListServicesElementRef, DcmDspAuthenticationWhiteListDIDElementRef, DcmDspAuthenticationWhiteListRIDElementRef and DcmDspAuthenticationWhiteListMemorySelectionElementRef certificate data, by repeating calling the sequence of KeyM_CertElementGetFirst and KeyM_CertElementGetNext until no further child element is available. The Dcm shall

use the following in parameter values:

```
certId: DcmDspAuthenticationConnectionCertificateRef -> KeyMCertificate.KeyMCertificateId  
certElementId: DcmDspAuthenticationWhiteListMemorySelectionElementRef -> KeyMCertificateElement.KeyMCertificateElementId  
|()
```

[SWS_Dcm_01518] White list size check [If the size of the read white list information in [SWS_Dcm_01516] is larger than the size in `DcmDspAuthenticationWhiteListMemorySelectionMaxSize`, the `Dcm` shall send a negative response with NRC 'Certificate verification failed - Invalid Content'.] ([RS_Diag_04230](#))

[SWS_Dcm_CONSTR_6087] Required size for white lists [If any of the optional `DcmDspAuthenticationWhiteListMemorySelectionElementRef` are configured, the corresponding `DcmDspAuthenticationWhiteListMemorySelectionMaxSize` shall be configured for that white list.] ([RS_Diag_04232](#))

7.5.2.10.4 Definition and verification of roles

The roles transmitted inside the certificates are used to assign a tester on one connection one or more roles. A single role itself is presented by a certain bit within the bitfield representation of the role. Within the `Dcm` there is static role assignment to each diagnostic service. A service can be executed if at least one role (bit) assigned to a service matches a role (bit) in the certificate.

[SWS_Dcm_CONSTR_6088] Supported role sizes [The parameter `DcmDspAuthenticationRoleSize` defines the size in bytes used in both, certificates and ECU internal static role configuration. All role parameters (e.g. `DcmDspServiceRole`) shall have values that would fit in the amount of bytes given by `DcmDspAuthenticationRoleSize`.] ([RS_Diag_04233](#))

[SWS_Dcm_01521] Integer interpretation of roles in certificates [The `Dcm` shall interpret all configured role integer values in the little endian format.] ([RS_Diag_04234](#))

[SWS_Dcm_01522] Role verification [Upon each role verification, the `Dcm` shall make a bitwise 'and' operation on the role provided in the certificate for that connection and the role assigned to the service. If the result is greater than 0, the `Dcm` shall treat the service as allowed to be executed.] ([RS_Diag_04233](#); [RS_Diag_04234](#))

[SWS_Dcm_01523] Failed role verification [Upon each role verification, the `Dcm` shall make a bitwise 'and' operation on the role provided in the certificate for that connection and the role assigned to the service. If the result is equal to 0, the `Dcm` shall stop the processing of that service and send a negative response 'authenticationRequired'.] ([RS_Diag_04233](#); [RS_Diag_04234](#))

An example of roles in certificates and services with assigned certificates is given in Figure 7.10. The provided certificate uses 1 byte for roles, with 5 role definitions in. The certificate will grant the tester rights for the roles 'AfterSales' and 'ExtendedUser'.

With that certificate, the tester can execute the services that have the corresponding permissions to be executed in that roles. In that example, the service 0x28 and 0x11 01 are both allowed to be executed. The routine with identifier 0x5678 is only allowed to be executed in role 'production', the *Dcm* will deny the execution with *NRC* 'authenticationRequired'.

	Development	Production	AfterSales	ExtendedUser	ePT	unused	unused	unused
Bit	7	6	5	4	3	2	1	0
Certificate	0	0	1	1	0	0	0	0
Service List								
Service 0x28	0	0	1	1	0	0	0	0
Service 0x11 01	0	0	1	0	0	0	0	0
RID 0x5678	0	1	0	0	0	0	0	0

Figure 7.10: Example for roles

7.5.2.10.5 Definition and verification of white lists

Besides the use of roles, the certificates can get extended with optional white lists for service execution. This allows the issuer of the certificate to extend the range of allowed services without updating the access lists in the ECU.

The white list is build out of the following elements:

- List of allowed services, 1-4 byte each
- List of allowed data identifiers (*DID*) and access information, 3 byte each
- List of allowed routine identifiers (*RID*) and access information, 3 byte each
- List of allowed user defined fault memories, 1 byte each

Parsing and access to the different elements of the white lists is done by invoking *KeyM_CertElementGetFirst* and *KeyM_CertElementGetNext* according to [*SWS_Dcm_01517*].

[SWS_Dcm_01524] White list for services definition [The white list for services is a set of values, each consisting of up to 4 bytes. Each value itself contains the first bytes of a diagnostic service that is allowed be executed.](*RS_Diag_04233*; *RS_Diag_04234*)

Example:

In the following example, a white list section within a certificate is shown. It contains 5 additional services that can be executed:

```

1 ...
2 SEQUENCE (1 elem)
3   SEQUENCE (2 elem)
4     OBJECT IDENTIFIER 3.9.3.345.3.1.3453.24.3.9.355.73
5       SET (6 elem)
6         OCTET STRING (1 byte) 85
7         OCTET STRING (2 byte) 1101
8         OCTET STRING (3 byte) 22123A
9         OCTET STRING (3 byte) 2E123A
10        OCTET STRING (4 byte) 310113F4
11 SEQUENCE (3 elem) ....
  
```

This will allow the Dcm to execute:

- Service 0x85 (with any sub-subfunction and DTCSettingControlOptionRecord byte)
- Service 0x11 01
- Read and WriteDataByIdentifier for [DID 0x123A](#)
- Routine Start for [RID 0x13F4](#)

Checks for white lists for services are all done at [DSD](#) level. The [Dcm](#) checks if the first bytes of a received diagnostic request match the values allowed by the white list. If a white list entry exists where all bytes are matching the first bytes of the request, the service is granted access.

For all other white lists, the [Dcm](#) performs the checks at [DSP](#) level in the corresponding service processors.

Please note that it is possible to allow a [DID](#) execution in two places in the white list: 1) as 3 byte service and 2) as [DID](#) in the [DID](#) list. The difference is that the service checks only the first 3 bytes of the [PDU](#) and will not detect the [DID](#) being used in multiple [DID](#) or dynamically defined [DID](#) requests, while the [DID](#) list element is verified in all services requesting the [DID](#).

[SWS_Dcm_01525] White list definition for [DIDs](#) [The white list for [DIDs](#) defines the set of data identifiers that are allowed to be read, written and controlled. Each entry shall contain in the first two bytes the [DID](#) number and in the third byte the following access definitions:

- Bit 0: Read access
- Bit1: Write access
- Bit2: Control access (by InputOutputControlByIdentifier)

A bit value of 0 means that the operation is forbidden, a bit value of 1 means that the operation is allowed. All not used bits (3-7) shall be set to zero.

[DID](#) numbers are always in big endian format (MSB first).] ([RS_Diag_04233](#); [RS_Diag_04234](#))

Example:

DID access record	Grants access to
0x1A90 0x01	All read data by identifier operations for DID 0x1A90
0x314F 0x05	All read data by identifier and IOcontrol operations for DID 0x314F

DID read operations are performed from various UDS services. Dcm checks on each DID read the access to that DID.

[SWS_Dcm_01526] White list definition for RIDs [The white list for RIDs defines the set of routine identifiers that have access to the sub-functions startRoutine, stopRoutine and requestRoutineResult. Each entry shall contain in the first two bytes the RID number and in the third byte the following access definitions:

- Bit 0: Access to sub-function 'startRoutine'
- Bit1: Access to sub-function 'stopRoutine'
- Bit2: Access to sub-function 'requestRoutineResult'

A bit value of 0 means that the sub-function is forbidden, a bit value of 1 means that the sub-function is allowed. All not used bits (3-7) shall be set to zero.

RID numbers are always in big endian format (MSB first).] ([RS_Diag_04233](#); [RS_Diag_04234](#))

Example:

RID access record	Grants access to
0x0240 0x01	StartRoutine is allowed for RID 0x0240
0x028A 0x07	All routine sub-functions are allowed for RID 0x28A

[SWS_Dcm_01527] White list definition for MemorySelection [The white list memory selection is used for the UDS service 0x19 with sub-functions 0x17, 0x18 and 0x19. The value in the white list defines the accepted parameter values for MemorySelection in the UDS request.] ([RS_Diag_04233](#); [RS_Diag_04234](#))

Transition into authenticated session

[SWS_Dcm_01528] Transition into authenticated state [If the proof of ownership verification in [\[SWS_Dcm_01511\]](#) was successful and resulted in CRYPTO_E_VER_OK and after the role and white list setting was done, the Dcm shall set the DcmAuthenticationState_{Channel} on that connection into DCM_AUTHENTICATED.] ([RS_Diag_04230](#))

[SWS_Dcm_01529] Successful ownership verification [If the result of Csm_SignatureVerify from [\[SWS_Dcm_01511\]](#) is Crypto_VerifyResultType equal

to CRYPTO_E_VER_OK, the `Dcm` shall send a positive response with authentication-ReturnParameter set to 'Ownership verified, Authentication complete' and providing a size of 0 for lengthOfSessionKeyInfo and no sessionKeyInfo.](RS_Diag_04230)

[SWS_Dcm_01530] Persisting authentication state [If the result of Csm_SignatureVerify from [SWS_Dcm_01511] is Crypto_VerifyResultType equal to CRYPTO_E_VER_OK and the mode rule referenced by `DcmDspAuthenticationPersistStateModeRuleRef` is evaluated to true, the `Dcm` shall persist the authentication state, role and white list on that connection.](RS_Diag_04230)

[SWS_Dcm_01531] Activation of role and white list [The `DCM` shall consider the role and white list for access control only, if the `DCM` is in authenticated state.]()

[SWS_Dcm_01532] Re-entering authenticated state [If the `Dcm` is already in authenticated state while a transition to authentication stated is requested according to [SWS_Dcm_01529], the `Dcm` shall overwrite the roles and white list and use only the role and white last from the last received certificate.](RS_Diag_04230)

7.5.2.10.6 AuthenticationConfiguration

[SWS_Dcm_00852] [If the configuration parameter `DcmSupportDTCSettingControlOptionRecord` is set to false and the request contains any data after the subfunction, the `Dcm` shall return NRC 0x13 (Incorrect message length or invalid format).]()

[SWS_Dcm_01533] Providing the authentication configuration [If `DcmDspAuthentication` is configured and an Authentication (0x29) service with subfunction equal to authenticationConfiguration is received, the `Dcm` shall send a positive response with authenticationReturnParameter set to 'AuthenticationConfiguration APCE'.](RS_Diag_04230)

7.5.2.11 Service 0x2A - ReadDataByPeriodicIdentifier

[SWS_Dcm_01093] Minimum length check for ReadDataByPeriodicIdentifier [On reception of the UDS Service ReadDataByPeriodicIdentifier (0x2A), the `Dcm` module shall check the request minimum length. If length of the request is wrong, the `Dcm` module shall send a NRC 0x13 (Incorrect message length or invalid format).](RS_Diag_04215)

[SWS_Dcm_01552] Read Periodic DID authentication check for statically defined DIDs [On reception of the UDS Service ReadDataByPeriodicIdentifier (0x2A), the `Dcm` shall check if the access to all static defined requested DIDs is authenticated and read the data identifiers periodically only if:

- a `DcmDspDidReadRole` is configured for that DID and the verification according to [SWS_Dcm_01522] was successful or

- the active white list on that connection has for each requested DID one entry with read access that matches that DID.

](RS_Diag_04233)

[SWS_Dcm_01553] Read Periodic DID authentication check for dynamically defined DIDs [On reception of the UDS Service ReadDataByPeriodicIdentifier (0x2A), the Dcm shall check if the access to all dynamically defined requested DIDs if the content is based of DIDs or parts of DIDs is authenticated and read the data identifiers periodically only if:

- a DcmDspDidReadRole is configured for that DID and the verification according to [SWS_Dcm_01522] was successful or
- the active white list on that connection has for each requested DID one entry with read access that matches that DID.

](RS_Diag_04233)

According to [SWS_Dcm_01537] the authentication checks are only executed if DcmDspAuthentication is configured. In case of a failed authentication the NRC handling is according to [SWS_Dcm_01544] and [SWS_Dcm_01551] applies.

[SWS_Dcm_00721] Session check for ReadDataByPeriodicIdentifier DID [On reception of the UDS Service ReadDataByPeriodicIdentifier (0x2A), for every requested periodicDIDs, the Dcm module shall check if the periodicDID can be read in the current session (see configuration parameter DcmDspDidReadSessionRef). If none of the periodicDID can be read in the current session, the Dcm module shall send a NRC 0x31 (RequestOutOfRange).](RS_Diag_04215)

[SWS_Dcm_00722] Security level check for ReadDataByPeriodicIdentifier DID [On reception of the UDS Service ReadDataByPeriodicIdentifier (0x2A), for every requested periodicDIDs, the Dcm module shall check if the periodicDID can be read in the current security level (see configuration parameter DcmDspDidReadSecurityLevelRef). If not, the Dcm module shall send NRC 0x33 (Security access denied).](RS_Diag_04215)

[SWS_Dcm_00820] Mode condition check for ReadDataByPeriodicIdentifier DID [On reception of the UDS Service ReadDataByPeriodicIdentifier (0x2A), for every requested periodicDIDs, the Dcm module shall check if the periodicDID can be read in the current mode condition (see configuration parameter DcmDspDidReadModeRuleRef). If not, the Dcm module shall send the calculated negative response code of the reference DcmModeRule](RS_Diag_04215)

[SWS_Dcm_01097] Checks for DDDIDs in ReadDataByPeriodicIdentifier [On reception of the UDS Service ReadDataByPeriodicIdentifier (0x2A), if verification has been successfully done ([SWS_Dcm_00721], [SWS_Dcm_00722] and [SWS_Dcm_00820]), and if the request contains one or more dynamically defined DID(s), the Dcm module shall do the session, security and mode dependencies checks for all source data in case the configuration parameter DcmDspDDDIDcheckPerSourceDID is set to TRUE.](RS_Diag_04215)

[SWS_Dcm_01098] Condition check for ReadDataByPeriodicIdentifier DID [On reception of the UDS Service ReadDataByPeriodicIdentifier (0x2A), for every requested periodicDIDs, the Dcm module shall invoke the ConditionCheckRead operation (or the respective C-Function) if configured. In case of a negative result, the returned Error-Code shall be used as final negative response code.](RS_Diag_04215)

[SWS_Dcm_01099] Support of dynamic length DIDs [On reception of the UDS Service ReadDataByPeriodicIdentifier (0x2A), for every requested periodicDIDs, with a configured dynamic length the Dcm module shall invoke the ReadDataLength operation (or the respective C-Function) to retrieve the length of the periodicDID. This length is valid for each ReadData operation till the periodicDID is removed from the scheduler or updated via a new request. This length shall further be used to check against the UUDT size.](RS_Diag_04215)

[SWS_Dcm_01100] Verifications on all sub-functions except stopSending [On reception of the UDS Service ReadDataByPeriodicIdentifier (0x2A) with transmission-Mode different than stopSending, the Dcm shall do the verification for session, security and mode rule.](RS_Diag_04215)

[SWS_Dcm_01426] No verifications for sub-function stopSending [On reception of the UDS Service ReadDataByPeriodicIdentifier (0x2A) with transmissionMode = stopSending, the Dcm shall skip the verification for security and mode rule.](RS_Diag_04215)

[SWS_Dcm_01427] Dcm behavior on sub-function stopSending without further parameters [On reception of the UDS Service ReadDataByPeriodicIdentifier (0x2A) with transmissionMode = stopSending and no periodicDataIdentifier in the request, the Dcm shall stop all scheduled periodicDataIdentifier transmissions.](RS_Diag_04215)

[SWS_Dcm_01428] Dcm behavior on sub-function stopSending with further parameters [On reception of the UDS Service ReadDataByPeriodicIdentifier (0x2A) with transmissionMode = stopSending and at least one periodicDataIdentifier is in the request, the Dcm shall stop the scheduled periodic data transmissions for all requested and in the current session supported periodicDataIdentifiers.](RS_Diag_04215)

[SWS_Dcm_00716] Serialisation with S/R data interfaces [To serialize the required AUTOSAR data types (signed- and unsigned integer) into the response message of ReadDataByPeriodicIdentifier responses the target endianness configured in DcmDspDataEndianness shall be considered for DcmDspData elements having DcmDspDataUsePort set to USE_DATA_SENDER_RECEIVER, USE_DATA_SENDER_RECEIVER_AS_SERVICE, USE_ECU_SIGNAL. In case DcmDspDataEndianness is not present, the DcmDspDataDefaultEndianness shall be used instead.](RS_Diag_04215)

[SWS_Dcm_00843] Ensuring maxim number of periodic DIDs [On reception of the UDS Service ReadDataByPeriodicIdentifier (0x2A), the Dcm module shall check if the periodicDataIdentifiers requested in a single request do not exceed the configured DcmDspMaxPeriodicDidToRead (maximum length check). Otherwise (in case the

number of elements is exceeded) the `Dcm` module shall send a `NRC 0x13` (Incorrect message length or invalid format).] ([RS_Diag_04215](#))

[SWS_Dcm_01094] Behavior on not supported transmissionMode [On reception of the `UDS Service ReadDataByPeriodicIdentifier(0x2A)`, the `Dcm` module shall check if the `transmissionMode` is supported, otherwise the `Dcm` module shall send a `NRC 0x31`(Request out of range).] ([RS_Diag_04215](#))

Note: With `UDS Service ReadDataByPeriodicIdentifier (0x2A)`, the tester can start one or more periodicDIDs.

Note: A request of `UDS Service ReadDataByPeriodicIdentifier` will contain DIDs represented by only one byte, the low byte of the configured DID. In all cases the complete `DID` will be constructed by adding `0xF2` as high byte.

[SWS_Dcm_01095] Verification if periodic DID is supported [On reception of the `UDS Service ReadDataByPeriodicIdentifier (0x2A)`, for every requested periodicDIDs, the `Dcm` module shall check if the periodicDID is supported (see configuration parameter `DcmDspDid`). If none of the periodicDIDs are supported, the `Dcm` module shall send `NRC 0x31` (Request out of range).] ([RS_Diag_04215](#))

[SWS_Dcm_01096] Behavior on unused DIDs [If a `DID` is set as unused (`DcmDspDidUsed` set to `FALSE`), the `Dcm` shall consider the `DID` as not supported.] ([RS_Diag_04215](#))

[SWS_Dcm_00851] Verification for free scheduler slots [On reception of the `UDS Service ReadDataByPeriodicIdentifier(0x2A)` with `transmissionMode` different than `0x04 "stopSending"`, the `Dcm` module shall check if all requested periodicDataIdentifiers not currently in the periodic scheduler can be added to the scheduler considering the free space of the scheduler (maximum size is defined by configuration parameter `DcmDspMaxPeriodicDidScheduler`). In case the requested periodicDataIdentifiers cannot be added to the scheduler, the `Dcm` shall send a `NRC 0x31` (`RequestOutOfRange`).] ([RS_Diag_04215](#))

Note : To optimize the resource consumption AUTOSAR has chosen a simplified approach to validate the request message. ISO recommends checking the size only for currently supported data identifiers.

7.5.2.11.1 Scheduler PeriodicTransmission

Note: The periodic response message layout is according to ISO 14229-1:2013. It contains the periodic data identifier and the data. The service `ID` and `A_PCI` byte is not part of the periodic response message.

[SWS_Dcm_01101] [The `Dcm` shall send all periodic scheduled response by using the configured `DcmDslPeriodicConnections`.] ([RS_Diag_04215](#))

Note : Only UUDT messages (IF-PDUs) are supported

ISO 14229-1:2018 defines two distinct scheduler types for ReadDataByPeriodicIdentifier. Depending on the scheduler type, the `Dcm` has different transmission strategies. The used scheduler type of the server is selected by setting `DcmDspPeriodicTransmissionSchedulerType` to one of the two types.

[SWS_Dcm_01568] Support of periodic transmission type 1 [If `DcmDspPeriodicTransmissionSchedulerType` is set to `SCHEDULER_TYPE1` and the `Dcm` is set up to send PDIDs, the `Dcm` shall trigger the transmission of all scheduled PDIDs each time the configured periodic transmission rate elapses.] ([RS_Diag_04215](#))

[SWS_Dcm_01569] Transmission of one PID per available connection [If PID transmission is triggered according to [\[SWS_Dcm_01568\]](#), `Dcm` shall transmit one PID per available periodic connection starting with the first PID in the list of scheduled PDIDs.] ([RS_Diag_04215](#))

[SWS_Dcm_01570] Continue in next main function if more PDIDs than available connections [If PID transmission is triggered according to [\[SWS_Dcm_01568\]](#) and more PDIDs are to transmit than the number of available periodic connections, the `Dcm` shall continue the transmission in the next `Dcm` main function call. Each time the available periodic connections shall be used.] ([RS_Diag_04215](#))

[SWS_Dcm_01571] Overlapping triggers and transmissions [If PID transmission is triggered according to [\[SWS_Dcm_01568\]](#) or [\[SWS_Dcm_01572\]](#) and the next periodic transmission rate trigger occurs before the `Dcm` has finished sending the scheduled PDIDs according to [\[SWS_Dcm_01570\]](#), the `Dcm` shall continue to transmit PDIDs according to [\[SWS_Dcm_01570\]](#) and schedule the PDIDs of the new trigger after the last previous PID has been transmitted.] ([RS_Diag_04215](#))

[SWS_Dcm_01572] Support of periodic transmission type 2 [If `DcmDspPeriodicTransmissionSchedulerType` is set to `SCHEDULER_TYPE2` the `Dcm` is set up to send PDIDs and the periodic transmission rate for a scheduler elapses, the `Dcm` shall trigger the transmission of the next scheduled PDIDs on all available periodic connections.] ([RS_Diag_04215](#))

[SWS_Dcm_01573] PID transmission sequence for scheduler type 2 [For scheduler type 2 transmission of PDIDs within a single elapse of the scheduler, the `Dcm` shall start to transmit the first configured PID and continue to transmit consecutively all other configured PDIDs. If the last PID is transmitted, the `Dcm` restarts the sequence with the first configured PID. If a PID is sent more than once within a single elapse of the scheduler then the PDIDs contents shall be reloaded with the applicable `DID` / Memory data for every re-transmission.] ([RS_Diag_04215](#))

[SWS_Dcm_01574] Continue in next periodic rate scheduler if more PDIDs than available connections [If a scheduler type 2 PID transmission is active according to [\[SWS_Dcm_01572\]](#) and the number of scheduled PDIDs is larger than the number of available periodic connections, the `Dcm` shall continue to send the next scheduled PDIDs the next time the periodic transmission rate scheduler elapses.] ([RS_Diag_04215](#))

The following example gives an overview about scheduler type 1 and type 2. They use PDIDs starting at 0xF201 and having all PIDID data set to zero. The PDIDs are send on CAN 2.0, using the CAN-IDs 0x6A0, 0x6A1.

Example 1: Scheduler Type 1

```

1 SchedulerRate:           1000ms
2
3 PeriodicRate:           4000ms
4
5 NumPeriodicAddr:       1
6
7 NumPDID                 2
8
9 1017.02 6A0             01 00 00 00 00 00 00 00
10
11 2016.97 6A0            02 00 00 00 00 00 00 00
12
13
14
15 5016.87 6A0            01 00 00 00 00 00 00 00
16
17 6016.78 6A0            02 00 00 00 00 00 00 00
18
19
20
21 9016.67 6A0            01 00 00 00 00 00 00 00
22
23 10016.63 6A0           02 00 00 00 00 00 00 00
24
25
26
27 13016.52 6A0           01 00 00 00 00 00 00 00
28
29 14016.46 6A0           02 00 00 00 00 00 00 00
  
```

Example 2: Scheduler Type 2

```

1 SchedulerRate:           1000ms
2
3 PeriodicRate:           4000ms
4
5 NumPeriodicAddr:       1
6
7 NumPDID                 2
8
9  945.01 6A0             01 00 00 00 00 00 00 00
10
11 4944.78 6A0            02 00 00 00 00 00 00 00
12
13 8944.66 6A0            01 00 00 00 00 00 00 00
14
15 12944.49 6A0           02 00 00 00 00 00 00 00
16
17 16944.33 6A0           01 00 00 00 00 00 00 00
18
19 20944.28 6A0           02 00 00 00 00 00 00 00
  
```

```
20
21 24944.01 6A0 01 00 00 00 00 00 00 00
```

Example 3: Aliasing with Scheduler Type 1

```
1 SchedulerRate: 1000ms
2
3 PeriodicRate: 4000ms
4
5 NumPeriodicAddr: 1
6
7 NumPDID 5
8
9
10 529.89 6A0 01 00 00 00 00 00 00 00
11
12 1529.70 6A0 02 00 00 00 00 00 00 00
13
14 2529.59 6A0 03 00 00 00 00 00 00 00
15
16 3529.35 6A0 04 00 00 00 00 00 00 00
17
18
19
20 4529.16 6A0 05 00 00 00 00 00 00 00 <-- Alias (should start over at
    01)
21
22 5529.02 6A0 01 00 00 00 00 00 00 00
23
24 6529.15 6A0 02 00 00 00 00 00 00 00
25
26 7529.03 6A0 03 00 00 00 00 00 00 00
27
28
29
30 8528.92 6A0 04 00 00 00 00 00 00 00 <-- Alias (should start over
    at 01)
31
32 9528.82 6A0 05 00 00 00 00 00 00 00
33
34 10528.46 6A0 01 00 00 00 00 00 00 00
35
36 11528.60 6A0 02 00 00 00 00 00 00 00
37
38
39
40 12528.47 6A0 03 00 00 00 00 00 00 00 <-- Alias (should start over
    at 01)
41
42 13528.36 6A0 04 00 00 00 00 00 00 00
43
44 14528.27 6A0 05 00 00 00 00 00 00 00
```

Example 4: Scheduler Type 2

```
1 SchedulerRate: 1000ms
```

```

2
3 PeriodicRate:                4000ms
4
5 NumPeriodicAddr:            1
6
7 NumPDID                      5
8
9
10   592.03 6A0    01 00 00 00 00 00 00 00
11
12   4591.56 6A0   02 00 00 00 00 00 00 00
13
14   8591.13 6A0   03 00 00 00 00 00 00 00
15
16  12590.69 6A0  04 00 00 00 00 00 00 00
17
18  16590.26 6A0  05 00 00 00 00 00 00 00
19
20  20589.78 6A0   01 00 00 00 00 00 00 00
21
22  24589.23 6A0   02 00 00 00 00 00 00 00
23
24  28588.90 6A0   03 00 00 00 00 00 00 00
25
26  32588.45 6A0   04 00 00 00 00 00 00 00
27
28  36588.05 6A0   05 00 00 00 00 00 00 00
  
```

Example 5: Scheduler 1 with Multiple Periodic Addresses

```

1 SchedulerRate:                1000ms
2
3 PeriodicRate:                4000ms
4
5 NumPeriodicAddr:            2
6
7 NumPDID                      5
8
9
10
11   778.69 6A0    01 00 00 00 00 00 00 00
12
13   778.73 6A1    02 00 00 00 00 00 00 00
14
15
16
17  1778.56 6A0    03 00 00 00 00 00 00 00
18
19  1778.61 6A1    04 00 00 00 00 00 00 00
20
21
22
23  2778.44 6A0    05 00 00 00 00 00 00 00
24
25
26
  
```

```

27  4778.24 6A0    01 00 00 00 00 00 00 00
28
29  4778.27 6A1    02 00 00 00 00 00 00 00
30
31
32
33  5778.08 6A0    03 00 00 00 00 00 00 00
34
35  5778.12 6A1    04 00 00 00 00 00 00 00
36
37
38
39  6778.12 6A0    05 00 00 00 00 00 00 00
40
41
42  8777.90 6A0    01 00 00 00 00 00 00 00
43
44  8777.93 6A1    02 00 00 00 00 00 00 00
45
46
47
48  9777.70 6A0    03 00 00 00 00 00 00 00
49
50  9777.73 6A1    04 00 00 00 00 00 00 00
51
52
53
54 10777.57 6A0    05 00 00 00 00 00 00 00

```

Example 6: Scheduler 2 with Multiple Periodic Addresses

```

1  SchedulerRate:           1000ms
2
3  PeriodicRate:           4000ms
4
5  NumPeriodicAddr:       2
6
7  NumPDID                 5
8
9
10
11
12  764.64 6A0    01 00 00 00 00 00 00 00
13
14  764.69 6A1    02 00 00 00 00 00 00 00
15
16
17
18  4762.79 6A0    03 00 00 00 00 00 00 00
19
20  4762.82 6A1    04 00 00 00 00 00 00 00
21
22
23
24  8762.35 6A0    05 00 00 00 00 00 00 00
25

```

```

26  8762.36 6A1    01 00 00 00 00 00 00 00
27
28
29
30 12762.01 6A0    02 00 00 00 00 00 00 00
31
32 12762.04 6A1    03 00 00 00 00 00 00 00
  
```

Example 7: Scheduler 2 with 2 Periodic Addresses and 1 PDID

```

1 SchedulerRate:          1000ms
2
3 PeriodicRate:           1000ms
4
5 NumPeriodicAddr:       2
6
7 NumPDID                 1
8
9
10
11   7.19 6A0    01 00 00 00 00 00 00 00
12
13   77.21 6A1   01 00 00 00 00 00 00 00
14
15
16
17 1077.61 6A0    01 00 00 00 00 00 00 00
18
19 1077.68 6A1   01 00 00 00 00 00 00 00
20
21
22
23 2078.29 6A0    01 00 00 00 00 00 00 00
24
25 2078.40 6A1   01 00 00 00 00 00 00 00
26
27
28
29 3078.26 6A0    01 00 00 00 00 00 00 00
30
31 3078.33 6A1   01 00 00 00 00 00 00 00
32
33
34
35 4079.13 6A0    01 00 00 00 00 00 00 00
36
37 4079.19 6A1   01 00 00 00 00 00 00 00
38
39
40
41 5079.17 6A0    01 00 00 00 00 00 00 00
42
43 5079.24 6A1   01 00 00 00 00 00 00 00
44
45
46
  
```


47	6079.75	6A0	01	00	00	00	00	00	00	00
48										
49	6079.82	6A1	01	00	00	00	00	00	00	00
50										
51										
52										
53	7080.77	6A0	01	00	00	00	00	00	00	00
54										
55	7080.91	6A1	01	00	00	00	00	00	00	00
56										
57										
58										
59	8080.90	6A0	01	00	00	00	00	00	00	00
60										
61	8080.97	6A1	01	00	00	00	00	00	00	00
62										
63										
64										
65	9081.09	6A0	01	00	00	00	00	00	00	00
66										
67	9081.15	6A1	01	00	00	00	00	00	00	00

[SWS_Dcm_01103] Require previous confirmation before next transmission [The `Dcm` shall trigger a transmission `DcmDslPeriodicTxPduRef` only after the `DcmDslPeriodicTxConfirmationPduId` transmit confirmation for the previously transmitted periodic response is received.] ([RS_Diag_04215](#))

[SWS_Dcm_01104] Transmission error behavior [In case of a PDID transmission error, the `Dcm` shall use always the same order of periodicDIDs per client. Transmission errors shall not influence this order, the `Dcm` shall continue to retry the transmission and start the next PDID only after the PDID was transmitted successfully.] ([RS_Diag_04215](#))

[SWS_Dcm_01105] No negative response for periodic DID after positive response [After periodicDIDs are started and a positive response was sent, the `Dcm` shall start to send the periodic `DID`.] ([RS_Diag_04215](#))

[SWS_Dcm_01106] Sending periodic DIDs without further condition checks [Each time the counter of a `periodicDataIdentifiers` elapses, the `Dcm` shall retrieve the `DID` data without validating further conditions (i.e. session, security, mode dependencies, `ConditionCheckRead` and `ReadDataLength`).] ([RS_Diag_04215](#))

[SWS_Dcm_01107] No periodic DIDs in default session [When the diagnostic session changes to `DefaultSession`, the `Dcm` shall stop all scheduled periodic `DIDs`.] ([RS_Diag_04215](#))

[SWS_Dcm_01108] On session change send only periodic DIDs with allowed security settings [When the diagnostic session changes to a `non-defaultSession`, any scheduled periodic `DID` that was restricted by security access shall be stopped.] ([RS_Diag_04215](#))

[SWS_Dcm_01109] On session change send only periodic DIDs allowed in the new session [When the diagnostic session changes to a non-defaultSession, any scheduled periodic DID that is not supported in the new session shall be stopped.] (*RS_Diag_04215*)

Note: The rate for a specific transmissionMode is defined as the time between any two consecutive response messages with the same periodicDataIdentifier, when only a single periodicDID is scheduled. If multiple periodicDIDs are scheduled concurrently, the effective period between the same periodicDataIdentifier will vary based upon the following design parameters:

- The main function recurrence `DcmTaskTime`
- The number of available periodic connections
- The number of periodicDIDs that can be scheduled concurrently (see configuration parameter `DcmDspMaxPeriodicDidScheduler`).

[SWS_Dcm_01575] Connection limitation for fast transmission [If `DcmDspPeriodicTransmissionMaxPeriodicFastTransmissions` is configured, the `Dcm` shall limit the number of used periodic connections for transmissionMode 'sendAtFastRate' to `DcmDspPeriodicTransmissionMaxPeriodicFastTransmissions` connections.] (*RS_Diag_04215*)

[SWS_Dcm_01576] Connection limitation for medium transmission [If `DcmDspPeriodicTransmissionMaxPeriodicMediumTransmissions` is configured, the `Dcm` shall limit the number of used periodic connections for transmissionMode 'sendAtFastRate' to `DcmDspPeriodicTransmissionMaxPeriodicMediumTransmissions` connections.] (*RS_Diag_04215*)

[SWS_Dcm_01577] Connection limitation for slow transmission [If `DcmDspPeriodicTransmissionMaxPeriodicSlowTransmissions` is configured, the `Dcm` shall limit the number of used periodic connections for transmissionMode 'sendAtFastRate' to `DcmDspPeriodicTransmissionMaxPeriodicSlowTransmissions` connections.] (*RS_Diag_04215*)

[SWS_Dcm_01578] Connection order for limited transmissions [If any of the transmission connection limitations according to [SWS_Dcm_01575], [SWS_Dcm_01576] or [SWS_Dcm_01577] is configured, the `Dcm` shall select the `DcmDslPeriodicConnection` in the order of definition. That means that the first transmitted periodic DID shall take the first configured connections and so forth.] (*RS_Diag_04215*)

[SWS_Dcm_01110] Periodic DID transmission and security level change [On any security level change, the `Dcm` shall stop any scheduled periodic DID, that was restricted by security access, but not supported by the new security level anymore.] (*RS_Diag_04215*)

[SWS_Dcm_01111] Periodic DDDID transmission and session change [If `DcmDspDDDIDcheckPerSourceDID` is set to TRUE upon a session change, the `Dcm` shall stop any scheduled periodic DDDID, that contains source data not supported in the current session.] (*RS_Diag_04215*)

[SWS_Dcm_01112] Periodic DDDID transmission and security level change [If `DcmDspDDDIDcheckPerSourceDID` is set to TRUE, upon any security level change, the `Dcm` shall stop any scheduled periodic DDDID that contains source data not supported in the current security level.]([RS_Diag_04215](#))

[SWS_Dcm_01115] Cancel pending read operations [Upon stopping the periodical read of a data identifier while a pending asynchronous `ReadData` operation is still in progress, the `Dcm` shall cancel the pending read operation by calling `ReadData` with `OpStatus=DCM_CANCEL`.]([RS_Diag_04215](#))

[SWS_Dcm_01117] Stop cancels pending transmissions [Upon stopping the periodical read of a data identifier, the `Dcm` shall cancel pending transmissions of that `DID`.]([RS_Diag_04215](#))

[SWS_Dcm_01118] Finalizing initiated transmissions upon stop [Upon stopping the periodical read of a data identifier the `Dcm` shall finalize already initiated transmissions of that `DID`.]([RS_Diag_04215](#))

7.5.2.12 Service 0x2C - DynamicallyDefineDataIdentifier

[SWS_Dcm_00259] [The `DSP` submodule shall implement the `DynamicallyDefineDataIdentifier` (service 0x2C, diagnostic data access) of the Unified Diagnostic Services.]([\(\)](#))

The `DynamicallyDefineDataIdentifier` service is implemented internally in `Dcm` module.

[SWS_Dcm_00866] [If `DcmDDDIDStorage` configuration parameter is set to FALSE, the `Dcm` shall initialize all DDDIDs as not present at power-up `Dcm_Init`.]([\(\)](#))

[SWS_Dcm_00867] [If `DcmDDDIDStorage` configuration parameter is set to TRUE, the `Dcm` shall restore the DDDID definition from NvM at power-up `Dcm_Init`.]([\(\)](#))

[SWS_Dcm_00868] [If `DcmDDDIDStorage` configuration parameter is set to TRUE, the `Dcm` shall trigger the storage of the DDDID definition to NvRam (via `NvM_SetRamBlockStatus`).]([\(\)](#))

[SWS_Dcm_00646] [On reception of service `DynamicallyDefineDataIdentifier` with subservice `defineByIdentifier` or `defineByMemoryAddress`, the `Dcm` module shall configure this new `DID` with associated information receive from the diagnostic request: Memory address and memory length or `DID` source, position and size.]([\(\)](#))

[SWS_Dcm_00861] [On reception of the `UDS` Service `DynamicallyDefineDataIdentifier` (0x2C), the `Dcm` module shall check if the DDDID will not exceed the configured parameter value `DcmDspDDDIDMaxElements`. Otherwise (in case the number of elements will be exceeded) the `Dcm` module shall send a `NRC 0x31` (`RequestOutOfRange`).]([\(\)](#))

[SWS_Dcm_00854] [On reception of the UDS Service DynamicallyDefineDataIdentifier (0x2C) with subservice defineByMemoryAddress, the Dcm shall check if the requested AddressAndLengthFormatIdentifier is supported (refer to configuration parameter DcmDspSupportedAddressAndLengthFormatIdentifier), Otherwise the NRC 0x31 (requestOutOfRange) shall be responded. In case the container AddressAndLengthFormatIdentifier is not present, the Dcm shall accept all possible AddressAndLengthFormatIdentifiers.]()

[SWS_Dcm_00647] [On reception of service DynamicallyDefineDataIdentifier with subservice clearDynamicallyDefinedDataIdentifier, the Dcm module shall remove the configuration of this DID.]()

[SWS_Dcm_00723] [On reception of the UDS Service DynamicallyDefineDataIdentifier (0x2C), the Dcm module shall check if the DDDID can be defined in the current session (see configuration parameter DcmDspDidReadSessionRef). If not, the Dcm module shall send a NRC 0x31 (RequestOutOfRange).]()

[SWS_Dcm_00724] [On reception of the UDS Service DynamicallyDefineDataIdentifier (0x2C), the Dcm module shall check if the DDDID can be defined in the current security level (see configuration parameter DcmDspDidReadSecurityLevelRef). If not, the Dcm module shall send NRC 0x33 (Security access denied).]()

[SWS_Dcm_00725] [On reception of the UDS Service DynamicallyDefineDataIdentifier (0x2C), the Dcm module shall check if the requested Source-DIDs are supported in the current session (see configuration parameter of referenced DID DcmDspDidReadSessionRef). If not, the Dcm module shall send a NRC 0x31 (RequestOutOfRange).]()

[SWS_Dcm_00726] [On reception of the UDS Service DynamicallyDefineDataIdentifier (0x2C), the Dcm module shall check if the requested Source-DID or the memoryRange are supported in the current security level (see configuration parameter of referenced DID DcmDspDidReadSecurityLevelRef or memoryRange DcmDspReadMemoryRangeSecurityLevelRef). If not, the Dcm module shall send a NRC 0x33 (Security access denied).]()

[SWS_Dcm_00821] [On reception of the UDS Service DynamicallyDefineDataIdentifier (0x2C), the Dcm module shall check if the requested Source-DID or the memoryRange are supported in the current mode condition (see configuration parameter of referenced DID DcmDspDidReadModeRuleRef or memoryRange DcmDspReadMemoryRangeModeRuleRef). If not, the Dcm module shall send the calculated negative response code of the referenced DcmModeRule.]()

In case of memory address(es), on reception of ReadDataByIdentifier or ReadDataByPeriodicIdentifier request for a dynamically defined DID, the Dcm will use the callout Dcm_ReadMemory for all contained memory addresses to access the data.

[SWS_Dcm_01051] [On reception of the UDS Service DynamicallyDefineDataIdentifier (0x2C), if the request message contains different MemoryIdValue compare to the configured values in DcmDspMemoryIdInfo container, the Dcm shall send a NRC 0x31 (RequestOutOfRange).]()

In case of **DID** source(s), on reception of **ReadDataByIdentifier** or **ReadDataByPeriodicIdentifier** request for a dynamically defined DID, the **Dcm** will use the configuration of the contained DIDs to read the data.

7.5.2.13 Service 0x2E - WriteDataByIdentifier

[SWS_Dcm_00255] [The **Dcm** module shall implement the **UDS** Service **WriteDataByIdentifier** (0x2E) of the Unified Diagnostic Services.]()

When using Service 0x2E, the request of the tester contains a 2-byte **DID** and a **dataRecord** with the data to be written. The configuration of the **Dcm** contains a list of supported DIDs and defines for each configured DID:

- The 2-byte **DID** (see configuration parameter **DcmDspDidIdentifier**)
- For every data of the DID:
 - The function **WriteData** to be used for this data (see configuration parameters **DcmDspDataWriteFnc** and **DcmDspDataUsePort**)

[SWS_Dcm_01496] Write UDS DID authentication check [On reception of the **UDS** Service **WriteDataByIdentifier** (0x2E), the **Dcm** shall check if the write access to the requested DID is authenticated and write the data identifier only if:

- a **DcmDspDidWriteRole** is configured for that DID and the verification according to **[SWS_Dcm_01522]** was successful or
- the active white list on that connection has for the requested DID one entry with write access that matches that DID.

] (**RS_Diag_04233**)

According to **[SWS_Dcm_01537]** the authentication checks are only executed if **DcmDspAuthentication** is configured. In case of a failed authentication the NRC handling is according to **[SWS_Dcm_01544]** and **[SWS_Dcm_01551]** applies.

[SWS_Dcm_00467] [On reception of the **UDS** Service **WriteDataByIdentifier** (0x2E), the **Dcm** module shall check if the **DID** is supported (see configuration parameter **DcmDspDid** and **DcmDspDidRange**) If not, the **Dcm** module shall send **NRC** 0x31 (Request out of range) .]()

[SWS_Dcm_00562] [If a **DID** is set as unused (**DcmDspDidUsed** set to **FALSE**), the **Dcm** shall consider the **DID** as not supported (according to **[SWS_Dcm_00467]**)]()

[SWS_Dcm_00468] [On reception of the **UDS** Service **WriteDataByIdentifier** (0x2E), the **Dcm** module shall check if the **DID** has a **Write** access configured (see configuration parameter **DcmDspDidWrite** in **DcmDspDidInfo**). If not, the **Dcm** module shall send **NRC** 0x31 (Request out of range).]()

[SWS_Dcm_00469] [On reception of the UDS Service WriteDataByIdentifier (0x2E), the Dcm module shall check if the DID can be written in the current session (see configuration parameter DcmDspDidWriteSessionRef). If not, the Dcm module shall send a NRC 0x31 (Request Out of Range).]()

[SWS_Dcm_00470] [On reception of the UDS Service WriteDataByIdentifier (0x2E), the Dcm module shall check if the DID can be written in the current security level (see configuration parameter DcmDspDidWriteSecurityLevelRef). If not, the Dcm module shall send NRC 0x33 (Security access denied).]()

[SWS_Dcm_00822] [On reception of the UDS Service WriteDataByIdentifier (0x2E), the Dcm module shall check if the DID can be written in the current mode condition (see configuration parameter DcmDspDidWriteModeRuleRef). If not, the Dcm module shall send the calculated negative response code of the referenced DcmModeRule.]()

[SWS_Dcm_00473] [On reception of the UDS Service WriteDataByIdentifier (0x2E), if all signals (DcmDspDidSignal) of the DID have fixed length (DcmDspDataType is different than UINT8_DYN), the Dcm module shall check if the received data length corresponds to the DID data length (addition of all DcmDspDataByteSize).]()

[SWS_Dcm_00395] [After all verifications (see [SWS_Dcm_00467], [SWS_Dcm_00468], [SWS_Dcm_00469], [SWS_Dcm_00470], [SWS_Dcm_00473]) the Dcm module shall write all the signals (DcmDspDidSignal) of the DID by either calling the configured function DcmDspDataWriteFnc (if parameter DcmDspDataUsePort is set to USE_DATA_SYNCH_FNC or USE_DATA_ASYNCH_FNC or USE_DATA_ASYNCH_FNC_ERROR) or the associated WriteData operations (if parameter DcmDspDataUsePort is set to USE_DATA_SYNCH_CLIENT_SERVER or USE_DATA_ASYNCH_CLIENT_SERVER or USE_DATA_ASYNCH_CLIENT_SERVER_ERROR) or the associated SenderReceiver interfaces (if parameter DcmDspDataUsePort is set to USE_DATA_SENDER_RECEIVER or to USE_DATA_SENDER_RECEIVER_AS_SERVICE) with the following parameter values:

Data: the dataRecord from the request

DataLength: the number of bytes in the dataRecord (get from the configuration if the data has fixed length (DcmDspDataType is different than UINT8_DYN) or from the diagnostic request length if the data has dynamic length (DcmDspDataType is set to UINT8_DYN)).]()

[SWS_Dcm_01433] [After all verifications (see [SWS_Dcm_00467], [SWS_Dcm_00468], [SWS_Dcm_00469], [SWS_Dcm_00470], [SWS_Dcm_00473]) for DID's with DcmDspDidUsePort is set to USE_ATOMIC_SENDER_RECEIVER_INTERFACE, USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE or USE_ATOMIC_NV_DATA_INTERFACE, the Dcm module shall write the data by writing the associated sender-receiver or NvDataInterface DataServices_{DID}.]()

[SWS_Dcm_00541] [If the data is configured as a BlockId of the NvRam (parameter DcmDspDataUsePort set to USE_BLOCK_ID), the Dcm shall :

1) Request NvM_SetBlockLockStatus(<DcmDspDataBlockIdRef>, FALSE), to temporarily unlock the NvM Block (It might be locked by executing this procedure before).

2) Request `NvM_WriteBlock(<DcmDspDataBlockIdRef >, <DataBuffer>)` with `BlockId` corresponding to the configuration parameter `DcmDspDataBlockIdRef`)

3) Poll for completion of write request, using `NvM_GetErrorStatus()`

4a) On success (`NVM_REQ_OK`), the `Dcm` shall issue `NvM_SetBlockLockStatus(<DcmDspDataBlockIdRef >, TRUE)` (to lock the `NvM` block against further updates from the application) and send a positive response message.

4b) Otherwise (on any `NvM` failure) the `Dcm` module shall trigger a negative response with `NRC 0x72 (GeneralProgrammingFailure).`]()

[SWS_Dcm_CONSTR_6039] Signals with variable datalength [Only the last signal (`DcmDspDidSignal`) of a `DID` can have variable datalength (`DcmDspDataType` is set to `UINT8_DYN`).]()

In other case the `Dcm` won't be able to split the data from the request.

[SWS_Dcm_00639] [To serialize the request message of `UDS Service WriteDataByIdentifier` request into the required AUTOSAR data types (signed- and unsigned integer), the target endianness configured in `DcmDspDataEndianness` shall be considered for `DcmDspData` elements having `DcmDspDataUsePort` set to `USE_DATA_SENDER_RECEIVER`, `USE_DATA_SENDER_RECEIVER_AS_SERVICE`. In case `DcmDspDataEndianness` is not present, the `DcmDspDataDefaultEndianness` shall be used instead.]()

[SWS_Dcm_CONSTR_6018] [`DcmDspData` elements used in service `0x2E` shall not have `DcmDspDataUsePorts` set to `USE_ECU_SIGNAL`.]()

[SWS_Dcm_CONSTR_6073] Dependency for DcmDspDataWriteFnc [`DcmDspDataWriteFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_FNC_ERROR`

]()

[SWS_Dcm_01582]{DRAFT} Atomic BndM write operation [After all verifications (see [\[SWS_Dcm_00467\]](#), [\[SWS_Dcm_00468\]](#), [\[SWS_Dcm_00469\]](#), [\[SWS_Dcm_00470\]](#), [\[SWS_Dcm_00473\]](#)) for `DIDs` with `DcmDspDidUsePort` is set to `USE_ATOMIC_BNDM`, the `Dcm` module shall write the data by writing the data to the associated `BlockId` from the `BndM` (`DcmDspDidBndMBlockIdRef`) using the block specific writing function `BndM_WriteBlock_<BlockId.Shortname>`.] ([RS_Diag_04243](#))

[SWS_Dcm_01583]{DRAFT} Not allowed atomic BndM write operation [If the `Dcm` calls `BndM_WriteBlock_ 'BlockId.Shortname'` according to [\[SWS_Dcm_01582\]](#) and `BndM_WriteBlock_ 'BlockId.Shortname'` returns `E_NOT_OK`, the `Dcm` shall return a negative response `0x24 (requestSequenceError)`.] ([RS_Diag_04243](#))

Note: The BndM needs to be set into writing mode as a precondition. This is either done by the coding sub-module in [Dcm](#) or by a CDD.

[SWS_Dcm_01584]{DRAFT} Positive response on atomic BndM write operation [If the [Dcm](#) has triggered and atomic BndM write operation according to [\[SWS_Dcm_01582\]](#), the [Dcm](#) shall return a positive response if the write operation if the BndM has called [Dcm_BndMWriteBlockFinish](#) with the parameter result set to [E_OK](#).] ([RS_Diag_04243](#))

[SWS_Dcm_01585]{DRAFT} Negative response on atomic BndM write operation [If the [Dcm](#) has triggered and atomic BndM write operation according to [\[SWS_Dcm_01582\]](#), the [Dcm](#) shall return a negative response 0x72 (generalProgrammingFailure) if the write operation if the BndM has called [Dcm_BndMWriteBlockFinish](#) with the parameter result set to [E_NOT_OK](#).] ([RS_Diag_04243](#))

[SWS_Dcm_01586]{DRAFT} [If the [Dcm](#) is reading or writing a [DID](#) with [DcmDspDidUsePort](#) set to [USE_ATOMIC_BNDM](#), the [Dcm](#) shall transform the [dataRecord](#) into the corresponding BndM type. The [DID](#) data element structure needs to be compatible to the referenced [ImplementationDataType](#) of the [BndMBlockDescriptor](#). For each [DID](#) data element a corresponding sub-element shall exist in the [ImplementationDataType](#) with the same shortname and the basetypes shall be compatible.] ([RS_Diag_04243](#))

[SWS_Dcm_CONSTR_6099]{DRAFT} Consistency of BndM Block configuration [If a [DcmDspDid](#) has [DcmDspDidUsePort](#) set to [USE_ATOMIC_BNDM](#), the parameter [DcmDspDidBndMBlockIdRef](#) shall be present.] ()

7.5.2.14 Service 0x2F - InputOutputControlByIdentifier

[SWS_Dcm_00256] [The [Dcm](#) module shall implement the [UDS](#) Service [InputOutputControlByIdentifier](#) (0x2F).] ([RS_Diag_04218](#))

When using Service 0x2F, the request of the tester contains a 2-byte DID.

The configuration of the [Dcm](#) contains a list of supported DID's. For each DID, the [Dcm](#) configuration specifies:

- The 2-bytes [DID](#) (see configuration parameter [DcmDspDidIdentifier](#))
- For every data of the [DID](#) :
 - The function [Xxx_ReturnControlToECU\(\)](#) for this data (see configuration parameters [DcmDspDataReturnControlToEcuFnc](#) and [DcmDspDataUsePort](#))
 - The function [Xxx_ResetToDefault\(\)](#) for this data (see configuration parameters [DcmDspDataResetToDefaultFnc](#) and [DcmDspDataUsePort](#))

- The function `Xxx_FreezeCurrentState()` for this `DID` (see configuration parameters `DcmDspDataFreezeCurrentStateFnc` and `DcmDspDataUsePort`)
- The function `Xxx_ShortTermAdjustment()` for this `DID` (see configuration parameters `DcmDspDataShortTermAdjustmentFnc` and `DcmDspDataUsePort`)
- The sizes of the control record used in the function `Xxx_ShortTermAdjustment()` (see configuration parameter and `DcmDspDataByteSize`)

[SWS_Dcm_00579] [The `Dcm` shall support `InputOutputControlParameter` definitions according to Table 7.29.] (*RS_Diag_04218*)

Hex	Description
00	returnControlToECU
01	resetToDefault
02	freezeCurrentState
03	shortTermAdjustment

Table 7.29: InputOutputControlParameter definitions

[SWS_Dcm_01554] IOControl DID authentication check [On reception of the UDS Service `InputOutputControlByIdentifier` (0x2F), the `Dcm` shall check if the control access to the requested `DID` is authenticated and control the IO only if:

- a `DcmDspDidControlRole` is configured for that `DID` and the verification according to [SWS_Dcm_01522] was successful or
- the active white list on that connection has for the requested `DID` one entry with control access that matches that `DID`.

] (*RS_Diag_04218*)

According to [SWS_Dcm_01522] the authentication checks are only executed if `DcmDspAuthentication` is configured. In case of a failed authentication the NRC handling is according to [SWS_Dcm_01544] and [SWS_Dcm_01551] applies.

[SWS_Dcm_00563] [On reception of the UDS Service `InputOutputControlByIdentifier` (0x2F), the `Dcm` module shall check if the `DID` is supported (see configuration parameter `DcmDspDid`) If not, the `Dcm` module shall send `NRC` 0x31 (Request out of range).] (*RS_Diag_04218*)

[SWS_Dcm_00564] [If a `DID` is set as unused (`DcmDspDidUsed` set to `FALSE`), the `Dcm` shall consider the `DID` as not supported (according to [SWS_Dcm_00563])] (*RS_Diag_04218*)

[SWS_Dcm_00565] [On reception of the UDS Service `InputOutputControlByIdentifier` (0x2F), the `Dcm` module shall check if the `DID` has a Control access configured (see configuration parameter `DcmDspDidControl` in `DcmDspDidInfo`). If not, the `Dcm` module shall send `NRC` 0x31 (Request out of range).] (*RS_Diag_04218*)

[SWS_Dcm_00566] [On reception of the UDS Service InputOutputControlByIdentifier (0x2F), the Dcm module shall check if the DID can be control in the current session (see configuration parameter [DcmDspDidControlSessionRef](#)). If not, the Dcm module shall send a NRC 0x31 (Request Out of Range).]()

[SWS_Dcm_00567] [On reception of the UDS Service InputOutputControlByIdentifier (0x2F), the Dcm module shall check if the DID can be control in the current security level (see configuration parameter [DcmDspDidControlSecurityLevelRef](#)). If not, the Dcm module shall send NRC 0x33 (Security access denied).]()

[SWS_Dcm_00823] [On reception of the UDS Service InputOutputControlByIdentifier (0x2F), the Dcm module shall check if the DID can be control in the current mode condition (see configuration parameter [DcmDspDidControlModeRuleRef](#)). If not, the Dcm module shall send the calculated negative response code of the referenced [DcmModeRule](#).]()

[SWS_Dcm_00580] [On reception of a request for UDS Service InputOutputControlByIdentifier (0x2F) , if all verifications have been successfully done (see [\[SWS_Dcm_00563\]](#), [\[SWS_Dcm_00565\]](#), [\[SWS_Dcm_00566\]](#), [\[SWS_Dcm_00567\]](#)) and if the data is configured as a "ECU signal" of the IoHwAb (parameter [DcmDspDataUsePort](#)), the Dcm shall call the Api IoHwAb_Dcm_<symbolic name of ECU signal (parameter [DcmDspDataEcuSignal](#))>() with InputOutputControlParameter for the 'action' parameter and in case of InputOutputControlParameter is set to 'shortTermAdjustment' the signal value for the "signal" parameter. In this case the requirements [\[SWS_Dcm_00396\]](#), [\[SWS_Dcm_00397\]](#), [\[SWS_Dcm_00398\]](#) and [\[SWS_Dcm_00399\]](#) doesn't apply.]([RS_Diag_04218](#))

[SWS_Dcm_00581] [In case of more than one supported I/O signal per DataIdentifier and the configuration parameter [DcmDspDidControlMask](#) is set to [DCM_CONTROLMASK_INTERNAL](#), the Dcm shall internally consider the parameter controlEnableMaskRecord and control only the included signals in the request message.]([RS_Diag_04218](#))

[SWS_Dcm_CONSTR_6051] [The configuration parameter [DcmDspDidControlMaskSize](#) shall be only present if [DcmDspDidControlMask](#) is equal to [DCM_CONTROLMASK_EXTERNAL](#) or [DCM_CONTROLMASK_INTERNAL](#).]()

[SWS_Dcm_01273] [If the configuration parameter [DcmDspDidControlMask](#) is set to [DCM_CONTROLMASK_EXTERNAL](#) or [DCM_CONTROLMASK_INTERNAL](#), or the element used in service 0x2F is configured to have an atomic S/R interface, the Dcm shall reject requests without included control enable mask record with the NRC 0x13 (incorrectMessageLengthOrInvalidFormat).]([RS_Diag_04218](#))

[SWS_Dcm_01274] [If the configuration parameter [DcmDspDidControlMask](#) is set to [DCM_CONTROLMASK_NO](#), the Dcm shall reject request with included control enable mask record with the NRC 0x13 (incorrectMessageLengthOrInvalidFormat).]([RS_Diag_04218](#))

[SWS_Dcm_CONSTR_6084] Sender-receiver communication for IOControls is limited to atomic S/R interfaces [If a DID has a configured [DcmDspDidUsePort](#)

= `USE_DATA_ELEMENT_SPECIFIC_INTERFACES`, the possible values of `DcmDspDataUsePort` are limited to non S/R interfaces.](*RS_Diag_04218*)

[SWS_Dcm_CONSTR_6085] Atomic S/R for IOControls are limited to non-NV interfaces [If a `DID` has a configured `DcmDspDidControl`, the possible values of `DcmDspDidUsePort` are limited to atomic S/R interface and `USE_DATA_ELEMENT_SPECIFIC_INTERFACES`.](*RS_Diag_04218*)

[SWS_Dcm_CONSTR_6086] Signals for DIDs with Atomic S/R are not shared with other DIDs [If a `DcmDspDid` is configured to have an atomic S/R interface, all `DcmDspDataElements` referenced by this `DID` shall be referenced only from this `DID`.](*RS_Diag_04218*)

[SWS_Dcm_CONSTR_6050] [If a `DcmDspDid` is used in service 0x2F and is configured to have an atomic S/R interface, the `DcmDspDidControlMask` shall be set to `DCM_CONTROLMASK_EXTERNAL` and the parameter `DcmDspDidControlMaskSize` shall be present with a value greater than zero.]()

[SWS_Dcm_00680] Mapping of internal ControlEnableMaskRecord to DID data elements [If `DcmDspDidControlMask` is set to `DCM_CONTROLMASK_INTERNAL`, the `ControlEnableMaskRecord` shall be mapped to the `DID` data elements by applying the following mapping :

- The most significant bit of the first byte of the `ControlEnableMask` shall correspond to the first `DID` data element
- The second most significant bit of the first byte of the `ControlEnableMask` shall correspond to the second `DID` data element and continuing on in this fashion utilizing as many `ControlEnableMask` bytes as necessary to map all `DID` data elements.

] (*RS_Diag_04218*)

The `controlEnableMaskRecord` is only present, if the `DataIdentifier` supports more than one signal.

The `Dcm` supports atomic S/R interfaces activated by the configuration `DcmDspDidUsePort` set to `USE_ATOMIC_SENDER_RECEIVER_INTERFACE` or `USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE`. In the text and requirements of this chapter the term 'atomic S/R interface' for IO control means that the IO controlled `DID` is configured to one of the two choices.

The service use case [TPS_SWCT_01654] and the [constr_1679] limits the S/R interfaces used for IOControl to explicit S/R communication. In implicit communication is not supported by the `Dcm`.

[SWS_Dcm_01434] IOControl General execution sequence [On reception of a request for `UDS` Service `InputOutputControlByIdentifier` (0x2F) the `Dcm` shall first execute the service verifications according to [SWS_Dcm_00563], [SWS_Dcm_00565], [SWS_Dcm_00566], [SWS_Dcm_00567] and on successful passing the verifications start the configured service processing.](*RS_Diag_04218*)

[SWS_Dcm_00396] [On reception of a request for UDS Service InputOutputControlByIdentifier (0x2F) with `inputOutputControlParameter` equal to `returnControlToEcu`, the `Dcm` module shall invoke all impacted configured function of the `controlEnableMaskRecord` (if parameter `DcmDspDataUsePort` set to `USE_DATA_SYNCH_FNC` or `USE_DATA_ASYNCH_FNC` or `USE_DATA_ASYNCH_FNC_ERROR`; see configuration parameter `DcmDspDataReturnControlToEcuFnc`). Alternatively call all the associated `ReturnControlToECU` operations (if parameter `DcmDspDataUsePort` set to `USE_DATA_SYNCH_CLIENT_SERVER` or `USE_DATA_ASYNCH_CLIENT_SERVER` or `USE_DATA_ASYNCH_CLIENT_SERVER_ERROR`) for every data of the `DID` received in the request.]([RS_Diag_04218](#))

[SWS_Dcm_00397] [On reception of a request for UDS Service InputOutputControlByIdentifier (0x2F) with `inputOutputControlParameter` equal to `resetToDefault`, the `Dcm` module shall invoke all impacted configured function of the `controlEnableMaskRecord` (if parameter `DcmDspDataUsePort` set to `USE_DATA_SYNCH_FNC` or `USE_DATA_ASYNCH_FNC` or `USE_DATA_ASYNCH_FNC_ERROR`; see configuration parameter `DcmDspDataResetToDefaultFnc`). Alternatively call all the associated `ResetToDefault` operations (if parameter `DcmDspDataUsePort` set to `USE_DATA_SYNCH_CLIENT_SERVER` or `USE_DATA_ASYNCH_CLIENT_SERVER` or `USE_DATA_ASYNCH_CLIENT_SERVER_ERROR`) for every data of the `DID` received in the request.]([RS_Diag_04218](#))

[SWS_Dcm_00398] [On reception of a request for UDS Service InputOutputControlByIdentifier (0x2F) with `inputOutputControlParameter` equal to `freezeCurrentState`, the `Dcm` module shall invoke all impacted configured function of the `controlEnableMaskRecord` (if parameter `DcmDspDataUsePort` set to `USE_DATA_SYNCH_FNC` or `USE_DATA_ASYNCH_FNC` or `USE_DATA_ASYNCH_FNC_ERROR`; see configuration parameter `DcmDspDataFreezeCurrentStateFnc`). Alternatively call all the associated `FreezeCurrentState` operations (if parameter `DcmDspDataUsePort` set to `USE_DATA_SYNCH_CLIENT_SERVER` or `USE_DATA_ASYNCH_CLIENT_SERVER` or `USE_DATA_ASYNCH_CLIENT_SERVER_ERROR`) for every data of the `DID` received in the request.]([RS_Diag_04218](#))

[SWS_Dcm_00399] [On reception of a request for UDS Service InputOutputControlByIdentifier (0x2F) with `inputOutputControlParameter` equal to `shortTermAdjustment`, the `Dcm` module shall invoke all impacted configured function of the `controlEnableMaskRecord` (if parameter `DcmDspDataUsePort` set to `USE_DATA_SYNCH_FNC` or `USE_DATA_ASYNCH_FNC` or `USE_DATA_ASYNCH_FNC_ERROR`; see configuration parameter `DcmDspDataShortTermAdjustmentFnc`). Alternatively call all the associated `ShortTermAdjustment` operations (if parameter `DcmDspDataUsePort` set to `USE_DATA_SYNCH_CLIENT_SERVER` or `USE_DATA_ASYNCH_CLIENT_SERVER` or `USE_DATA_ASYNCH_CLIENT_SERVER_ERROR`) for every data of the `DID` received in the request.]([RS_Diag_04218](#))

[SWS_Dcm_00858] Cancel active IO controls on session change [On any session transition, the `Dcm` shall stop all active IO controls according to [\[SWS_Dcm_01435\]](#) which are not supported by the new session.]([RS_Diag_04119](#))

[SWS_Dcm_00628] Cancel active IO controls in default sessions [On a session transition to default session, the `Dcm` shall stop all active IO controls according to [\[SWS_Dcm_01435\]](#).] ([RS_Diag_04119](#))

[SWS_Dcm_00859] Cancel active IO controls on security level change [On any security level change, the `Dcm` shall stop all active IO controls according to [\[SWS_Dcm_01435\]](#) which are not support by the new security level anymore.] ([RS_Diag_04119](#))

[SWS_Dcm_01435] Dcm cancel IO control sequence [If the `Dcm` needs to cancel an active IO control due to [\[SWS_Dcm_00858\]](#), [\[SWS_Dcm_00628\]](#) or [\[SWS_Dcm_00859\]](#), the `Dcm` shall do the following:

- For controlled data elements with `DcmDspDataUsePort` set to `USE_ECU_SIGNAL`: call to `IoHwAb_Dcm_<symbolic ECU signal name>()` with 'action' parameter set to `ReturnControlToECU`.
- For controlled data elements with `DcmDspDataUsePort` set to `USE_DATA_ASYNC_CLIENT_SERVER` or `USE_DATA_SYNC_CLIENT_SERVER` or `USE_DATA_ASYNC_CLIENT_SERVER_ERROR`: call the C/S interface operation `ReturnControlToECU`.
- For controlled data elements with `DcmDspDataUsePort` set to `USE_DATA_SYNC_FNC` or `USE_DATA_ASYNC_FNC` or `USE_DATA_ASYNC_FNC_ERROR`: call the configured function `Xxx_ReturnControlToECU` (see parameter `DcmDspDataReturnControlToEcuFnc`)
- For controlled `DIDs` with is configured atomic S/R interfaces: update the data element `IOOperationRequest` with `inputOutputControlParameter = 0x00`, the `controlEnableMask = 0xFFFFFFFF1` and data element `underControl = 0x00`.

] ([RS_Diag_04119](#))

[SWS_Dcm_00640] [To serialize the required AUTOSAR data types (signed- and unsigned integer) from the request message (in case of `inputOutputControlParameter` is set to 'shortTermAdjustment') / into the response message of UDS Service `InputOutputControlByIdentifier` responses, the target endianness configured in `DcmDspDataEndianness` shall be considered for `DcmDspData` elements having `DcmDspDataUsePort` set to `USE_ECU_SIGNAL`. In case `DcmDspDataEndianness` is not present, the `DcmDspDataDefaultEndianness` shall be used instead.] ([RS_Diag_04218](#))

[SWS_Dcm_00682] [The `controlState` in the `ControlStatusRecord` for positive response message of `IoControl` service shall be retrieved using the associated `ReadData` operation/function/SenderReceiver after application processing on the IO control request is positively finalized.] ([RS_Diag_04218](#))

¹The size of the mask depends on the parameter `DcmDspDidControlMaskSize`

Beside the Client/Server interface, the `Dcm` provides the SenderReceiver interface `IOControlRequest_DID`. The `underControl` data element of this interface is calculated by the `Dcm` with one state bit for each data element identical to the `CEMR`. Applications can directly derive the active control enable status without the need to maintain internal states.

The bit-mask `underControl` contains the accumulated status about which data elements of this particular I/O is currently under diagnostic control. The normal operation state could be derived if the value of `underControl` is set to `0x00` (which is the initial value). Each set bit indicates a data element which is under diagnostic control via `FreezeCurrentState`, `ResetToDefault` or `ShortTermAdjustment`.

[SWS_Dcm_01436] Calculation of the underControl data element [The `Dcm` shall calculate the `underControl` data element of the S/R interface `IOControlRequest_DID`. The `underControl` is a bitfield of the same size than the `CEMR` of the controlled `DID`. Each bit represents the same data element as in the `CEMR`. A value of 0 indicates, that the corresponding data element is currently not controlled by the `Dcm`, a value of 1 indicates that it is controlled. The initial value is 0, each control request for a data element with `inputOutputControlParameter` equal to `ResetToDefault`, `FreezeCurrentState` or `ShortTermAdjustment` will set the corresponding bit value to 1, and each control request with `inputOutputControlParameter` set to `ReturnControlToECU` will set the bit value to 0.] (*RS_Diag_04218*)

With each I/O Control request a command `IOOperationRequest` is provided to the application to update the input or the respective output. `IOOperationRequest` contains the `inputOutputControlParameter`, the `controlEnableMask` and in case of `ShortTermAdjustment` the `controlState`.

To identify that previous operation has finished (e.g Write `IOControlRequest_{DID}`), the user can use the update flag mechanism from the [RTE](#).

The application needs to update their output values and finalizes the request with the response message `IOOperationResponse` to the `Dcm`. The possible values are:

- `0x00` positive response (similar to `E_OK`)
- `0x10` `generalReject`
- `0x21` `busyRepeatRequest`
- `0x22` `conditionsNotCorrect`
- `0x26` `FailurePreventsExecutionOfRequestedAction`
- `0x31` `requestOutOfRange`
- `0x78` `ResponsePending` (similar to `E_PENDING`)

Based on the write trigger of the `SW-C` to `IOControlResponse_DID.IOOperationResponse`, the `Dcm` will:

1. wait for final processing (`0x78`)

2. send a positive response message (0x00)
3. send a negative response message (all other values, except 0xFF)

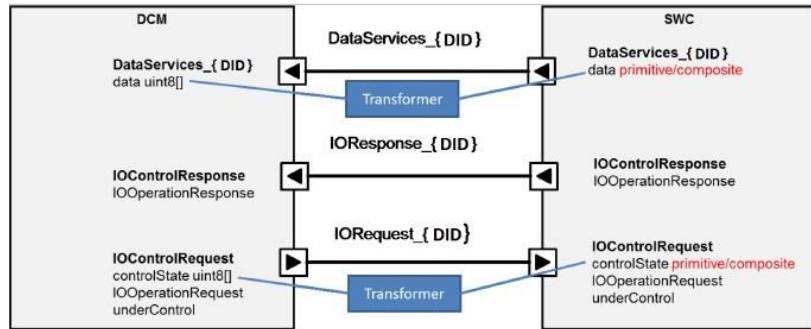


Figure 7.11: IO-Control with Sender/Receiver interfaces

[SWS_Dcm_01437] inputOutputControlParameter idle state [The `inputOutputControlParameter` of data element of `IOOperationRequest` from S/R interface `IOControlRequest_DID` shall have an initial value of 0xFF. This value indicates the application, that the `Dcm` is currently not processing an `UDS` service to control this `DID`.] ([RS_Diag_04218](#))

[SWS_Dcm_01438] inputOutputControlParameter after processing an UDS InputOutputControlByIdentifier (0x2F) service [If the `Dcm` is processing an `InputOutputControlByIdentifier` request with `inputOutputControlParameter` equal to `ResetToDefault`, `FreezeCurrentState` or `ShortTermAdjustment`, the `Dcm` shall set the `inputOutputControlParameter` of data element of `IOOperationRequest` from S/R interface `IOControlRequest_DID` to the idle state 0xFF after the application has set the `IOControlResponse_DID.IOOperationResponse` to 0x00 and before processing other `InputOutputControlByIdentifier` requests.] ([RS_Diag_04218](#))

Upon the `Dcm` writes `IOOperationRequest` of `IOControlRequest_DID` the `SWC` processes the IO control request. The `SWC` informs the `Dcm` about the current processing state by updating `IOControlResponse_DID.IOOperationResponse`.

[SWS_Dcm_01439] Positive response based on IOOperationResponse [If the `Dcm` is processing an `InputOutputControlByIdentifier` request, it shall reply with a positive response, if the applications set `IOControlResponse_DID.IOOperationResponse` to 0x00.] ([RS_Diag_04218](#))

[SWS_Dcm_01440] Negative response based on IOOperationResponse [If the `Dcm` is processing an `InputOutputControlByIdentifier` request, it shall reply with a negative response with the `NRC` `IOControlResponse_DID.IOOperationResponse`, if the applications set `IOControlResponse_DID.IOOperationResponse` a value different to 0x00 and 0x78.] ([RS_Diag_04218](#))

[SWS_Dcm_01441] RCRRP based on IOOperationResponse [If the `Dcm` is processing an `InputOutputControlByIdentifier` request and the `IOControlResponse_DID.IOOperationResponse` has a value of `0x78`, the `Dcm` shall wait until the `IOControlResponse_DID.IOOperationResponse` gets a value different to `0x78` and send `RCRRP` according to [\[SWS_Dcm_00024\]](#).] ([RS_Diag_04218](#))

Note: As shown in figure ??: Sequence diagram for `resetToDefault`, `freezeCurrentState` and `shortTermAdjustment`, the trigger from the `SW-C` to the `Dcm` is the explicit S/R write operation. The use of the `RTE` functionality "IsUpdated" is a possible mechanism for the `Dcm` to detect a write from `SW-C` to the S/R data element.

[SWS_Dcm_01275] Common action for all inputOutputControlParameter operations with atomic S/R [If the `Dcm` is processing an `InputOutputControlByIdentifier` request for a `DID` configured atomic S/R interface, the `Dcm` module shall update in the `IOControlRequest_DID` the data element `IOOperationRequest` with

1. `controlEnableMask` = `controlEnableMaskRecord` of the request message
2. `inputOutputControlParameter` = `inputOutputControlParameter` from the request message

] ([RS_Diag_04218](#))

The value `0xFF` of the `inputOutputControlParameter` of the command `IOOperationRequest` is the 'idle' state. The values `0x00` (`ReturnControlToECU`), `0x01` (`ResetToDefault`), `0x02` (`FreezeCurrentState`) or `0x03` (`ShortTermAdjustment`) start the request processing and include the control option `inputOutputControlParameter`, `controlEnableMask` and `controlState` (for `ShortTermAdjustment` only).

[SWS_Dcm_01277] Additional action for InputOutputControl operations for ShortTermAdjustment with atomic S/R [If the `Dcm` is processing an `InputOutputControlByIdentifier` request with `inputOutputControlParameter` equal to `ShortTermAdjustment` for a `DID` with configured atomic S/R interface, in addition to [\[SWS_Dcm_01275\]](#) the `Dcm` module shall update in the `IOControlRequest_DID` the data element `controlState` with content of the `controlState` from the request message.] ([RS_Diag_04218](#))

Note: The `controlState` is a separate data element that it can be optionally processed by a data transformer to transform the byte stream into a composite type (see [Figure 7.11: IO-Control with Sender/Receiver interfaces](#)).

An example of the `Dcm` S/R interaction is given in [Figure 7.12](#) and [Figure 7.13](#). For `ReturnControlToECU` the data from the request is provided to the application, the `Dcm` will continue to finalize the request after writing the data into the S/R ports. All other sub-functions will wait for the application providing the response `0x00` or an code.

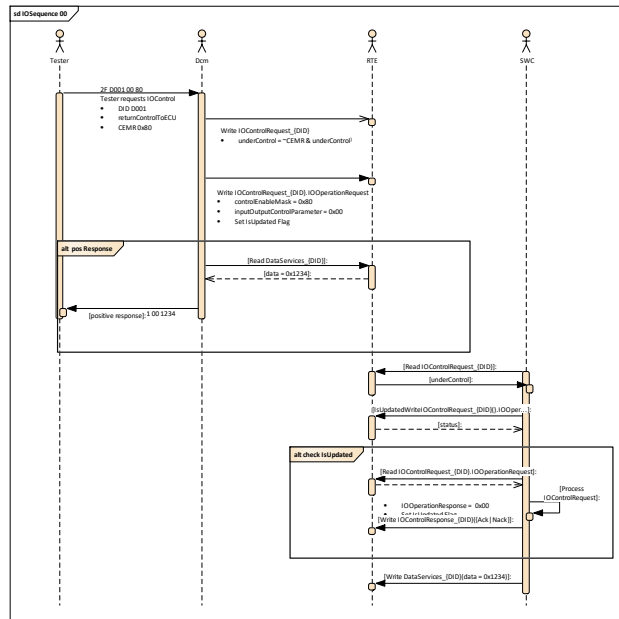


Figure 7.12: IOSequence 00

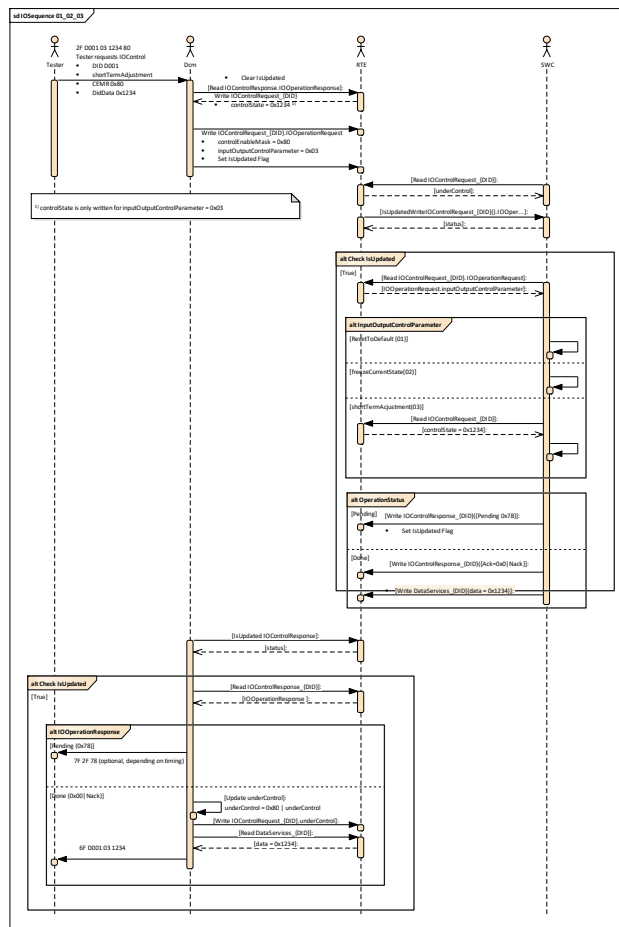


Figure 7.13: IOSequence 01 02 03

[SWS_Dcm_CONSTR_6048] Composite sub elements accessible only by read [Composite sub elements can only be referred from Read `DID` i.e. Write and Control `DID` are not supported.]()

[SWS_Dcm_CONSTR_6030] [The ReturnControlToEcu functionality is existing if at least one of the following parameters are activated : `DcmDspDidFreezeCurrentState` in `ECUC_Dcm_00624` : or `DcmDspDidResetToDefault` in `ECUC_Dcm_00623` : or `DcmDspDidShortTermAdjustment` in `ECUC_Dcm_00625` :.]()

[SWS_Dcm_CONSTR_6059] Dependency for `DcmDspDataFreezeCurrentStateFnc` [`DcmDspDataFreezeCurrentStateFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC_ERROR`

]()

[SWS_Dcm_CONSTR_6063] Dependency for `DcmDspDataResetToDefaultFnc` [`DcmDspDataResetToDefaultFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC_ERROR`

]()

[SWS_Dcm_CONSTR_6064] Dependency for `DcmDspDidControlMaskSize` [`DcmDspDidControlMaskSize` shall be only present if `DcmDspDidControlMask` is equal to `DCM_CONTROLMASK_EXTERNAL` or `DCM_CONTROLMASK_INTERNAL`.]()

[SWS_Dcm_CONSTR_6081] Dependency for `DcmDspDidControlMaskBitPosition` [The value configured for `DcmDspDidControlMaskBitPosition` shall be lower than `DcmDspDidControlMaskSize * 8`.]()

[SWS_Dcm_CONSTR_6065] Dependency for `DcmDspDataReturnControlToEcuFnc` [`DcmDspDataReturnControlToEcuFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC_ERROR`

]()

[SWS_Dcm_CONSTR_6066] Dependency for `DcmDspDataShortTermAdjustmentFnc` [`DcmDspDataShortTermAdjustmentFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC_ERROR`

]()

[SWS_Dcm_CONSTR_6082] Dependency for `DcmDspDidControlMaskSize` [`DcmDspDidControlMaskSize` larger than 4 shall be only allowed if `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_CLIENT_SERVER`, `USE_DATA_ASYNCH_CLIENT_SERVER_ERROR` or `USE_DATA_SYNCH_CLIENT_SERVER`. Note: `ControlEnableMask` larger than 32 bits is a very rare use case. Therefore the `Dcm` supports only C/S interfaces to solve this use case.]()

7.5.2.15 Service 0x31 - RoutineControl

[SWS_Dcm_00257] [The `Dcm` module shall implement the UDS Service RoutineControl (0x31) for subFunctions `startRoutine`, `stopRoutine` and `requestsRoutineResults`.]
()

A tester can use UDS Service 0x31 to start, stop or obtain the results of a routine identified by a 2-byte `routineIdentifier`. The `Dcm` module configuration contains a list of the `routineIdentifiers` (see configuration parameter `DcmDspRoutineIdentifier`) supported by the DCM. For each `routineIdentifier`, the `Dcm` configuration specifies:

- The function `Xxx_Start()` associated with this `routineIdentifier` (see configuration parameters `DcmDspStartRoutineFnc` and `DcmDspRoutineUsePort`)
- List of signal available in the request and in the response (see configuration parameters `DcmDspStartRoutineIn` and `DcmDspStartRoutineOut`)
- The function `Xxx_Stop()` associated with this `routineIdentifier` (see configuration parameters `DcmDspStopRoutineFnc` and `DcmDspRoutineUsePort`)
- List of signal available in the request and in the response (see configuration parameters `DcmDspStopRoutineIn` and `DcmDspStopRoutineOut`)
- The function `Xxx_RequestResults()` associated with this `routineIdentifier` (see configuration parameters `DcmDspRequestRoutineResultsFnc` and `DcmDspRoutineUsePort`)
- List of signal available in the request and in the response (see configuration parameters `DcmDspRequestRoutineResultsIn` and `DcmDspRequestRoutineResultsOut`)

[SWS_Dcm_01442] [If `DcmDspRoutineUsePort` is set to true, the `Dcm` shall call the corresponding operation of the C/S interfaces `RoutineServices_RoutineName` to process this routine.]([RS_Diag_04224](#))

A routine handler processes the diagnostic routine control request. The `Dcm` passes the `routineControlOption` as input parameters to the routine handler. The routine processes the request and writes the result to the output parameters of the routine handler call. In case of shared Rx and Tx buffer, there are two cases where writing to the output parameters of the routine control can cause overwriting of the input parameters:

- a routine control uses arrays as output parameters (call by reference)
- a routine control writes to output parameters and returns `DCM_E_PENDING`. It is called again with `DcmOpstatus` set to `DCM_PENDING`. The input parameters are derived from the Rx buffer that was overwritten by the previous write to the output parameters

For efficient memory consumption it is controllable, if the `Dcm` applies further strategies to separate input and output parameters in those situations.

[SWS_Dcm_01580] Input and output consistency during C/S based routine handling [If `DcmDspRoutineUsePort` is set to `TRUE` and the `ClientServerOperation.diagArgIntegrity` of that operation is also set to `TRUE`, the `Dcm` shall ensure that the routine handler in the C/S interfaces `RoutineServices_RoutineName` writing to the output parameters will not overwrite the input parameters.]([RS_Diag_04224](#))

[SWS_Dcm_01581] "Routine argument integrity for C/S calls [If `DcmDspRoutineUsePort` is set to `true` and `DcmDspRoutineInterfaceArgumentIntegrity` is configured, the `Dcm` shall create the corresponding operation of the C/S interface `RoutineServices_RoutineName` with `ClientServerOperation.diagArgIntegrity` set to `DcmDspRoutineInterfaceArgumentIntegrity`.]([RS_Diag_04224](#))

[SWS_Dcm_CONSTR_6097] Availability of `DcmDspRoutineInterfaceArgumentIntegrity` [`DcmDspRoutineInterfaceArgumentIntegrity` shall only be available if the corresponding `DcmDspRoutine` has `DcmDspRoutineUsePort` is set to `true`.]()

[SWS_Dcm_01443] [If `DcmDspRoutineUsePort` is set to `false`, the `Dcm` shall use the configured callout functions for routine operations.]([RS_Diag_04224](#))

[SWS_Dcm_00568] [On reception of the `UDS` Service `RoutineControl` (0x31), the `Dcm` module shall check if the Routine is supported (see configuration parameter `DcmDspRoutine`) If not, the `Dcm` module shall send `NRC` 0x31 (Request out of range).]()

[SWS_Dcm_00569] [If a Routine is set as unused (`DcmDspRoutineUsed` set to `FALSE`), the `Dcm` shall consider the Routine as not supported (according to [\[SWS_Dcm_00568\]](#))]()

[SWS_Dcm_01555] StartRoutine authentication check [On reception of the `UDS` Service `RoutineControl` (0x31) with sub-function `startRoutine`, the `Dcm` shall check if the access to the requested routine identifier is authenticated and process the routine only if:

- a `DcmDspStartRoutineRole` is configured for that routine and the verification according to [\[SWS_Dcm_01522\]](#) was successful or
- the active white list on that connection has one RID entry with sub-function access set to `startRoutine` that matches that service and sub-function.

]([RS_Diag_04233](#))

[SWS_Dcm_01556] StopRoutine authentication check [On reception of the UDS Service RoutineControl (0x31) with sub-function `stopRoutine`, the Dcm shall check if the access to the requested routine identifier is authenticated and process the routine only if:

- a `DcmDspStopRoutineRole` is configured for that routine and the verification according to [\[SWS_Dcm_01522\]](#) was successful or
- the active white list on that connection has one RID entry with sub-function access set to `stopRoutine` that matches that service and sub-function.

]([RS_Diag_04233](#))

[SWS_Dcm_01557] RequestRoutineResult authentication check [On reception of the UDS Service RoutineControl (0x31) with sub-function `requestRoutineResult`, the Dcm shall check if the access to the requested routine identifier is authenticated and process the routine only if:

- a `DcmDspRequestRoutineResultsRole` is configured for that routine and the verification according to [\[SWS_Dcm_01522\]](#) was successful or
- the active white list on that connection has one RID entry with sub-function access set to `requestRoutineResults` that matches that service and sub-function.

]([RS_Diag_04233](#))

According to [\[SWS_Dcm_01537\]](#) the authentication checks are only executed if `DcmDspAuthentication` is configured. In case of a failed authentication the NRC handling is according to [\[SWS_Dcm_01544\]](#) and [\[SWS_Dcm_01551\]](#) applies.

[SWS_Dcm_00570] [On reception of the UDS Service RoutineControl (0x31), the Dcm module shall check if the Routine can be executed in the current session (see configuration parameters `DcmDspStartRoutineCommonAuthorizationRef`, `DcmDspStopRoutineCommonAuthorizationRef` and `DcmDspRequestRoutineResultsCommonAuthorizationRef`). If not, the Dcm module shall send a NRC 0x31 (Request Out of Range).]()

[SWS_Dcm_00571] [On reception of the UDS Service RoutineControl (0x31), the Dcm module shall check if the Routine can be executed in the current security level (see configuration parameter `DcmDspStartRoutineCommonAuthorizationRef`, `DcmDspStopRoutineCommonAuthorizationRef` and `DcmDspRequestRoutineResultsCommonAuthorizationRef`). If not, the Dcm module shall send NRC 0x33 (Security access denied).]()

[SWS_Dcm_00869] [On reception of the UDS Service RoutineControl (0x31), the Dcm module shall check if the SubFunction to the corresponding Routine is supported (see existence of configuration container [DcmDspStopRoutine](#) for SubFunction 0x02; [DcmDspRequestRoutineResults](#) for SubFunction 0x03). If not, the Dcm module shall send NRC 0x12 (SubFunction not supported).]()

[SWS_Dcm_01169] [On reception of the UDS Service RoutineControl (0x31) with SubFunction startRoutine, the Dcm module shall check if the Routine can be executed in the current mode condition (see configuration parameter [DcmDspStartRoutineCommonAuthorizationRef](#)). If not, the Dcm module shall send the calculated negative response code of the referenced [DcmModeRule](#).]()

[SWS_Dcm_01170] [On reception of the UDS Service RoutineControl (0x31) with SubFunction stopRoutine, the Dcm module shall check if the Routine can be executed in the current mode condition (see configuration parameter [DcmDspStopRoutineCommonAuthorizationRef](#)). If not, the Dcm module shall send the calculated negative response code of the referenced [DcmModeRule](#).]()

[SWS_Dcm_01171] [On reception of the UDS Service RoutineControl (0x31) with SubFunction requestRoutineResults, the Dcm module shall check if the Routine can be executed in the current mode condition (see configuration parameter [DcmDspRequestRoutineResultsCommonAuthorizationRef](#)). If not, the Dcm module shall send the calculated negative response code of the referenced [DcmModeRule](#).]()

Routines have different input and output parameters depending on the routine configuration (e.g. [DcmDspStartRoutineIn](#) for input parameter for the routine start service). The signature of the called routine operations Xxx_Start, Xxx_Stop and Xxx_RequestResults is depending on this configuration. The defined parameters for input and output routine data are optional, and marked in brackets '['] in the definition in [\[SWS_Dcm_01203\]](#), [\[SWS_Dcm_01204\]](#) and [\[SWS_Dcm_91013\]](#).

[SWS_Dcm_01360] [For each configured routine input signal in [DcmDspStartRoutineInSignal](#), [DcmDspStopRoutineInSignal](#) or [DcmDspRequestRoutineResultsInSignal](#) with a signal type unequal to VARIABLE_LENGTH, the optional parameter 'DcmDspRoutineSignalType dataIn_n' shall be provided in the corresponding operations in [\[SWS_Dcm_01203\]](#), [\[SWS_Dcm_01204\]](#) or [\[SWS_Dcm_91013\]](#).]()

[SWS_Dcm_01361] [For a configured routine input signal in [DcmDspStartRoutineInSignal](#), [DcmDspStopRoutineInSignal](#) or [DcmDspRequestRoutineResultsInSignal](#) with a signal type equal to VARIABLE_LENGTH the optional parameter const 'uint8 * dataInVar' shall be provided in the corresponding operations in [\[SWS_Dcm_01203\]](#) [\[SWS_Dcm_01204\]](#) or [\[SWS_Dcm_91013\]](#).]()

[SWS_Dcm_01362] [For each configured routine output signal in [DcmDspStartRoutineOutSignal](#), [DcmDspStopRoutineOutSignal](#) or [DcmDspRequestRoutineResultsOutSignal](#) with a signal type unequal to VARIABLE_LENGTH the optional parameter 'DcmDspRoutineSignalType dataOut_n' shall be provided in the corresponding operations in [\[SWS_Dcm_01203\]](#), [\[SWS_Dcm_01204\]](#) or [\[SWS_Dcm_91013\]](#).]()

[SWS_Dcm_01363] [For a configured routine output signal in [DcmDspStartRoutineOutSignal](#), [DcmDspStopRoutineOutSignal](#) or [DcmDspRequestRoutineResultsOutSignal](#) with a signal type equal to VARIABLE_LENGTH the optional parameter `const 'uint8 * dataOutVar'` shall be provided in the corresponding operations in [\[SWS_Dcm_01203\]](#), [\[SWS_Dcm_01204\]](#) or [\[SWS_Dcm_91013\]](#).]()

[SWS_Dcm_01364] [The optional in/out parameter `currentDataLength` in [\[SWS_Dcm_01203\]](#), [\[SWS_Dcm_01204\]](#) or [\[SWS_Dcm_91013\]](#) is always present if at least one of the routine input signal data or routine output signal data have a signal with routine type 'VARIABLE_LENGTH'.]()

Note: The 'currentDataLength' parameter as in/out parameter contains the data length in bytes of the 'dataInVar' while calling the operation and it returns the length in bytes of the 'dataOutVar'. As 'dataInVar' and 'dataOutVar' are optional, 'currentDataLength' is only present if at least one of this optional parameter is used.

[SWS_Dcm_00590] [When receiving a request for UDS Service RoutineControl (0x31) if all verifications have been successfully done (see [\[SWS_Dcm_00568\]](#), [\[SWS_Dcm_00570\]](#), [\[SWS_Dcm_00571\]](#)), the `Dcm` module shall split the routineControlOptionRecord received according of the list of input signal configured for this routine (see configuration parameters [DcmDspStartRoutineIn](#), [DcmDspStopRoutineIn](#), [DcmDspRequestRoutineResultsIn](#)).]()

[SWS_Dcm_00400] [When receiving a request for UDS Service RoutineControl (0x31) with subfunction startRoutine, if all verifications have been successfully done (see [\[SWS_Dcm_00568\]](#), [\[SWS_Dcm_00570\]](#), [\[SWS_Dcm_00571\]](#)), the `Dcm` module shall call the configured `Xxx_Start()` function passing the `dataIn`, calculated from routineControlOptionRecord (see [\[SWS_Dcm_00590\]](#)), and the `dataOut` reference according of the list of output signal configured for this routine (see configuration parameter [DcmDspStartRoutineOut](#)).]()

[SWS_Dcm_00401] [Upon completing [\[SWS_Dcm_00400\]](#), when `Xxx_Start()` returns `E_OK`, the `Dcm` module shall reply with a positive response with the data returned by `Xxx_Start()` in the `dataOut` as routineStatusRecord (`dataOut` are merged according to the list of output signal configured for this routine (see configuration parameter [DcmDspStartRoutineOut](#))).]()

[SWS_Dcm_00402] [When receiving a request for UDS Service RoutineControl (0x31) with subfunction stopRoutine, if all verifications have been successfully done (see [\[SWS_Dcm_00568\]](#), [\[SWS_Dcm_00570\]](#), [\[SWS_Dcm_00571\]](#)), the `Dcm` module shall call the configured `Xxx_Stop()` function passing the `dataIn`, calculated from routineControlOptionRecord (see [\[SWS_Dcm_00590\]](#)), and the `dataOut` reference according of the list of output signal configured for this routine (see configuration parameter [DcmDspStopRoutineOut](#)).]()

[SWS_Dcm_00403] [Upon completing [\[SWS_Dcm_00402\]](#), when `Xxx_Stop()` returns `E_OK`, the `Dcm` module shall reply with a positive response with the data returned by `Xxx_Stop()` in the `dataOut` as routineStatusRecord (`dataOut` are merged according

to the list of output signal configured for this routine (see configuration parameter `DcmDspStopRoutineOut`)).]()

[SWS_Dcm_00404] [When receiving a request for `UDS Service RoutineControl (0x31)` with subfunction `requestRoutineResults`, if all verifications have been successfully done (see `[SWS_Dcm_00568]`, `[SWS_Dcm_00570]`, `[SWS_Dcm_00571]`), the `Dcm` module shall call the configured `Xxx_RequestResults()` function passing the `dataIn`, calculated from `routineControlOptionRecord` (see `[SWS_Dcm_00590]`) and provide the `dataOut` reference according of the list of output signal configured for this routine (see configuration parameter `DcmDspRequestRoutineResultsOut`)).]()

[SWS_Dcm_00405] [Upon completing `[SWS_Dcm_00404]`, when `Xxx_RequestResults()` returns `E_OK`, the `Dcm` module shall reply with a positive response with the data returned by `Xxx_RequestResults()` in the `dataOut` as `routineStatusRecord` (`dataOut` are merged according to the list of output signal configured for this routine (see configuration parameter `DcmDspRequestRoutineResultsOut`)).]()

[SWS_Dcm_00641] [To serialize the required AUTOSAR data types (signed- and unsigned integer) from the request message / into the response message of `UDS Service RoutineControl`, the target endianness configured in `DcmDspRoutineSignalEndianness` shall be considered for `DcmDspRoutine` signals having `set` to fixed length (`DcmDspRoutineSignalType` set to other value than `VARIABLE_LENGTH`)).]()

[SWS_Dcm_CONSTR_6072] Dependency for `DcmDspRoutineSignalEndianness` [In case `DcmDspRoutineSignalEndianness` is not present, the `DcmDspDataDefaultEndianness` shall be used instead.]()

[SWS_Dcm_01139] [The `Dcm` shall follow the `NRC` handling for `RoutineControlService` according to ISO 14229-1 [1].]()

[SWS_Dcm_01140] [On reception of the `UDS Service RoutineControl (0x31)`, the `Dcm` module shall check the overall length of the request. If length of the request is wrong, the `Dcm` module shall send `NRC 0x13` (Incorrect message length or invalid format) to the tester.]()

[SWS_Dcm_01141] [The `Dcm` shall call the appropriate routine functions of the `SWC` after having performed the total length check and the Mode rules, security level and session checks (`DcmDspStartRoutineCommonAuthorizationRef`, `DcmDspStopRoutineCommonAuthorizationRef` and `DcmDspRequestRoutineResultsCommonAuthorizationRef`)).]()

Note: Subsequent checks have to be performed by the `SWC`.

[SWS_Dcm_01194] [On reception of the `UDS Service RoutineControl (0x31)`, for every requested `RID` inside the `OBD` range (`E000-E0FF`), the `Dcm` shall implicitly allow subfunction `StartRoutine`].]()

[SWS_Dcm_00701] [On reception of the `UDS Service RoutineControl (0x31)`, for every requested `RID` inside the `OBD` range (`E000-E0FF`) and usage of `UDS` interface, the `Dcm`

module shall use the `routineInfo` byte value from `DcmDspRoutineInfoByte` in the response to the tester.]()

[SWS_Dcm_01330] [If `DcmDspEnableObdMirror` is set to true, an explicitly configured `RID` inside the `OBD` range (E000-E0FF) shall use the `UDS` interface.]()

[SWS_Dcm_01331] [If `DcmDspEnableObdMirror` is set to false, all requests within the `OBD RID` range shall use the `UDS` interface.]()

[SWS_Dcm_01332] [On reception of the `UDS` Service RoutineControl (0x31), for every requested `RID` inside the `OBD` range (E000-E0FF), the `Dcm` module shall handle the `RID` as defined for `OBD` Service \$08 (see [SWS_Dcm_00418], [SWS_Dcm_00947], [SWS_Dcm_00419], [SWS_Dcm_00420], [SWS_Dcm_00948], [SWS_Dcm_01192]) if `DcmDspEnableObdMirror` is set to true and `RID` not explicitly configured.]()

[SWS_Dcm_01333] [On reception of the `UDS` Service RoutineControl (0x31), for every requested `RID` inside the `OBD` range (E000-E0FF) and usage of `OBD` interface, the `Dcm` shall use the `routineInfo` byte value from `DcmDspRequestControlInfoByte` in the response to the tester.]()

If `DcmDspEnableObdMirror` is set to FALSE or the `RID` is explicitly configured inside the `OBD TestId` range (E000-E0FF), the access to the `OBD` data shall be given in the following way:

[SWS_Dcm_01390] [On reception of an `UDS` Service RoutineControl (0x31) request with one or more "availability `OBDDTestIds`" as parameter, the `Dcm` module shall respond with the corresponding supported (=configured) `RIDs`.]()

[SWS_Dcm_01391] [On reception of an `UDS` Service RoutineControl (0x31) request "availability `OBDDTestIds`" together with other `OBDDTestIds` as parameter, the `Dcm` module shall ignore the request.]()

[SWS_Dcm_01392] [On reception of an `UDS` Service RoutineControl (0x31) request with a `OBDDTestIds` that is not an "availability `OBDDTestIds`", the `Dcm` module shall invoke the configured `Xxx_Start()` function.]()

[SWS_Dcm_01393] [As specified in [3, SAE J1979], unused data bytes shall be filled with \$00.]()

[SWS_Dcm_01394] [If `Xxx_Start()` doesn't return `E_OK`, the `Dcm` shall return `NRC` 0x22.]()

[SWS_Dcm_00668] [If the operation `Start()` returns value `E_NOT_OK`, the `Dcm` module shall send a negative response with `NRC` code equal to `ErrorCode` parameter value.]()

[SWS_Dcm_00669] [If the operation `Start()` returns value `DCM_E_FORCE_RCRRP`, the `Dcm` module shall start the transmission of `NRC` 0x78.]()

[SWS_Dcm_00670] [If the operation `Stop()` returns value `E_NOT_OK`, the `Dcm` module shall send a negative response with `NRC` code equal to `ErrorCode` parameter value.]()

[SWS_Dcm_00671] [If the operation `Stop()` returns value `DCM_E_FORCE_RCRRP`, the `Dcm` module shall start the transmission of `NRC 0x78`.]()

[SWS_Dcm_00672] [If the operation `RequestResults()` returns value `E_NOT_OK`, the `Dcm` module shall send a negative response with `NRC` code equal to `ErrorCode` parameter value.]()

[SWS_Dcm_00673] [If the operation `RequestResults()` returns value `DCM_E_FORCE_RCRRP`, the `Dcm` module shall start the transmission of `NRC 0x78`.]()

[SWS_Dcm_CONSTR_6071] Dependency for `DcmDspStartRoutineFnc`, `DcmDspStopRoutineFnc`, `DcmDspRequestRoutineResultsFnc`, `DcmDspStartRoutineConfirmationFnc`, `DcmDspStopRoutineConfirmationFnc` [The following configuration parameters shall only be present if `DcmDspRoutineUsePort` is set to `FALSE`.

- `DcmDspStartRoutineFnc`
- `DcmDspStopRoutineFnc`
- `DcmDspRequestRoutineResultsFnc`
- `DcmDspStartRoutineConfirmationFnc`
- `DcmDspStopRoutineConfirmationFnc`

]()

7.5.2.16 Service 0x3E - Tester Present

[SWS_Dcm_00251] [The `Dcm` module shall implement the Tester Present (service 0x3E, diagnostic communication and security) of the Unified Diagnostic Services for the subfunction values 0x00 and 0x80.]()

[SWS_Dcm_01558] Skipping authentication check for tester present [The `Dcm` shall process the UDS service 0x3E (TesterPresent) independently from the current authentication state.]([RS_Diag_04230](#))

This service is used to keep one or multiple servers in a diagnostic session being different than the defaultSession.

7.5.2.17 Service 0x3D - WriteMemoryByAddress

[SWS_Dcm_00488] [The `Dcm` module shall implement the WriteMemoryByAddress (service 0x3D) of the Unified Diagnostic Services.]()

This service is used to write data using a physical memory address.

[SWS_Dcm_00855] [On reception of the UDS Service WriteMemoryByAddress

(0x3D), the *Dcm* shall check if the requested *AddressAndLengthFormatIdentifier* is supported (refer to configuration parameter *DcmDspSupportedAddressAndLengthFormatIdentifier*), Otherwise the *NRC 0x31* (requestOutOfRange) shall be responded. In case the container *AddressAndLengthFormatIdentifier* is not present, the *Dcm* shall accept all possible *AddressAndLengthFormatIdentifiers*.]()

[SWS_Dcm_00489] [On reception of the *UDS Service WriteMemoryByAddress* (0x3D), the *Dcm* shall check if the complete memory range to write to (from 'memoryAddress' parameter to 'memoryAddress + memorySize -1') is inside the allowed memory ranges (check of *DcmDspWriteMemoryRangeLow* and *DcmDspWriteMemoryRangeHigh* parameters for each *DcmDspWriteMemoryRangeInfo* container or *DcmDspWriteMemoryRangeByLabelLow* and *DcmDspWriteMemoryRangeByLabelHigh* parameters for each *DcmDspWriteMemoryRangeByLabelInfo* container). If not, the *Dcm* module shall send *NRC 0x31* (Request out of range).]()

[SWS_Dcm_00490] [On reception of the *UDS Service WriteMemoryByAddress* (0x3D), the *Dcm* shall check if the complete memory range (from 'memoryAddress' parameter to 'memoryAddress + memorySize -1') can be written in the current security level (see *DcmDspWriteMemoryRangeSecurityLevelRef*). If security level is not correct, the *Dcm* module shall send *NRC 0x33* (securityAccessDenied).]()

[SWS_Dcm_00825] [On reception of the *UDS Service WriteMemoryByAddress* (0x3D), the *Dcm* shall check if the complete memory range (from 'memoryAddress' parameter to 'memoryAddress + memorySize -1') can be written in the current mode condition (see *DcmDspWriteMemoryRangeModeRuleRef*). If mode condition is not correct, the *Dcm* module shall send the calculated negative response code of the referenced *dcmModeRule*.]()

[SWS_Dcm_00491] [On reception of the *UDS Service WriteMemoryByAddress* (0x3D), and after verification of the validity of the request (see [SWS_Dcm_00489] and [SWS_Dcm_00490]) the *Dcm* module shall call the callout *Dcm_WriteMemory*.]()

[SWS_Dcm_01052] [On reception of the *UDS Service WriteMemoryByAddress* (0x3D), if the request message contains different *MemoryIdValue* compare to the configured values in *DcmDspMemoryIdInfo* container, the *Dcm* shall send a *NRC 0x31* (RequestOutOfRange).]()

[SWS_Dcm_01056] [The configured ranges of memory address (*DcmDspReadMemoryRangeHigh* and *DcmDspReadMemoryRangeLow* or *DcmDspReadMemoryRangeByLabelHigh* and *DcmDspReadMemoryRangeByLabelLow*) shall not overlap each other.]()

[SWS_Dcm_01358] [On reception of the *UDS Service WriteMemoryByAddress* (0x3D), the *Dcm* shall check if the complete memory range (from 'memoryAddress' parameter to 'memoryAddress + memorySize -1') can be written in the current session (see *DcmDspWriteMemoryRangeSessionLevelRef*). If the session is not correct, the *Dcm* module shall send *NRC 0x31* (RequestOutOfRange).]()

[SWS_Dcm_00643] [If the operation `Dcm_WriteMemory` returns `DCM_WRITE_FAILED`, the `Dcm` module shall send a negative response with `NRC` code equal to the parameter `ErrorCode` parameter value.]()

[SWS_Dcm_00837] [If the call to `Dcm_WriteMemory` returns `DCM_WRITE_FORCE_RCRRP`, the `Dcm` shall invoke the transmit request for RCR-RP (`NRC` 0x78 transmission) and the `Dcm` shall not realize further invocation of the operation till RCR-RP is transmitted.]()

[SWS_Dcm_00838] [After transmit confirmation of a RCR-RP transmitted on the context of **[SWS_Dcm_00837]**, the `Dcm` calls, from `Dcm_MainFunction` (due to call context), `Dcm_WriteMemory` again with `OpStatus` = `DCM_FORCE_RCRRP_OK`.]()

7.5.2.18 Service 0x23 - ReadMemoryByAddress

This service is used to read data using a physical memory address.

[SWS_Dcm_00492] [The `Dcm` module shall implement the `ReadMemoryByAddress` (service 0x23) of the Unified Diagnostic Services.]()

[SWS_Dcm_00853] [On reception of the `UDS` Service `ReadMemoryByAddress` (0x23), the `Dcm` shall check if the requested `AddressAndLengthFormatIdentifier` is supported (refer to configuration parameter `DcmDspSupportedAddressAndLengthFormatIdentifier`), Otherwise the `NRC` 0x31 (`requestOutOfRange`) shall be responded. In case the container `DcmDspAddressAndLengthFormatIdentifier` is not present, the `Dcm` shall accept all possible `AddressAndLengthFormatIdentifiers`. .]()

[SWS_Dcm_00493] [On reception of the `UDS` Service `ReadMemoryByAddress` (0x23), the `Dcm` shall check if the complete memory range to read from (from 'memoryAddress' parameter to 'memoryAddress + memorySize -1') is inside the allowed memory ranges (check of `DcmDspReadMemoryRangeLow` and `DcmDspReadMemoryRangeHigh` parameters for each `DcmDspReadMemoryRangeInfo` container or `DcmDspReadMemoryRangeByLabelLow` and `DcmDspReadMemoryRangeByLabelHigh` parameters for each `DcmDspReadMemoryRangeByLabelInfo` container). If not, the `Dcm` module shall send `NRC` 0x31 (`Request out of range`).]()

[SWS_Dcm_00494] [On reception of the `UDS` Service `ReadMemoryByAddress` (0x23), the `Dcm` shall check if the complete memory range (from 'memoryAddress' parameter to 'memoryAddress + memorySize -1') can be readen in the current security level (see `DcmDspReadMemoryRangeSecurityLevelRef`). If security level is not correct, the `Dcm` module shall send `NRC` 0x33 (`securityAccessDenied`).]()

[SWS_Dcm_00826] [On reception of the `UDS` Service `ReadMemoryByAddress` (0x23), the `Dcm` shall check if the complete memory range (from 'memoryAddress' parameter to 'memoryAddress + memorySize -1') can be readen in the current mode condition (see `DcmDspReadMemoryRangeModeRuleRef`). If mode condition is not

correct, the `Dcm` module shall send calculated negative response code of the referenced `DcmModeRule`.]()

[SWS_Dcm_00495] [On reception of the `UDS` Service `ReadMemoryByAddress` (0x23), and after verification of the validity of the request (see [\[SWS_Dcm_00493\]](#) and [\[SWS_Dcm_00494\]](#)) the `Dcm` module shall call the callout `Dcm_ReadMemory`.]()

[SWS_Dcm_01053] [On reception of the `UDS` Service `ReadMemoryByAddress` (0x23), if the request message contains different `MemoryIdValue` compare to the configured values in `DcmDspMemoryIdInfo` container, the `Dcm` shall send a `NRC` 0x31 (`RequestOutOfRange`).]()

[SWS_Dcm_01158] [The configured ranges of memory address (`DcmDspReadMemoryRangeHigh` and `DcmDspReadMemoryRangeLow` or `DcmDspReadMemoryRangeByLabelHigh` and `DcmDspReadMemoryRangeByLabelLow`) shall not overlap each other.]()

[SWS_Dcm_01359] [On reception of the `UDS` Service `ReadMemoryByAddress` (0x23), the `Dcm` shall check if the complete memory range (from 'memoryAddress' parameter to 'memoryAddress + memorySize -1') can be read in the current session (see `DcmDspReadMemoryRangeSessionLevelRef`). If the session is not correct, the `Dcm` module shall send `NRC` 0x31 (`RequestOutOfRange`).]()

[SWS_Dcm_00644] [If the operation `Dcm_ReadMemory` returns `DCM_READ_FAILED`, the `Dcm` module shall send a negative response with `NRC` code equal to the parameter `ErrorCode` parameter value.]()

[SWS_Dcm_00839] [If the call to `Dcm_ReadMemory` returns `DCM_READ_FORCE_RCRRP`, the `Dcm` shall invoke the transmit request for RCR-RP (`NRC` 0x78 transmission) and the `Dcm` shall not realize further invocation of the operation till RCR-RP is transmitted.]()

[SWS_Dcm_00840] [After transmit confirmation of a RCR-RP transmitted on the context of [\[SWS_Dcm_00839\]](#), the `Dcm` calls, from `Dcm_MainFunction` (due to call context), `Dcm_ReadMemory` again with `OpStatus` = `DCM_FORCE_RCRRP_OK`.]()

7.5.2.19 Service 0x34 - RequestDownload

This service is used to request the start of a download process.

[SWS_Dcm_00496] [The `Dcm` module shall implement the `RequestDownload` (service 0x34) of the Unified Diagnostic Services.]([RS_Diag_04033](#))

[SWS_Dcm_00856] [On reception of the `UDS` Service `RequestDownload` (0x34), the `Dcm` shall check if the requested `AddressAndLengthFormatIdentifier` is supported (refer to configuration parameter `DcmDspSupportedAddressAndLengthFormatIdentifier`), Otherwise the `NRC` 0x31 (`requestOutOfRange`) shall be responded. In case the container `AddressAndLengthFormatIdentifier` is not present, the `Dcm` shall accept all possible `AddressAndLengthFormatIdentifiers`.]()

[SWS_Dcm_01057] [On reception of the [UDS ServiceRequestDownload \(0x34\)](#), if the request message contains different [MemoryIdValue](#) compare to the configured values in [DcmDspMemoryIdInfo](#) container, the [Dcm](#) shall send a [NRC 0x31 \(RequestOutOfRange\)](#).] ()

[SWS_Dcm_01132] [[NRC](#) described in [Table 7.30](#) shall be the responsibility of the callout function.] ()

NRC	Use Case
0x31 requestOutOfRange	The specified dataFormatIdentifier is not valid.
0x70 uploadDownload-NotAccepted	An attempt to download to a server's memory cannot be accomplished due to some fault conditions. Note: this NRC will be handled by the callout only if mode rule is not used for this case

Table 7.30: NRC managed by callout function for service 0x34

Note: the callout function can, if needed, return also other [NRC](#) but the ones above won't be treated by the [Dcm](#) module.

[SWS_Dcm_00757] [If the operation [Xxx_ProcessRequestDownload](#) returns value [E_NOT_OK](#), the [Dcm](#) module shall send a negative response with [NRC](#) code equal to the parameter [ErrorCode](#) parameter value.] ()

[SWS_Dcm_01417] [Upon calling [Xxx_ProcessRequestDownload](#), the [Dcm](#) shall write the maximum possible buffer size into the [BlockLength](#) parameter.] ([RS_Diag_04033](#))

[SWS_Dcm_01418] [If the function call [Xxx_ProcessRequestDownload](#) returns a requested buffer length larger than the supported buffer length of the current protocol connection, the [Dcm](#) shall report the [Det](#) error [DCM_E_INTERFACE_BUFFER_OVERFLOW](#).] ([RS_Diag_04033](#)) For definition of [DCM_E_INTERFACE_BUFFER_OVERFLOW](#) see [Table 7.1](#).

[SWS_Dcm_01419] [If the function call [Xxx_ProcessRequestDownload](#) returns a requested buffer length smaller or equal than the supported buffer length of the current protocol connection, the [Dcm](#) shall return the [BlockLength](#) value within the [maxNumberOfBlockLength](#) parameter of the positive response.] ([RS_Diag_04033](#))

7.5.2.20 Service 0x35 - RequestUpload

This service is used to request the start of a upload process.

[SWS_Dcm_00499] [The [Dcm](#) module shall implement the [RequestUpload](#) (service 0x35) of the [Unified Diagnostic Services](#).] ([RS_Diag_04033](#))

[SWS_Dcm_00857] [On reception of the [UDS RequestUpload \(0x35\)](#), the [Dcm](#) shall check if the requested [AddressAndLengthFormatIdentifier](#) is supported (refer

to configuration parameter `DcmDspSupportedAddressAndLengthFormatIdentifier`, Otherwise the `NRC 0x31` (`requestOutOfRange`) shall be responded. In case the container `AddressAndLengthFormatIdentifier` is not present, the `Dcm` shall accept all possible `AddressAndLengthFormatIdentifiers`.]()

[SWS_Dcm_01055] [On reception of the `UDS RequestUpload (0x35)`, if the request message contains different `MemoryIdValue` compare to the configured values in `DcmDspMemoryIdInfo` container, the `Dcm` shall send a `NRC 0x31` (`RequestOutOfRange`).]()

[SWS_Dcm_01133] [`NRC` described in Table 7.31 shall be the responsibility of the callout function.]()

NRC	Use Case
0x31 - requestOutOfRange	The specified <code>dataFormatIdentifier</code> is not valid.
0x70 - uploadDownloadNotAccepted	An attempt to download to a server's memory cannot be accomplished due to some fault conditions. Note: this <code>NRC</code> will be handled by the callout only if mode rule is not used for this case

Table 7.31: NRC managed by callout function for service 0x35

Note: the callout function can, if needed, return also other `NRC` but the ones above won't be treated by the `Dcm` module.

[SWS_Dcm_00758] [If the operation `Xxx_ProcessRequestUpload` returns value `E_NOT_OK`, the `Dcm` module shall send a negative response with `NRC` code equal to the parameter `ErrorCode` parameter value.]()

[SWS_Dcm_01420] [Upon calling `Xxx_ProcessRequestUpload`, the `Dcm` shall write the maximum possible buffer size into the `BlockLength` parameter.](*RS_Diag_04033*)

[SWS_Dcm_01421] [If the function call `Xxx_ProcessRequestUpload` returns a requested buffer length larger than the supported buffer length of the current protocol connection, the `Dcm` shall report the `Det` error `DCM_E_INTERFACE_BUFFER_OVERFLOW`.](*RS_Diag_04033*) For definition of `DCM_E_INTERFACE_BUFFER_OVERFLOW` see Table 7.1.

[SWS_Dcm_01422] [If the function call `Xxx_ProcessRequestUpload` returns a requested buffer length smaller or equal than the supported buffer length of the current protocol connection, the `Dcm` shall return the `BlockLength` value within the `maxNumberOfBlockLength` parameter of the positive response.](*RS_Diag_04033*)

7.5.2.21 Service 0x36 - TransferData

This service is used to transfer data during a download or upload process.

[SWS_Dcm_00502] [The `Dcm` module shall implement the `TransferData` (service 0x36) of the Unified Diagnostic Services.] ([RS_Diag_04033](#))

[SWS_Dcm_00503] [On reception of the `UDS` Service `TransferData` (0x36), if a download process is running (`RequestDownload` service has been previously received) and the request format is correct, the `Dcm` module shall call the callout `Xxx_ProcessTransferDataWrite`.] ([RS_Diag_04033](#))

[SWS_Dcm_00504] [On reception of the `UDS` Service `TransferData` (0x36), if an upload process is running (`RequestUpload` service has been previously received) and the request format is correct, the `Dcm` module shall call the callout `Xxx_ProcessTransferDataRead`.] ([RS_Diag_04033](#))

[SWS_Dcm_00645] [On reception of the `UDS` Service `TransferData` (0x36), if a block sequence error is detected, the `Dcm` module shall trigger a negative response with `NRC` 0x73 (`WrongBlockSequenceCounter`)] ()

[SWS_Dcm_01444] [On reception of the `UDS` Service `TransferData` (0x36), if a file download is running (`RequestFileTransfer` service has been previously received with 0x01 (`AddFile`) or 0x03 (`ReplaceFile`)) and the request format is correct, the `Dcm` module shall call the callout `Dcm_WriteFile`.] ()

[SWS_Dcm_01445] [On reception of the `UDS` Service `TransferData` (0x36), if a file or directory information upload is running (`RequestFileTransfer` service has been previously received with 0x04 (`ReadFile`) or 0x05 (`ReadDir`)) and the request format is correct, the `Dcm` module shall call the callout `Dcm_ReadFileOrDir`.] ()

[SWS_Dcm_01173] [NRCs described in Table 7.32 shall be the responsibility of the callout function.] ()

NRC	Use Case
0x24 - requestSequenceError	only for the following conditions: If the <code>RequestDownload</code> or <code>RequestUpload</code> service is active, but the server has already received all data as determined by the <code>memorySize</code> parameter in the active <code>RequestDownload</code> or <code>RequestUpload</code> service
0x31 - requestOutOfRange	Only for the following conditions: The <code>transferRequestParameterRecord</code> contains additional control parameters (e.g. additional address information) and this control information is invalid. The <code>transferRequestParameterRecord</code> is not consistent with the server's memory alignment constraints
0x71 - transferDataSuspended	The data transfer operation was halted due to some fault.
0x72 - generalProgrammingFailure	If the server detects an error when finalizing the data transfer between the client and server (e.g., via an integrity check).
0x92 - voltageTooHigh	The voltage measured is higher than the maximum acceptable voltage for downloading data.
0x93 - voltageTooLow	The voltage measured is under the minimum acceptable voltage for downloading data.

NRC	Use Case
-----	----------

Table 7.32: NRC managed by callout function for service 0x36

Note: the callout function can, if needed, return also other NRCs but the ones above won't be treated by the `Dcm` module.

7.5.2.22 Service 0x37 - RequestTransferExit

This service is used to terminate a download or upload process.

[SWS_Dcm_00505] [The `Dcm` module shall implement the `RequestTransferExit` (service 0x37) of the Unified Diagnostic Services.] (*RS_Diag_04033*)

[SWS_Dcm_01134] [NRC described in Table 7.33 shall be the responsibility of the callout function.] ()

NRC	Use Case
0x24 - requestSequenceError	The programming process is not completed when a request for this service is received.
0x72 - generalProgrammingFailure	If the server detects an error when finalizing the data transfer between the client and server (e.g., via an integrity check).

Table 7.33: NRC managed by callout function for service 0x37

Note: the callout function can, if needed, return also other NRC but the ones above won't be treated by the `Dcm` module.

[SWS_Dcm_00759] [If the operation `Xxx_ProcessRequestTransferExit` returns value `E_NOT_OK`, the `Dcm` module shall send a negative response with NRC code equal to the parameter `ErrorCode` parameter value.] ()

7.5.2.23 Service 0x38 - RequestFileTransfer

[SWS_Dcm_01083] [The `Dcm` module shall implement the `RequestFileTransfer` (service 0x38) of the Unified Diagnostic Services.] ()

This service is used to request the start of a file transfer process according to ISO-14229-1.

[SWS_Dcm_01446] [If the `DcmRequestFileTransferUsePort` is set to `TRUE` the `Dcm` shall use C/S calls of the interface `RequestFileTransfer` for all `RequestFileTransfer` related callouts.] ()

[SWS_Dcm_01447] [If the `DcmRequestFileTransferUsePort` is set to `FALSE` the `Dcm` shall use C function calls for all `RequestFileTransfer` related callouts.] ()

[SWS_Dcm_01448] [If the `fileSizeParameterLength` parameter in the `RequestFileTransfer` request is present and outside the closed interval `[0x01..0x08]`, the `Dcm` shall send a negative response with `NRC 0x31 (RequestOutOfRange)`.]()

[SWS_Dcm_01449] [If the `modeOfOperation` parameter in the `RequestFileTransfer` request is not supported (`0x00` or greater than `0x05`), the `Dcm` shall send a negative response with `NRC 0x31 (RequestOutOfRange)`.]()

[SWS_Dcm_01450] [The `Dcm` shall process `RequestFileTransfer` according to [5] and, in case of `modeOfOperation` equal to `0x01 (AddFile)` call `XXX_ProcessRequestAddFile` after the full length check.]()

[SWS_Dcm_01451] [The `Dcm` shall process `RequestFileTransfer` according to [5] and, in case of `modeOfOperation` equal to `0x02 (DeleteFile)` call `XXX_ProcessRequestDeleteFile` after the full length check.]()

[SWS_Dcm_01452] [The `Dcm` shall process `RequestFileTransfer` according to [5] and, in case of `modeOfOperation` equal to `0x03 (ReplaceFile)` call `XXX_ProcessRequestReplaceFile` after the full length check.]()

[SWS_Dcm_01453] [The `Dcm` shall process `RequestFileTransfer` according to [5] and, in case of `modeOfOperation` equal to `0x04 (ReadFile)` call `XXX_ProcessRequestReadFile` after the full length check.]()

[SWS_Dcm_01454] [The `Dcm` shall process `RequestFileTransfer` according to [5] and, in case of `modeOfOperation` equal to `0x05 (ReadDir)` call `XXX_ProcessRequestReadDir` after the full length check.]()

[SWS_Dcm_01088] [If any of the file transfer operations `XXX_ProcessRequest<yyy>` returns value `E_NOT_OK`, the `Dcm` module shall send a negative response with `NRC` code equal to the parameter `ErrorCode` parameter value.]()

[SWS_Dcm_01455] [The `Dcm` shall use the value configured in `DcmRequestFileTransferFileSizeOrDirInfoParameterLength` as value sent in `fileSizeOrDirInfoParameterLength` in the response.]()

[SWS_Dcm_01090] [The `Dcm` shall use the value configured in `DcmRequestFileTransferFileSizeOrDirInfoParameterLength` as number of bytes sent in `fileSizeUncompressedOrDirInfoLength` and `fileSizeCompressed` in the response.]()

[SWS_Dcm_01456] [The `Dcm` shall use the value configured in `DcmRequestFileTransferLengthFormatIdentifier` as value sent in `lengthFormatIdentifier` in the response.]()

[SWS_Dcm_01091] [The `Dcm` shall use the value configured in `DcmRequestFileTransferLengthFormatIdentifier` as number of bytes sent in `maxNumberOfBlockLength` in the response.]()

[SWS_Dcm_01457] [If the `maxNumberOfBlockLength` does not fit into the requested `lengthFormatIdentifier` number of bytes, the `Dcm` shall send `NRC 0x10 (generalReject)`.]()

[SWS_Dcm_01458] [If the `fileSizeUncompressedOrDirInfoLength` or `fileSizeCompressed` do not fit into the requested `fileSizeOrDirInfoParameterLength` number of bytes, the `Dcm` shall send `NRC 0x10` (generalReject).]()

7.5.2.24 Service 0x85 - ControlDTCSetting

An external test tool can request an ECU to either disable or enable `DTC` storage in the ECUs error memory by sending a `UDS Service 0x85` request with sub-function `0x01` ("ON") or `0x02` ("OFF").

[SWS_Dcm_00249] [The `Dcm` module shall implement `UDS Service ControlDTCSetting (0x85)` to enable or disable the storage of `DTCs` in the ECUs error memory.]([RS_Diag_04159](#))

[SWS_Dcm_01399] [If the `Dcm` receives a `ControlDTCSetting (0x85)` service with `DTCSettingControlOptionRecord != 0xFFFFFFFF`, the `Dcm` shall send a `NRC 0x31` (requestOutOfRange).]([RS_Diag_04159](#))

[SWS_Dcm_01063] [On reception of `UDS Service 0x85` with subfunction `0x01` (`DTCSettingType "ON"`), the `Dcm` shall call `Dem_EnabledDTCSetting()` with `ClientId = ClientId` for this `Dcm` instance (see [DcmDemClientRef](#)).]([RS_Diag_04115](#))

[SWS_Dcm_00783] [In case of `Dem_EnabledDTCSetting` returns `E_OK` (see [\[SWS_Dcm_01063\]](#)), the `Dcm` shall invoke a mode switch of the `ModeDeclarationGroupPrototype DcmControlDTCSetting` by calling `SchM_Switch_<bsnp>_DcmControlDTCSetting (RTE_MODE_DcmControlDTCSetting_ENABLEDDTCSETTING)`.]()

[SWS_Dcm_00406] [On reception of `UDS Service 0x85` with subfunction `0x02` (`DTCSettingType "OFF"`), the `Dcm` shall call `Dem_DisabledDTCSetting()` with `ClientId = ClientId` for this `Dcm` instance (see [DcmDemClientRef](#)).]([RS_Diag_04115](#))

[SWS_Dcm_00784] [In case of `Dem_DisabledDTCSetting` returns `E_OK` (see [\[SWS_Dcm_00406\]](#)), the `Dcm` shall invoke a mode switch of the `ModeDeclarationGroupPrototype DcmControlDTCSetting` by calling `SchM_Switch_<bsnp>_DcmControlDTCSetting (RTE_MODE_DcmControlDTCSetting_DISABLEDTCSETTING)`.]()

[SWS_Dcm_00751] [In case the `DTCSetting` is disabled and a transitions to default session or upon any diagnostic session change where the new session does not support `UDS Service ControlDTCSetting` anymore, the `Dcm` module shall call `Dem_EnabledDTCSetting()` with the following parameters

- `ClientId`: `ClientId` for this `Dcm` instance (see [DcmDemClientRef](#))

and switch the mode `DcmControlDTCSetting` to `DCM_ENABLEDDTCSETTING`.]()

For some use-cases the `Dcm` may re-enable the `controlDTCSetting` due to external changed mode conditions:

[SWS_Dcm_00752] [In case the `DTCSetting` is disabled and at least one referenced arbitrary `ModeDeclarationGroupPrototypes` (see configuration parameter `DcmDspControlDTCSettingReEnableModeRuleRef`) for service `ControlDTCSetting` (0x85) with `DTCSettingType` "OFF" (0x02) are not fulfilled anymore, the `Dcm` module shall call `Dem_EnabledDTCSetting()` with the following parameters:

- `ClientId`: Client Id for this `Dcm` instance (see `DcmDemClientRef`)

and switch the mode `DcmControlDTCSetting` to `DCM_ENABLEDDTCSETTING`]()

Note: If at least one `ModeDeclarationGroupPrototypes` is configured (see configuration parameter `DcmDspControlDTCSettingReEnableModeRuleRef`) for service `ControlDTCSetting` (0x85) with `DTCSettingType` "OFF" (0x02), it is recommended to activate the condition check for the according sub-function 0x02(see configuration parameter `DcmDsdSubServiceModeRuleRef`).

Note: This active observation of the referenced mode declaration groups can either be achieved by polling the mode condition in each `MainFunction` cycle or by attaching to the change notification of mode declaration group (SchM will trigger a `BSWEntity` in `Dcm` on changes of this mode declaration group).

[SWS_Dcm_00829] [If the configuration parameter `DcmSupportDTCSettingControlOptionRecord` is set to true and the length of `DTCSettingControlOptionRecord` in the request is different from 3 bytes, the `Dcm` shall return NRC 0x13 (Incorrect message length or invalid format).]()

[SWS_Dcm_01564] [If the configuration parameter `DcmSupportDTCSettingControlOptionRecord` is set to false and the request contains any data after the sub-function, the `Dcm` shall return NRC 0x13 (Incorrect message length or invalid format).]()

7.5.2.25 Service 0x87 - LinkControl

This service is used to gain bus bandwidth for diagnostic purposes.

The Service `LinkControl` (0x87) is user optional. There are different project specific use cases which are not handled in the default `Dcm`. One use case is to switch the bandwidth in application an other use case performs an OEM bootloader jump.

Therefore the service `LinkControl` needs to be implemented project specific as external service (refer to Chapter 8.8 `Dcm` as Service-Component).

7.5.3 OBD Services

7.5.3.1 Overview

The following table defines the `OBD` Services supported by the `DCM`.

Relevant OB D Service Identifier	Support in the DCM
\$01	Supported
\$02	Supported
\$03	Supported
\$04	Supported
\$06	Supported
\$07	Supported
\$08	Supported
\$09	Supported
\$0A	Supported

Table 7.34: Support for **OBD services in the DCM**

7.5.3.2 General behavior

In many cases, the **Dcm** protocol allows the bundling of several requests (for example several "PIDs") and the corresponding bundling of the responses. The descriptions of the behavior for the individual services do not explicitly consider this. As the **Dcm** needs to comply with **OB**D standard (as is defined through various requirements below), the **Dcm** might need to repeat the steps defined below to parse a request and assemble a valid response.

In a vehicle there can be 3 different types of **OB**D ECUs:

- Master ECU (one per vehicle)
- Primary ECU (several per vehicle)
- Dependent / Secondary ECUs (several per vehicle)

From the Basic Software point of view Dependent / Secondary ECUs doesn't need any specific **OB**D functionality. In Dependent / Secondary ECUs **OB**D-relevant information will not be stored in the Basic Software (e.g. no direct communication with the scan tool). The respective **OB**D functionality might be handled in Dependent / Secondary ECUs by a SWC.

The following **OB**D requirements are only valid for Master and Primary ECUs. If necessary the **OB**D requirements differentiate between Master and Primary Requirement.

The following table gives an overview about which **OB**D functionality must be supported in a Master ECU, Primary ECU or Dependent / Secondary ECU:

Functionality	Master ECU	Primary ECU	Dependent / Secondary ECU
OB D Scantool Communication	Yes	Yes	No

Table 7.35: Overview about **OBD functionality in different **OB**D ECUs**

[SWS_Dcm_00077] [When calling the DEM module for OBD services, the Dcm module shall use the following values for the parameter DTCTOrigin:
Service \$0A uses DEM_DTC_ORIGIN_PERMANENT_MEMORY
All other services use DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY] (*RS_Diag_04058*)

7.5.3.3 Service \$01 - Request Current Powertrain Diagnostic Data

[SWS_Dcm_00243] [The Dcm module shall implement the OBD service \$01 (Request Current Powertrain diagnostic Data) in compliance to all provisions of the OBD standard.] ()

Using Service \$01, an external test tool can request an emission-related ECU to return PID-values or to return the supported PIDs. OBD reserves certain PIDs for the special purpose of obtaining the list of available PIDs in a certain range. These PIDs are called "availability PIDs" and are \$00, \$20, \$40, \$60, \$80, \$A0, \$C0 and \$E0.

The Dcm collects the PID information from 1 to n SW-Cs. This applies in particular for PIDs which contain several data values for potentially different sources. Example: PID\$83 reports Nox Sensor Data for sensor 1 and sensor 2 in one composed PID which might come from different SW-C.

The Dcm configuration defines the PIDs that are available on the ECU. The Dcm configuration defines for each such PID:

- The PID Identifier (see configuration parameter *DcmDspPidIdentifier*)
- Indication of the PID is used or not (for postbuild configuration) (see configuration parameter *DcmDspPidUsed*)
- The size of the PID (see configuration parameter *DcmDspPidSize*)
- The supported information for this PID (see configuration parameter *DcmDspPidSupportInfo*)
- List of data (*DcmDspPidData*) for the PID with the following configuration for every data
 - The length of the data associated with the PID (see configuration parameter *DcmDspPidDataByteSize*)
 - The position of the data in the PID (see configuration parameter *DcmDspPidByteOffset*)
 - The reference to the supported information container (see configuration parameter *DcmDspPidDataSupportInfo*)
 - The *Xxx_ReadData()* function that the Dcm must call to obtain the current value of the data or the name of the port that the Dcm uses to obtain the

current value through the RTE from a SW-C (see configuration parameters `DcmDspPidDataReadFnc` and `DcmDspPidDataUsePort`)

[SWS_Dcm_00407] [On reception of an OBD Service \$01 request with only "availability PIDs" as parameter, the Dcm shall respond with the corresponding supported (=configured) PIDs encoded according to the OBD standard.]()

To obtain the value for a PID, the Dcm uses the configured `Xxx_ReadData()` functions for every data of the PID. To provide OBD Service \$01, the Dcm relies on external functions that allow it to obtain the value of the PIDs. There is one such function per data of every PID that is needed by the DCM.

When using a `Xxx_ReadData()` function, the Dcm provides a buffer of the correct length, which is filled by the function with the data PID value.

[SWS_Dcm_00408] [On reception of an OBD Service \$01 request with only PIDs that are not "availability PIDs", the Dcm shall obtain the current value of these PIDs by invoking the configured `Xxx_ReadData()` functions for every data of the PID and shall return these values as response to Service \$01.]()

[SWS_Dcm_00943] [On reception of an OBD Service \$01 request with a mixture of "availability PIDs" and not "availability PIDs", this request shall be ignored by the Dcm.]()

The entity providing the actual implementation of the `Xxx_ReadData()` function for a specific signal of a PID might be a SW-C or another basic software module. The origin of the function is not known to the Dcm but is part of the Dcm configuration. Some PIDs are provided by the DEM. These PIDs are also explicitly configured in the Dcm configuration and it is the responsibility of a correct Dcm configuration to make the `Xxx_ReadData()` function point to the correct function provided by the DEM.

[SWS_Dcm_CONSTR_6069] Dependency for `DcmDspPidDataReadFnc` [`DcmDspPidDataReadFnc` shall be only present if `DcmDspPidDataUsePort` is set to `USE_DATA_SYNCH_FNC`.]()

For certain PIDs, the Dem provides the function to obtain the PID value. Which PIDs come from the Dem are part of the Dcm configuration.

Note: For PIDs where Dem provides the function, `DcmDspPidDataUsePort` for that PID should be set to `USE_DATA_SYNCH_FNC` and `DcmDspPidDataReadFnc` shall point to the function `Dem_DcmReadDataOfPID<NN>` where <NN> represents the Id of the PID.

The data byte A of the PIDs contain the support status of the subsequent data bytes. Since not all data values might be available due to the particular vehicle configuration (e.g. there is only a Nox-sensor 1 available in the vehicle in the example above), the PID response contains in this data byte A the information about the support status of the following data values. Note, that the PIDs always contain the same number of bytes - even if not all values are really available.

[SWS_Dcm_00621] [If a `PID` contains support information (presence of `DcmDspPidDataSupportInfo` container) the `Dcm` shall add the support information in the diagnostic response.]()

[SWS_Dcm_00622] [If a `PID` contains support information (presence of `DcmDspPidDataSupportInfo` container) the `Dcm` shall calculate the support information value according to the available data for this `PID`: for every `DcmDspPidData` container existing for this `PID`, the associated support information bits, referenced in `DcmDspPidDataSupportInfo`, shall be set to one.]()

The response to the OBD-tester needs to be composed out of the available data values. Data bytes that are not provided by an `SW-C` need to be replaced with fill-byte to obtain a complete `PID` contents.

[SWS_Dcm_00623] [When responding to `OBD Service $01`, the `Dcm` shall put fill-bytes between `DcmDspPidData` in the `PID` whenever content bytes are missing in order to fit to the `PID` size (see configuration parameter `DcmDspPidSize`).]()

[SWS_Dcm_00944] [The `Dcm` shall set the fill bytes to 0x00.]()

Note: If other fill-bytes than 0x00 are needed by legislation, the application has to provide the value of the fill-byte.

[SWS_Dcm_00718] [To serialize the required AUTOSAR data types (signed- and unsigned integer) into the response message of `OBD Service $01` responses the target endianness configured in `DcmDspPidDataEndianness` shall be considered for `DcmDspPidData` elements having `DcmDspPidDataUsePort` set to `USE_DATA_SENDER_RECEIVER` or `USE_DATA_SENDER_RECEIVER_AS_SERVICE`. In case `DcmDspPidDataEndianness` is not present, the `DcmDspDataDefaultEndianness` shall be used instead.]()

[SWS_Dcm_CONSTR_6068] Dependency for `DcmDspPidDataEndianness` [In case `DcmDspPidDataEndianness` is not present, the `DcmDspDataDefaultEndianness` shall be used instead.]()

7.5.3.4 Service \$02 - Request Power Train FreezeFrame Data

[SWS_Dcm_00244] [The `Dcm` shall implement `OBD Service $02` (Request Power Train FreezeFrame Data) in compliance to all provisions of the `OBD` standard.]()

For OBD-relevant FreezeFrames AUTOSAR only supports frame 0, which is the minimum required by legislation.

[SWS_Dcm_00409] [The `Dcm` shall ignore all requests regarding record-numbers that are not 0.]()

[SWS_Dcm_00972] [On reception of an `OBD Service $02` request with a mixture of "availability `PIDs`" and not "availability `PIDs`", this request shall be ignored by the `Dcm`.]()

[SWS_Dcm_00973] [When responding to OBD Service \$02, the Dcm shall put fill-bytes between DcmDspPidData in the PID whenever content bytes are missing in order to fit to the PID size (see configuration parameter DcmDspPidSize).]()

[SWS_Dcm_00974] [The Dcm shall set the fill bytes to 0x00.]()

Note: If other fill-bytes than 0x00 are needed by legislation, the application has to provide the value of the fill-byte.

The following sections define how specific PIDs are handled by the Dcm.

7.5.3.4.1 Service \$02 - PID\$02

An external tester can request the DTC that caused a FreezeFrame to be stored by using the Service \$02 with the PID value \$02.

[SWS_Dcm_00279] [On reception of a request for Service \$02 with PID \$02, the Dcm shall call Dem_DcmGetDTCOfOBDFreezeFrame() with FrameNumber set to 0x00 to get the DTC number.](RS_Diag_04058)

The Dem module returns the corresponding DTC. Note that this 2-byte DTC is packed into the 4-byte data returned by the call to Dem_DcmGetDTCOfOBDFreezeFrame (). see Dem specification on how this is done.

[SWS_Dcm_01061] [If Dem_DcmGetDTCOfOBDFreezeFrame returns E_NOT_OK, the Dcm shall answer positively with \$0000 (indicates no stored freeze frame data).]()

7.5.3.4.2 Service \$02 - availability PID

Using Service \$02, an external tester may request the supported PIDs for a specific freeze-frame by using the "availability PIDs".

[SWS_Dcm_00284] [On reception of a service \$02 request with an "availability PID", the Dcm shall respond with the corresponding supported (=configured) PIDs encoded according to the OBD standard.]()

7.5.3.4.3 Service \$02 - other PIDs

Using Service \$02, an external tester may request the values of specific PIDs in specific FreezeFrames.

[SWS_Dcm_00286] [On reception of a service \$02 request with a PID that is not an "availability PID" and is not \$02, the Dcm shall call Dem_DcmReadDataOfOBDFreezeFrame() for every data of the PID with the following parameter values:

- PID = the PID received in the OBD request

- DestBuffer = a buffer in which the callee can write the value of the `PID`
- BufSize = the size of the DestBuffer, this must be at least equal to the size needed to store the value of the `PID` as configured in the DCM
- DataElementIndexOfPid = implicit index (from 0 to n) of the DataElement calculated by Dcm according to the order of the DataElement positions in the `PID` (see parameter `DcmDspPidByteOffset`)

]()

Note that is not necessary for the `Dcm` module to lock or unlock the record updates of the `Dem` module.

[SWS_Dcm_00287] [Upon the completion of **[SWS_Dcm_00286]**, the `Dcm` shall generate a response message including the respective `PID`, FreezeFrame Number and the associated data record for the requested FreezeFrame number.]()

[SWS_Dcm_01252] [If `Dem_DcmReadDataOfOBDFreezeFrame()` returns `E_NOT_OK` and a single `PID` is requested, the `Dcm` shall not provide any answer.]()

[SWS_Dcm_01253] [If `Dem_DcmReadDataOfOBDFreezeFrame()` returns `E_NOT_OK` and all `PIDs` from the requested multiple `PID(s)` are not supported, the `Dcm` shall not provide any answer.]()

[SWS_Dcm_01254] [If `Dem_DcmReadDataOfOBDFreezeFrame()` returns `E_NOT_OK` and at least one `PID` from the requested multiple `PID(s)` is supported, the `Dcm` shall send a positive response including the data of the supported `PID(s)`.]()

7.5.3.5 Service \$03 \$07 \$0A - Obtaining DTCs

[SWS_Dcm_00245] [The `Dcm` module shall implement `OBD` Service \$03 (Request emission-related diagnostic trouble codes) in compliance to all provisions of the `OBD` standard.]()

[SWS_Dcm_00410] [The `Dcm` module shall implement `OBD` Service \$07 (Request Emission-Related Diagnostic Trouble Codes Detected during Current or Last Completed Driving Cycle) in compliance to all provisions of the `OBD` standard.]()

[SWS_Dcm_00411] [The `Dcm` module shall implement `OBD` Service \$0A (Request Emission-Related Diagnostic Trouble Codes with Permanent Status) in compliance to all provisions of the `OBD` standard.]()

An external test tool can request an emission-related ECU to report all stored, pending or permanent emission-related DTCs by sending the request \$03, \$07, \$0A respectively.

[SWS_Dcm_00289] [When receiving a request for **OBD Service \$03**, the **Dcm** module shall obtain from the DEM all DTCs in primary memory and with a "confirmed" status using the functions `Dem_SetDTCFilter()` and `Dem_GetNextFilteredDTC().]()`

Note: The **Dcm** module can get an indication of the number of records that will be found using `Dem_GetNextFilteredDTC()` by using `Dem_GetNumberOfFilteredDTC()`. This allows the implementation to calculate the total size of the response before cycling through the DTCs.

[SWS_Dcm_00412] [When receiving a request for **OBD Service \$07**, the **Dcm** module shall obtain from the DEM module all DTCs in primary memory with a "pending" status using the functions `Dem_SetDTCFilter()` and `Dem_GetNextFilteredDTC().]()`

Note: The **Dcm** module can get an indication of the number of records that will be found using `Dem_GetNextFilteredDTC()` by using `Dem_GetNumberOfFilteredDTC()`. This allows the implementation to calculate the total size of the response before cycling through the DTCs.

[SWS_Dcm_00330] [When receiving a request for **OBD Service \$0A**, the **Dcm** module shall obtain from the DEM all DTCs stored in permanent memory using the functions `Dem_SetDTCFilter()` and `Dem_GetNextFilteredDTC().]()`

Note: The **Dcm** module can get an indication of the number of records that will be found using `Dem_GetNextFilteredDTC()` by using `Dem_GetNumberOfFilteredDTC()`. This allows the implementation to calculate the total size of the response before cycling through the DTCs.

The following table illustrates the parameters the **Dcm** module must use when calling `Dem_SetDTCFilter()` in response to a request for **OBD Service \$03**, **\$07** or **\$0A**.

Parameters to Dem_SetDTCFilter			
OBD Service	\$03	\$07	\$0A
ClientId	Client Id for this Dcm instance (see DcmDemClientRef)	Client Id for this Dcm instance (see DcmDemClientRef)	Client Id for this Dcm instance (see DcmDemClientRef)
DTCStatusMask	0x08 (confirmed bit set)	0x04 (pending bit set)	0x00
DTCFormat	DEM_DTC_FORMAT_OBD	DEM_DTC_FORMAT_OBD	DEM_DTC_FORMAT_OBD
DTCOrigin	DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY	DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY	DEM_DTC_ORIGIN_PERMANENT
FilterWithSeverity	DEM_FILTER_WITH_SEVERITY_NO	DEM_FILTER_WITH_SEVERITY_NO	DEM_FILTER_WITH_SEVERITY_NO
DTCSeverityMask	Not relevant	Not relevant	Not relevant
FilterForFaultDetectionCounter	DEM_FILTER_FOR_FDC_NO	DEM_FILTER_FOR_FDC_NO	DEM_FILTER_FOR_FDC_NO

Table 7.36: Dem_SetDTCFilter Parameters

When using paged buffer mechanism, in some case, it's possible that the number of **DTC** matching the filter change between the calculation of the total size, needed for

the first page transmission, and the sending of the further pages. For this reason, the requirement [SWS_Dcm_00587] and [SWS_Dcm_00588] shall be considered for the implementation of this service.

[SWS_Dcm_01227] [Dem_GetNextFilteredDTC returns DEM_NO_SUCH_ELEMENT and at least one matching element could be retrieved before, the Dcm shall send a positive response including these data elements and the number of DTCs.]()

[SWS_Dcm_01228] [If Dem_GetNextFilteredDTC returns DEM_NO_SUCH_ELEMENT and no matching element could be retrieved before, the Dcm shall send a positive response with the number of DTCs set to 0x00.]()

7.5.3.6 Service \$04 - Clear/reset emission-related diagnostic information

[SWS_Dcm_00246] [The Dcm module shall implement OBD Service \$04 (Clear/reset emission-related diagnostic information) in compliance to all provisions of the OBD standard.]()

An external test tool can request an emission-related ECU to clear the error memory by sending the request \$04.

[SWS_Dcm_00004] [When receiving a request for OBD Service \$04, the Dcm module shall call the interface Dem_SelectDTC with the following parameter values:

- ClientId: Client Id for this Dcm instance (see DcmDemClientRef)
- DTC = DEM_DTC_GROUP_ALL_DTCS
- DTCFormat = DEM_DTC_FORMAT_OBD
- DTCOrigin = DEM_DTC_ORIGIN_OBD_RELEVANT_MEMORY

] (*RS_Diag_04058*)

Note: this interface will always return E_OK.

[SWS_Dcm_01401] [After calling Dem_SelectDTC, the Dcm shall call the interface Dem_ClearDTC() with the following parameter value:

- ClientId: Client Id for this Dcm instance (see DcmDemClientRef)

]()

[SWS_Dcm_00413] [In case Dem_ClearDTC() returns E_OK, the Dcm module shall send a positive response.]()

[SWS_Dcm_00703] [In case Dem_ClearDTC() returns DEM_PENDING, the Dcm shall invoke Dem_ClearDTC() on next Dcm_MainFunction call again. It is up to the Dcm to send NRC 78 (ResponsePending) to respect the response behaviour.]()

[SWS_Dcm_00704] [In case Dem_ClearDTC() returns DEM_CLEAR_FAILED, the Dcm shall send a negative response 0x22 (conditionsNotCorrect).] ()

[SWS_Dcm_00967] [In case Dem_ClearDTC() returns DEM_CLEAR_BUSY, the Dcm shall send a negative response 0x22 (ConditionsNotCorrect).] ()

[SWS_Dcm_01067] [In case Dem_ClearDTC() returns DEM_CLEAR_MEMORY_ERROR, the Dcm module shall send a negative response 0x22 (ConditionNotCorrect).] ()

7.5.3.7 Service \$06 - Request On-Board Monitoring Test-results for Specific Monitored Systems

7.5.3.7.1 General requirements

[SWS_Dcm_00414] [The Dcm module shall implement OBD Service \$06 (Request On-Board Monitoring Test-results for Specific Monitored Systems) in compliance to all provisions of the OBD standard.] ()

Using Service \$06, an external test tool can request an emission-related ECU to return the DTR's associated with the OBDMID or to return the supported OBDMIDs. OBD reserves certain OBDMIDs for the special purpose of obtaining the list of supported OBDMIDs in a certain range. These OBDMIDs are called "availability OBDMIDs" and are \$00, \$20, \$40, \$60, \$80, \$A0, \$C0 and \$E0.

A tester request for supported OBDMIDs may contain up to six (6) "availability OBDMIDs".

[SWS_Dcm_00945] [On reception of an OBD Service \$06 request with "availability OBDMIDs" together with other OBDMIDs as parameter, the Dcm module shall ignore the request.] ()

[SWS_Dcm_00956] [On reception of an OBD Service \$06 request with multiple non-availability OBDMIDs, the Dcm module shall ignore the request.] ()

7.5.3.7.2 Test results obtained via Dem interaction

The maintenance of the DTRs lies within the responsibility of the DEM. SW-Cs reporting DTRs use dedicated interfaces offered by the DEM. Upon requests from the tester the Dcm retrieves the information from the DEM using dedicated DEM interfaces. There is no direct interaction between the Dcm and SW-Cs.

[SWS_Dcm_00957] [On reception of an OBD Service \$06 request with only "availability OBDMID(s)" as parameter(s), the Dcm module shall obtain the supported OBDMIDs by calling the Dem interface Dem_DcmGetAvailableOBDMIDs() for each "availability OBDMID (\$00, \$20, ...)" contained within the request and concatenate the results within the response message according to ISO-15031-5 [2].] ()

[SWS_Dcm_00958] [On reception of an **OB**D Service \$06 request with an **OB**DMID that is not an "availability **OB**DMID", the **Dcm** module shall call the DEM interface `Dem_DcmGetNumTIDsOfOBDMID()` to obtain the TIDs available for the requested **OB**DMID and then recurrently call the interface `Dem_DcmGetDTRData()` for the number of reported TIDs to obtain the associated DTR data.]()

7.5.3.8 Service \$08 - Request Control of On-Board System, Test or Component

[SWS_Dcm_00417] [The **Dcm** module shall implement **OB**D Service \$08 (Request Control of On-Board System, Test or Component) in compliance to all provisions of the **OB**D standard.]()

Using Service \$08, an external test tool can control an on-board system, test or component using a TID. **OB**D reserves certain TIDs for the special purpose of obtaining the list of supported TIDs in a certain range. These TIDs are called "availability TIDs" and are \$00, \$20, \$40, \$60, \$80, \$A0, \$C0 an \$E0.

The **Dcm** module's configuration defines the TIDs that are available on the ECU for the purpose of **OB**D Service \$08. The configuration defines for each such TID (see configuration parameter `DcmDspRequestControlTestId`):

- the name of the port the **Dcm** uses to access the `RequestControlServices` interface (see configuration parameter `DcmDspRequestControl`)
- the number of bytes this function takes as input (see configuration parameter `DcmDspRequestControlInBufferSize`)
- the number of bytes this function writes as output (see configuration parameter `DcmDspRequestControlOutBufferSize`)

To provide **OB**D Service \$08, the **Dcm** relies on external functions configured per TID.

[SWS_Dcm_00418] [On reception of an **OB**D Service \$08 request with one or more "availability TIDs" as parameter, the **Dcm** module shall respond with the corresponding supported (=configured) TIDs.]()

[SWS_Dcm_00947] [On reception of an **OB**D Service \$08 request "availability TIDs" together with other TIDs as parameter, the **Dcm** module shall ignore the request.]()

[SWS_Dcm_00419] [On reception of an **OB**D Service \$08 request with a TID that is not an "availability TID", the **Dcm** module shall invoke the configured `Xxx_RequestControl()` function with the following parameters values: `InBuffer`: data contained in the **OB**D request (the size of which must correspond to the size configured in the **Dcm** module's configuration) `OutBuffer`: space in which the `RequestControl` function can store its result (the size of the buffer is taken from the **Dcm** module's configuration)]()

[SWS_Dcm_00420] [After the execution of **[SWS_Dcm_00419]**, the **Dcm** module shall respond to the service request using the data stored by the `RequestControl` function in the `OutBuffer`.]()

[SWS_Dcm_00948] [As specified in [3, SAE J1979], unused data bytes shall be filled with \$00.] ()

[SWS_Dcm_01192] [If Xxx_RequestControl() doesn't return E_OK, the Dcm shall return NRC 0x22.] ()

7.5.3.9 Service \$09 - Request Vehicle Information

[SWS_Dcm_00421] [The Dcm module shall implement OBD Service \$09 (Request Vehicle Information) in compliance to all provisions of the OBD standard.] ()

Using Service \$09, an external test tool can request vehicle information or can obtain lists of supported vehicle information. OBD reserves certain InfoTypes for the special purpose of obtaining the list of supported InfoTypes in a certain range. These Infotypes are called "availability InfoTypes" and are \$00, \$20, \$40, \$60, \$80, \$A0, \$C0 an \$E0.

The Dcm module's configuration defines the InfoTypes and associated data that are available on one or several SW-C. The configuration defines for each such InfoType:

- The value of InfoType (see configuration parameter [DcmDspVehInfoInfoType](#))
- For every data of the InfoType:
 - The position of this data in the InfoType (see configuration parameter [DcmDspVehInfoDataOrder](#))
 - the size of the value of the InfoType data (see configuration parameter [DcmDspVehInfoDataSize](#))
 - the function that the Dcm module must call to obtain the value for this InfoType data OR the port-name through which the Dcm module can obtain the value for this InfoType data (see configuration parameter [DcmDspVehInfoDataReadFnc](#) and [DcmDspVehInfoDataUsePort](#)).

To provide OBD Service \$09, the Dcm relies on external functions that allow it to obtain the value of an InfoType data. There is one such function per InfoType data that is needed by the DCM.

When invoking a Xxx_GetInfotypeValueData() function, the Dcm module provides a buffer of the correct size in which the value of the InfoType data can be stored. The entity providing the actual implementation of the Xxx_GetInfotypeValueData() function for a specific InfoType data might be a SW-C or another basic software module. The origin of the function is part of the Dcm module's configuration.

Certain InfoTypes needed by the Dcm to provide Service \$09 are provided by the DEM. This is handled in the Dcm configuration.

[SWS_Dcm_00422] [On reception of an OBD Service \$09 request with one or more "availability InfoTypes" as parameter, the Dcm module shall respond with the corresponding supported (=configured) InfoTypes.] ()

[SWS_Dcm_00949] [On reception of an OBD Service \$09 request "availability InfoTypes" together with other InfoTypes as parameter, the Dcm module shall ignore the request.]()

[SWS_Dcm_00423] [On reception of an OBD Service \$09 request for an InfoType that is not an "availability InfoType", the Dcm module shall obtain the value of this InfoType by invoking all the configured Xxx_GetInfotypeValueData() function for every data of this InfoType and shall return the value as response to Service \$09.]()

[SWS_Dcm_00684] [In case DcmDspVehInfoNODIProvResp is set to FALSE, in addition to collect the available InfoType value contributions from the individual SW-C, the Dcm shall compute the data byte NofDataItems in the diagnostic response, which defines the number of DataItems included in one InfoType.]()

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

[SWS_Dcm_01167] [In case DcmDspVehInfoNODIProvResp is set to TRUE, the Dcm shall take over the value returned by the provider and report it as NofDataItems in the diagnostic response.]()

[SWS_Dcm_CONSTR_6045] [In case the responsibility is on provider side (DcmDspVehInfoNODIProvResp is set to TRUE), only one DcmDspVehInfoData container shall be allowed.]()

[SWS_Dcm_CONSTR_6046] [In case DcmDspVehInfoDataUsePort is set to FALSE and DcmDspVehInfoDataReadFnc is set to either Dem_DcmGetInfoTypeValue08 or Dem_DcmGetInfoTypeValue0B then DcmDspVehInfoNODIProvResp shall be set to TRUE.]()

Note : The integrator has to make sure that the buffer determined by the DcmDspVehInfoDataSize is sufficiently sized to receive the data returned by the provider of the data.

[SWS_Dcm_01191] [If Xxx_GetInfoTypeValueData() doesn't return E_OK or E_PENDING, the Dcm shall return NRC 0x12.]()

7.5.4 Interaction usecases

The Dcm shall be able to manage a jump to the bootloader / jump due to ECUReset request. Due to the diversity of possibility to realize this jump, this will be done using callout call.

7.5.4.1 Jump to Bootloader

4 different use cases have been identified for the jump to the bootloader, if all preconditions are fulfilled and assuming the 'suppressPosRspMsgIndicationBit' flag is set to 'false':

1. The application immediately sends a final positive response and then jumps to the bootloader
2. The application first sends a [NRC 0x78](#) response, then the final positive response and afterwards jumps to the bootloader
3. The application immediately jumps to the bootloader and the bootloader sends the final positive response
4. The application first sends a [NRC 0x78](#) response, then jumps to the bootloader and the bootloader sends the final positive response

Note: In case the 'suppressPosRspMsgIndicationBit' flag is set to 'true', use case '1' and use case '3' will not issue a positive response.

[SWS_Dcm_00532] [On reception of service DiagnosticSessionControl if the provided session is used to jump to OEM bootloader (parameter [DcmDspSessionForBoot](#) set to DCM_OEM_BOOT or DCM_OEM_BOOT_RESPAPP) the [Dcm](#) shall prepare the jump to the OEM bootloader (see [\[SWS_Dcm_00535\]](#)) by triggering the mode switch of ModeDeclarationGroupPrototype [DcmEcuReset](#) to JUMPTOBOOTLOADER.] ([RS_Diag_04098](#))

Note: By this mode switch the [Dcm](#) informs the BswM to prepare the jump to the bootloader.

[SWS_Dcm_00592] [On reception of service DiagnosticSessionControl if the provided session is used to jump to System Supplier bootloader (parameter [DcmDspSessionForBoot](#) set to DCM_SYS_BOOT or DCM_SYS_BOOT_RESPAPP) the [Dcm](#) shall prepare the jump to the System Supplier bootloader (see [\[SWS_Dcm_00535\]](#)) by triggering the mode switch of ModeDeclarationGroupPrototype [DcmEcuReset](#) to JUMPTOSYSSUPPLIERBOOTLOADER.] ([RS_Diag_04098](#))

Note: By this mode switch the [Dcm](#) informs the BswM to prepare the jump to the bootloader.

[SWS_Dcm_01164] [In case the service DiagnosticSessionControl implies an ECU reset, the [Dcm](#) shall ignore all further requests while that reset is being processed.] ()

[SWS_Dcm_00654] [In case the ModeDeclarationGroupPrototype [DcmEcuReset](#) is switched to mode JUMPTOBOOTLOADER or JUMPTOSYSSUPPLIERBOOTLOADER and the configuration parameter [DcmSendRespPendOnRestart](#) is set to TRUE, the [Dcm](#) shall trigger transmission of [NRC 0x78 - RCR-RP](#).] ([RS_Diag_04098](#))

Note: This final transmission of [NRC 0x78](#) before switching to Bootloader shall reload the P2* timeout in the client.

[SWS_Dcm_01175] [In case the ModeDeclarationGroupPrototype DcmEcuReset can not be switched JUMPTOBOOTLOADER or JUMPTOSYSSUPPLIERBOOTLOADER, the Dcm shall answer negatively to the request with NRC 0x22 (Conditions not correct).]

()

[SWS_Dcm_00535] [If the jump to bootloader is requested (see [SWS_Dcm_00532], [SWS_Dcm_00592], the configuration parameter DcmSendRespPendOnRestart is set to TRUE (see [SWS_Dcm_00654]) and the configuration parameter DcmDspSessionForBoot is set to DCM_OEM_BOOT or DCM_SYS_BOOT, the Dcm shall call Dcm_SetProgConditions after a successful transmission of NRC 0x78 (Response pending).] (RS_Diag_04098)

This will allow to store all relevant information prior to jumping to the bootloader.

Note: It is up to the software integrator to decide where to store that data. Usually it will be stored in non-volatile memory like e.g. data flash. It is also acceptable to "store" this data in a RAM section which is not initialized out of reset.

[SWS_Dcm_01163] [In the context of a request to jump to the bootloader (see [SWS_Dcm_00532] and [SWS_Dcm_00592]), after Dcm_SetProgConditions returns E_OK according to [SWS_Dcm_00535], the Dcm shall trigger the mode switch of the ModeDeclarationGroupPrototype DcmEcuReset to EXECUTE.] (RS_Diag_04098)

[SWS_Dcm_01177] [If the jump to bootloader is requested (see [SWS_Dcm_00532], [SWS_Dcm_00592], the configuration parameter DcmSendRespPendOnRestart is set to TRUE (see [SWS_Dcm_00654]), and the configuration parameter DcmDspSessionForBoot is set to DCM_OEM_BOOT_RESPAPP or DCM_SYS_BOOT_RESPAPP, the Dcm shall initiate the final response after a successful transmission of NRC 0x78 (Response pending).] (RS_Diag_04098)

[SWS_Dcm_00995] [If the NRC 0x78 (Response Pending) response in [SWS_Dcm_00535] is not sent successfully the Dcm shall cancel the current request.] ()

[SWS_Dcm_00997] [If the NRC 0x78 (Response Pending) response in [SWS_Dcm_00535] is not sent successfully no jump to the bootloader shall be performed.] ()

Note: If the NRC 0x78 (Response Pending) response has not been sent correctly the Dcm will stay in the application and wait for the next request from the Client.

[SWS_Dcm_01178] [In case the ModeDeclarationGroupPrototype DcmEcuReset is switched to mode JUMPTOBOOTLOADER or JUMPTOSYSSUPPLIERBOOTLOADER, the configuration parameter DcmSendRespPendOnRestart is set to FALSE and the configuration parameter DcmDspSessionForBoot is set to DCM_OEM_BOOT_RESPAPP or DCM_SYS_BOOT_RESPAPP, the Dcm shall initiate the final response.] ()

[SWS_Dcm_01179] [In case the final response has been successfully sent according to [SWS_Dcm_01177] or [SWS_Dcm_01178], the Dcm shall call Dcm_SetProgConditions.] ()

[SWS_Dcm_01180] [If `Dcm_SetProgConditions` returns `E_OK` according to [\[SWS_Dcm_01179\]](#), the `Dcm` shall trigger the mode switch of the `ModeDeclarationGroupPrototype DcmEcuReset` to `EXECUTE`.]()

[SWS_Dcm_01181] [If `Dcm_SetProgConditions` returns `E_NOT_OK` according to [\[SWS_Dcm_01179\]](#), the `Dcm` shall not request any reset, shall not perform the jump to bootloader, and shall not switch the `ModeDeclarationGroupPrototype DcmEcuReset` to `EXECUTE`.]()

[SWS_Dcm_00720] [In case the `ModeDeclarationGroupPrototype DcmEcuReset` is switched to mode `JUMPTOBOOTLOADER` or `JUMPTOSYSSUPPLIERBOOTLOADER`, the configuration parameter `DcmSendRespPendOnRestart` is set to `FALSE` and the configuration parameter `DcmDspSessionForBoot` it set to `DCM_OEM_BOOT` or `DCM_SYS_BOOT`, the `Dcm` shall call `Dcm_SetProgConditions` immediately. (see [\[SWS_Dcm_00532\]](#) and [\[SWS_Dcm_00592\]](#))]()

[SWS_Dcm_00719] [If `Dcm_SetProgConditions` returns `E_OK` according to [\[SWS_Dcm_00720\]](#), the `Dcm` shall trigger the mode switch of the `ModeDeclarationGroupPrototype DcmEcuReset` to `EXECUTE` without sending a `NRC 0x78` (Response pending).]()

In case of [\[SWS_Dcm_00719\]](#), the exact response handling depends on the state of the 'suppressPosRspMsgIndicationBit' (`TRUE` or `FALSE`) in the request message.

[SWS_Dcm_00715] [If the jump to bootloader is requested (see [\[SWS_Dcm_00532\]](#) and [\[SWS_Dcm_00592\]](#)) and if the call to `Dcm_SetProgConditions` returns `E_NOT_OK` (see [\[SWS_Dcm_00535\]](#) and [\[SWS_Dcm_00720\]](#)), no further action shall be taken by the `Dcm` and negative response `NRC 0x22` (Conditions not correct) shall be returned.]()

7.5.4.2 Jump due to ECUReset

On reception of an `ECUReset Service 0x11` request, if the configuration parameter `DcmResponseToEcuReset` is set to `AFTER_RESET`, the `Dcm` will set the `ResponseRequired` flag by calling `Dcm_SetProgConditions`.

[SWS_Dcm_01423] Answer to ECUReset request [On reception of an `ECUReset Service 0x11` request, if `DcmResponseToEcuReset` is set to `AFTER_RESET`, the `Dcm` shall answer to `EcuReset` service after the reset.] ([RS_Diag_04098](#))

[SWS_Dcm_01424] Answer to ECUReset request [On reception of an `ECUReset Service 0x11` request, if `DcmResponseToEcuReset` is set to `BEFORE_RESET`, the `Dcm` shall answer to `EcuReset` service before the reset.] ([RS_Diag_04098](#))

[SWS_Dcm_01425] Answer to ECUReset request [If the `Dcm` initiates a reset and `DcmSendRespPendOnRestart` is set to `TRUE`, the `Dcm` shall trigger transmission of `NRC 0x78` (Response pending) before the reset.] ([RS_Diag_04098](#))

7.5.4.3 Jump from Bootloader / ECUReset

[SWS_Dcm_00536] [At *Dcm* initialization, the *Dcm* shall call *Dcm_GetProgConditions* to know if the initialization is the consequence of a jump from the bootloader / ECUReset.] (*RS_Diag_04098*)

Note: It is the responsibility of the software integrator to ensure that the data contained in *Dcm_ProgConditionsType* is valid when *Dcm_Init* is called. E.g. if this data is stored in non-volatile memory, it may take some time to make it available after an ECU reset. This has to be taken into account when designing the ECU's startup process.

[SWS_Dcm_00537] [If the initialization of the *Dcm* is the consequence of a jump from the bootloader / ECUReset (see [\[SWS_Dcm_00536\]](#)), the *Dcm* shall call *ComM_DCM_ActiveDiagnostic(NetworkId)* to request the *ComManager* for the full communication mode.] ()

[SWS_Dcm_00767] [When the *ComM* reports full communication to the *Dcm*, the *Dcm* shall send the Response to the Service Id passed in the *Dcm_ProgConditionsType*.] (*RS_Diag_04098*)

[SWS_Dcm_00768] [If the initialization of the *Dcm* is the consequence of a jump from the bootloader (see [\[SWS_Dcm_00536\]](#)) and the application is updated by an FLASH download (*Dcm_ProgConditionsType.ApplUpdated* == True), the *Dcm* shall call *BswM_Dcm_ApplicationUpdated()* to notify the *BswM* that the application was updated.] ()

7.5.4.4 Flags management

7.5.4.4.1 Jump to Bootloader

[SWS_Dcm_01182] [On reception of a UDS Service 0x10 request (Diagnostic Session Control) with subfunction 0x02 (Start Programming Session), the *Dcm* shall set the *ReprogrammingRequest* flag and, if indicated for this service, the *ResponseRequired* flag by calling *Dcm_SetProgConditions*.] ()

[SWS_Dcm_01183] [Depending on configuration parameter *DcmDspSessionForBoot*, either the application shall send the positive response (if *suppressPosRspMsgIndicationBit* = FALSE) or after an ECU reset, when the bootloader is started, it shall send a response and clear the *ResponseRequired* flag. In either case, the bootloader shall clear the *ReprogrammingRequest* flag.] ()

[SWS_Dcm_01185] [In case that, during jump to Bootloader, the *Dcm_SetProgConditions* API returns *E_NOT_OK*, a DET error shall be reported *DCM_E_SET_PROG_CONDITIONS_FAIL* and normal functionality shall resume.] ()

7.5.4.4.2 Jump from Bootloader

After successful reprogramming of the application software, the bootloader will update the *ApplUpdated* flag and the *ResponseRequired* flags.

After an ECU reset, when the newly programmed application is started for the first time, the *Dcm* will read the *ApplUpdated* and *ResponseRequired* flag by calling *Dcm_GetProgConditions*. During this function call the *ApplUpdated* and *ResponseRequired* flags are cleared by the integration code.

7.6 Error notification

The Default Error Tracer module is just help for BSW development and integration. It must not be contained inside the production code. The [API](#) is defined, but the functionality can be chosen and implemented according to the development needs (e.g. errors count, send error information via a serial interface to an external logger, and so on).

7.7 Synchronous and Asynchronous implementation

The *Dcm* can access data using an R-Port requiring either a synchronous or an asynchronous *ClientServerInterface DataServices_{Data}*. In the *Dcm* SWS, the parameter *DcmDspDataUsePort* is set to *USE_DATA_SYNCH_CLIENT_SERVER* or *USE_DATA_ASYNC_CLIENT_SERVER* or *USE_DATA_ASYNC_CLIENT_SERVER_ERROR*.

In case of *USE_DATA_SYNCH_CLIENT_SERVER*, the interface shall be compatible with the *Dem* interface "DataServices_<Data>" (no *OpStatus* parameter). The parameter *OpStatus* and return parameter *DCM_E_PENDING* shall only be available in case of *USE_DATA_ASYNC_CLIENT_SERVER* or *USE_DATA_ASYNC_CLIENT_SERVER_ERROR*.

Note: a *Dcm* implementation using *AsynchronousServerCallPoint* or *SynchronousServerCallPoint* when calling service processors is completely an implementation decision. This only indicates that the operation uses the status of the operation to allow an asynchronous processing by the *SW-C* (initiating a request, checking if a request is still pending, or canceling a pending request, see [\[SWS_Dcm_00686\]](#)).

There is no correlation to the operation signature (i.e. existence of *OpStatus* parameter and *DCM_E_PENDING* return code) that demands *AsynchronousServerCallPoint* or *SynchronousServerCallPoint* usage.

[SWS_Dcm_01187] [If an asynchronous interface is used, the *Dcm* shall consider the Output data (OUT) only valid after the last call to the interface that returns *E_OK*.]()

[SWS_Dcm_01188] [If an asynchronous interface is used, the `Dcm` shall consider the OUT-parameter `ErrorCode` only valid after the last call to the interface that returns `E_NOT_OK`.]()

Note : The "last call" to the interface is the call that returns with a value that indicates that processing has finished, i.e. `E_OK` or `E_NOT_OK`.

Note : INOUT parameter are a combination of the requirements above, i.e. on each call of the interface the parameters shall have a valid in-value, and the `Dcm` considers the out-value valid only after the last call of the interface.

[SWS_Dcm_01189] [If an asynchronous interface is used, the `Dcm` shall provide the values originating from the request for the Input data (IN) on every call to the interface.]()

Note: Requirements [\[SWS_Dcm_01187\]](#), [\[SWS_Dcm_01188\]](#) and [\[SWS_Dcm_01189\]](#) do not apply for functions where a deviant behaviour is explicitly specified.

7.8 DID configuration

The configuration of the `Dcm` contains a list of supported DIDs which can be configured in two ways:

- The individual `DID` configuration, which required one connection (either via a port or a c-function) per configured data element of the respective `DID` to access to the data (reading, writing and controlling). The interface `DataServices` should be used for each `DID` in this case.
- The `DID` range configuration, used to handle a set of DIDs sharing the same behavior uniformly in one `SW-C` with only one port-connection. The interface `DataServices_DIDRange_{Range}` should be used in this case. Using this configuration allows an interface optimization. The following parameters shall be configured in order to use the `DIDRange` optimization: `DcmDspDidRangeIdentifierLowerLimit` and `DcmDspDidRangeIdentifierUpperLimit` which delimited the range of the DIDs. `DcmDspDidRangeMaxDataLength` and `DcmDspDidRangeHasGaps`.

[SWS_Dcm_01174] [If `DcmVinRef` is configured then the VIN shall be fetched once by the `Dcm` during startup by calling `Dcm_GetVin`.]()

7.8.1 Individual DID

The individual `DID` can be configured in ECUC_Dcm_00601: `DcmDspDid`. A unique `DID` identifier is configured on 2 bytes in ECUC_Dcm_00602: `DcmDspDidIdentifier`. In case the `DID` refers to other `DIDs`, the link between them can be configured

in ECUC_Dcm_00606 : [DcmDspDidRef](#)². Each [DID](#) allows to access to signal data values (by reading and/or writing). The signal reference (to [DcmDspData](#)) and the position of the data in the diagnostic answer (for reading) or request (for writing) can be configured in ECUC_Dcm_00813: [DcmDspDidSignal](#).

The configuration of the data belonging to the [DID](#) can be provided in the container [DcmDspData](#) (ECUC_Dcm_00869). This container collects the following information:

- The Data endianness of the data belonging to the [DID](#) (ECUC_Dcm_00986: [DcmDspDataEndianness](#))
- The length and the type of the data (ECUC_Dcm_00605: [DcmDspDataByteSize](#), ECUC_Dcm_00985 : [DcmDspDataType](#))
- The interface to be used to access to the data (ECUC_Dcm_00713: [DcmDspDataUsePort](#))
- The NRAM blockId to access the data (ECUC_Dcm_00809 : [DcmDspDataBlockIdRef](#))
- The interfaces to the application in order to :
 - Check if the [DID](#) can be accessed in the current conditions. This can be achieved by checking for each [DataElement](#) if the conditions to read the data are satisfied (ECUC_Dcm_00677: [DcmDspDataConditionCheckReadFnc](#) and ECUC_Dcm_00955: [DcmDspDataConditionCheckReadFncUsed](#))
 - Request to freeze the current state of an [IOControl](#) (ECUC_Dcm_00674: [DcmDspDataFreezeCurrentStateFnc](#))
 - Get the scaling information of the [DID](#). This can be achieved by getting the scaling information for each [DataElement](#) (ECUC_Dcm_00676: [DcmDspDataGetScalingInfoFnc](#))
 - Request the data length of a [DataElement](#) (ECUC_Dcm_00671: [DcmDspDataReadDataLengthFnc](#))
 - Read a certain ECU signal (ECUC_Dcm_00824: [DcmDspDataReadEcuSignal](#)).
 - Access in reading or writing to the data (ECUC_Dcm_00669 : [DcmDspDataReadFnc](#), ECUC_Dcm_00670: [DcmDspDataWriteFnc](#))
 - Request to reset an [IOControl](#) to default value (ECUC_Dcm_00673 : [DcmDspDataResetToDefaultFnc](#))
 - Request to return control to ECU of an [IOControl](#) (ECUC_Dcm_00672 : [DcmDspDataReturnControlToEcuFnc](#))

²Overview of [DcmDspDid](#) container is described in chapter [10.2.5.7.1](#)

- Request to adjust the IO signal (ECUC_Dcm_00675 : [DcmDspDataShortTermAdjustmentFnc](#))

It is also possible to configure an alternative diagnosis representation via ECUC_Dcm_00993: [DcmDspDiagnosisScaling](#).

The following example shows how to configure the containers [DcmDspDid](#) and [DcmDspData](#) for an individual [DID](#) 0xF080. This configuration allows access to a byte of data via synchronous C APIs [WriteDID_F080](#) (for writing) and [ReadDID_F080](#) (for reading).

- [DcmDspDidIdentifier](#)=0xF080
- [DcmDspDidByteOffset](#)=0
- [DcmDspDidDataRef](#)=[DcmDspData_F080](#)
- [DcmDspDataByteSize](#)=8
- [DcmDspDataType](#)=[UINT8_N](#)
- [DcmDspDataUsePort](#)=[USE_DATA_SYNCH_FNC](#)
- [DcmDspDataWriteFnc](#)=[WriteDID_F080](#)
- [DcmDspDataReadFnc](#)=[ReadDID_F080](#)

[SWS_Dcm_CONSTR_6067] Dependency for [DcmDspDataBlockIdRef](#) [[DcmDspDataBlockIdRef](#) shall be only present if [DcmDspDataUsePort](#) is set to [USE_BLOCK_ID](#).]()

7.8.2 DID ranges

[DID](#) ranges are in general the same as the 'normal' [DID](#) read and write function, except that the [DID](#) is also passed as a parameter. This allows to treat the [DID](#) range in a switch/case in the read or the write function.

The ranges can be applied for reading ([ReadDataByIdentifier](#) 0x22) and writing ([WriteDataByIdentifier](#) 0x2E) [DIDs](#).

The ranges can be configured in ECUC_Dcm_00937 : [DcmDspDidRange](#). Each configured range is by default accessible by service 0x22 and 0x2E. In case the range should be limited to reading or writing, the referenced [DcmDspDidInfo](#) container should be defined accordingly.

It is also possible to define gaps within the range ([DcmDspDidRangeHasGaps](#)). By activating this feature, the [Dcm](#) invokes each time a [DID](#) is requested within the configured range, the operation [IsDidAvailable](#) has to check the current availability. And as the [DIDs](#) of the specified range can have different length, the length of the longest [DID](#) has to be configured ([DcmDspDidRangeMaxDataLength](#)) in order to reserve enough buffer passed to the respective function.

In general, the range functionality can also be used for a single `DID` if you specifically want to pass the `DID` as a parameter. Then lower `DID` and upper `DID` should be the same.

`ReadDidRangeDataLength` operation allows to request the application to return the data length of a `DID` Range.

[SWS_Dcm_CONSTR_6020] Definition of allowed `DID` access [Any defined range shall only reference via `DcmDspDidRangeInfoRef`. The sub-containers `DcmDspDidControl` and `DcmDspDidDefineinDcmDspDidInfo` shall not be used] .]()

[SWS_Dcm_CONSTR_6021] `DID` ranges cannot be mapped on `DDIDs`, because service `0x2C` `DDID` does not support the range feature. Practically `DcmDspDidRangeIdentifierLowerLimit` and `DcmDspDidRangeIdentifierUpperLimit` should not include `DIDs` of the range `0xF200` till `0xF3FF`. [Any defined range shall only reference `DcmDspDidInfo` via `DcmDspDidRangeInfoRef`, having set `DcmDspDidDynamicallyDefined` == `False`.]()

7.9 Startup behavior

[SWS_Dcm_00334] [`Dcm_Init` shall initialize all `Dcm` global variables with the values of the configuration]()

8 API specification

This section defines:

- The syntax and semantics of the functions that are provided and required from other BSW modules. These take the form of "C"-APIs.
- The syntax and semantics of a subset of those functions which are used by software-components through the RTE. These take the form of descriptions using the concepts of the Software-Component Template.

8.1 Imported types

In this chapter all types included from the following files are listed.

[SWS_Dcm_00333] [

<i>Module</i>	<i>Header File</i>	<i>Imported Type</i>
BndM	bndm.h	BndM_BlockIdType
	bndm.h	BndM_ResultType
	bndm_externals.h	BndM_Block{BlockId.Shortname}Type
ComStack_Types	ComStack_Types.h	BufReq_ReturnType
	ComStack_Types.h	NetworkHandleType
	ComStack_Types.h	PduIdType
	ComStack_Types.h	PduInfoType
	ComStack_Types.h	PduLengthType
	ComStack_Types.h	RetryInfoType
	ComStack_Types.h	TpDataStateType
Csm	Rte_Csm_Type.h	Csm_ResultType
Dem	Dem.h	Dem_DTCRequestType
	Dem.h	Dem_DTCSeverityType
	Dem.h	Dem_DTCTranslationFormatType
	Rte_Dem_Type.h	Dem_DTCFormatType
	Rte_Dem_Type.h	Dem_DTCOriginType
	Rte_Dem_Type.h	Dem_UdsStatusByteType
KeyM	Rte_KeyM_Type.h	KeyM_ResultType
NvM	Rte_NvM_Type.h	NvM_BlockIdType
SchM	SchM.h	SchM_ReturnType
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType

]()

8.2 Type definitions

The [Dcm](#) module shall ensure that implementation-specific types are not "visible" outside of Dcm. Otherwise, the complete architecture would be corrupted.

This section lists the types which are defined by the [Dcm](#) SWS.

8.2.1 Dcm_StatusType

[SWS_Dcm_00976] [

Name	Dcm_StatusType		
Kind	Type		
Derived from	uint8		
Range	DCM_E_OK	0x00	This value is representing a successful operation.
	DCM_E_ROE_NOT_ACCEPTED	0x06	ResponseOnOneEvent request is not accepted by DCM (e.g. old ResponseOnOneEvent is not finished) (used at API: Dcm_ResponseOnOneEvent())
	DCM_E_PERIODICID_NOT_ACCEPTED	0x07	Periodic transmission request is not accepted by DCM (e.g. old Periodic transmission is not finished) (used at API: Dcm_ResponseOnOneDataByPeriodicId ())
Description	Base item type to transport status information.		
Available via	Dcm.h		

]()

8.2.2 Dcm_CommunicationModeType

[SWS_Dcm_00981] [

Name	Dcm_CommunicationModeType		
Kind	Type		
Derived from	uint8		
Range	DCM_ENABLE_RX_TX_NORM	0x00	Enable the Rx and Tx for normal communication
	DCM_ENABLE_RX_DISABLE_TX_NORM	0x01	Enable the Rx and disable the Tx for normal communication
	DCM_DISABLE_RX_ENABLE_TX_NORM	0x02	Disable the Rx and enable the Tx for normal communication
	DCM_DISABLE_RX_TX_NORMAL	0x03	Disable Rx and Tx for normal communication
	DCM_ENABLE_RX_TX_NM	0x04	Enable the Rx and Tx for network management communication
	DCM_ENABLE_RX_DISABLE_TX_NM	0x05	Enable Rx and disable the Tx for network management communication
	DCM_DISABLE_RX_ENABLE_TX_NM	0x06	Disable the Rx and enable the Tx for network management communication
	DCM_DISABLE_RX_TX_NM	0x07	Disable Rx and Tx for network management communication
	DCM_ENABLE_RX_TX_NORM_NM	0x08	Enable Rx and Tx for normal and network management communication





	DCM_ENABLE_RX_DISABLE_TX_NORM_NM	0x09	Enable the Rx and disable the Tx for normal and network management communication
	DCM_DISABLE_RX_ENABLE_TX_NORM_NM	0x0A	Disable the Rx and enable the Tx for normal and network management communication
	DCM_DISABLE_RX_TX_NORM_NM	0x0B	Disable Rx and Tx for normal and network management communication
Description	-		
Available via	Dcm.h		

⌋()

8.2.3 Dcm_ConfigType

[SWS_Dcm_00982] ⌈

Name	Dcm_ConfigType		
Kind	Structure		
Elements	Implementation specific		
	Type	-	
	Comment	-	
Description	This type defines a data structure for the post build parameters of the DCM . At initialization the DCM gets a pointer to a structure of this type to get access to its configuration data, which is necessary for initialization.		
Available via	Dcm.h		

⌋()

8.2.4 Dcm_ReturnReadMemoryType

[SWS_Dcm_00985] ⌈

Name	Dcm_ReturnReadMemoryType		
Kind	Type		
Derived from	uint8		
Range	DCM_READ_OK	0x00	Reading has been done
	DCM_READ_PENDING	0x01	Reading is pending, another call is request to finalize the reading
	DCM_READ_FAILED	0x02	Reading has failed



△

	DCM_READ_FORCE_RCRRP	0x03	Reading is pending, the Response pending transmission starts immediately
Description	Return values of Callout Dcm_ReadMemory		
Available via	Dcm.h		

]()

8.2.5 Dcm_ReturnWriteMemoryType

[SWS_Dcm_00986] [

Name	Dcm_ReturnWriteMemoryType		
Kind	Type		
Derived from	uint8		
Range	DCM_WRITE_OK	0x00	Writing has been done
	DCM_WRITE_PENDING	0x01	Writing is pending, another called is requested
	DCM_WRITE_FAILED	0x02	The writing has failed
	DCM_WRITE_FORCE_RCRRP	0x03	Writing is pending, the Response pending transmission starts immediately
Description	Return type of callout Dcm_WriteMemory		
Available via	Dcm.h		

]()

8.2.6 Dcm_EcuStartModeType

[SWS_Dcm_00987] [

Name	Dcm_EcuStartModeType		
Kind	Type		
Derived from	uint8		
Range	DCM_COLD_START	0x00	The ECU starts normally
	DCM_WARM_START	0x01	The ECU starts from a bootloader jump
Description	Allows the DCM to know if a diagnostic response shall be sent in the case of a jump from bootloader		
Available via	Dcm.h		

]()

8.2.7 Dcm_ProgConditionsType

[SWS_Dcm_00988] [

Name	Dcm_ProgConditionsType	
Kind	Structure	
Elements	ConnectionId	
	Type	uint16
	Comment	Unique id of the connection on which the request has been received
	TesterAddress	
	Type	uint16
	Comment	Source address of the received request if meta data is enabled, otherwise the value as configured in DcmDslProtocolRxTesterSource Addr
	Sid	
	Type	uint8
	Comment	Service identifier of the received request
	SubFnclId	
	Type	uint8
	Comment	Identifier of the received subfonction
	ReprogrammingRequest	
	Type	boolean
	Comment	Set to true in order to request reprogramming of the ECU.
	ApplUpdated	
Type	boolean	
Comment	Indicate whether the application has been updated or not.	
ResponseRequired		
Type	boolean	
Comment	Set to true in case the flashloader or application shall send a response.	
Description	Used in Dcm_SetProgConditions() to allow the integrator to store relevant information prior to jumping to bootloader / jump due to ECUReset request.	
Available via	Dcm.h	

]()

8.2.8 Dcm_MsgItemtype

[SWS_Dcm_00989] [

Name	Dcm_MsgItemtype
Kind	Type
Derived from	uint8



△

Description	Base type for diagnostic message item
Available via	Dcm.h

]()

8.2.9 Dcm_MsgType

[SWS_Dcm_00990] [

Name	Dcm_MsgType
Kind	Pointer
Type	Dcm_MsgItemType*
Description	Base type for diagnostic message (request, positive or negative response)
Available via	Dcm.h

]()

8.2.10 Dcm_MsgLenType

[SWS_Dcm_00991] [

Name	Dcm_MsgLenType
Kind	Type
Derived from	uint32
Description	Length of diagnostic message (request, positive or negative response). The maximum length is dependent of the underlying transport protocol/media.
Available via	Dcm.h

]()

8.2.11 Dcm_MsgAddInfoType

Please note that the following table describes a struct type definition - including its struct items "elements".

[SWS_Dcm_00992] [

Name	Dcm_MsgAddInfoType	
Kind	Structure	
Elements	reqType	
	Type	bit
	Comment	(Pos LSB+0) 0 = physical request 1 = functional request
	suppressPosResponse	
	Type	bit
	Comment	Position LSB+1 0 = no (do not suppress) 1 = yes (no positive response will be sent)
Description	Additional information on message request. Datastructure: Bitfield	
Available via	Dcm.h	

]()

8.2.12 Dcm_IdContextType

[SWS_Dcm_00993] [

Name	Dcm_IdContextType
Kind	Type
Derived from	uint8
Description	This message context identifier can be used to determine the relation between request and response confirmation.
Available via	Dcm.h

]()

8.2.13 Dcm_MsgContextType

Please note that the following table describes a struct type definition - including its struct items "elements".

[SWS_Dcm_00994] [

Name	Dcm_MsgContextType	
Kind	Structure	
Elements	reqData	
	Type	Dcm_MsgType
	Comment	Request data, starting directly after service identifier (which is not part of this data)
	reqDataLen	
	Type	Dcm_MsgLenType
	Comment	Request data length (excluding service identifier)
	resData	
	Type	Dcm_MsgType
	Comment	Positive response data, starting directly after service identifier (which is not part of this data).
	resDataLen	
	Type	Dcm_MsgLenType
	Comment	Positive response data length (excluding service identifier)
	msgAddInfo	
	Type	Dcm_MsgAddInfoType
	Comment	Additional information about service request and response (see: Dcm_MsgAddInfo)
	resMaxDataLen	
	Type	Dcm_MsgLenType
Comment	The maximal length of a response is restricted by the size of the buffer. The buffer size can depend on the diagnostic protocol identifier which is assigned to this message, e. g. an OBD protocol id can obtain other properties than the enhanced diagnostic protocol id. The resMaxDataLen is a property of the diagnostic protocol assigned by the DSL. The value does not change during communication. It cannot be implemented as a constant, because it can differ between different diagnostic protocols.	
idContext		
Type	Dcm_IdContextType	
Comment	This message context identifier can be used to determine the relation between request and response confirmation. This identifier can be stored within the application at request time, so that the response can be assigned to the original request. Background: Within the confirmation, the message context is no more valid, all message data is lost. You need an additional information to determine the request to which this confirmation belongs.	
dcmRxPduld:		
Type	PduldType	
Comment	Pdu identifier on which the request was received. The Pduld of the request can have consequences for message processing. E. g. an OBD request will be received on the OBD Pduld and will be processed slightly different than an enhanced diagnostic request received on the physical	
Description	This data structure contains all information which is necessary to process a diagnostic message from request to response and response confirmation.	
Available via	Dcm.h	

10

8.2.14 Dcm_ExtendedOpStatusType

[SWS_Dcm_91015] [

Name	Dcm_ExtendedOpStatusType		
Kind	Type		
Derived from	uint8		
Range	DCM_INITIAL	0x00	Indicates the initial call to the operation
	DCM_PENDING	0x01	Indicates that a pending return has been done on the previous call of the operation
	DCM_CANCEL	0x02	Indicates that the Dcm requests to cancel the pending operation
	DCM_FORCE_RCRRP_OK	0x03	Confirm a response pending transmission Variation
	DCM_POS_RESPONSE_SENT	0x04	Indicates that a positive response has been sent successfully
	DCM_POS_RESPONSE_FAILED	0x05	Indicates that a positive response has not been sent successfully
	DCM_NEG_RESPONSE_SENT	0x06	Indicates that a negative response has been sent successfully
	DCM_NEG_RESPONSE_FAILED	0x07	Indicates that a negative response has not been sent successfully
Description	–		
Available via	Dcm.h		

]()

8.3 Function definitions

This section defines the functions provided for other modules.

8.3.1 Functions provided for other BSW components

8.3.1.1 Dcm_Init

[SWS_Dcm_00037] [

Service Name	Dcm_Init
Syntax	<pre>void Dcm_Init (const Dcm_ConfigType* ConfigPtr)</pre>
Service ID [hex]	0x01





Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ConfigPtr	Pointer to configuration set in Variant Post-Build.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Service for basic initialization of DCM module.	
Available via	Dcm.h	

]([SRS_BSW_00438](#), [SRS_BSW_00101](#))

The call of this service is mandatory before using the [Dcm](#) module for further processing.

8.3.1.2 Dcm_GetVersionInfo

[SWS_Dcm_00065] [

Service Name	Dcm_GetVersionInfo	
Syntax	<pre>void Dcm_GetVersionInfo (Std_VersionInfoType* versionInfo)</pre>	
Service ID [hex]	0x24	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	versionInfo	Pointer to where to store the version information of this module.
Return value	None	
Description	Returns the version information of this module	
Available via	Dcm.h	

]([SRS_BSW_00407](#), [SRS_BSW_00482](#), [SRS_BSW_00003](#))

8.3.1.3 Dcm_DemTriggerOnDTCStatus

[SWS_Dcm_00614] [

Service Name	Dcm_DemTriggerOnDTCStatus	
Syntax	<pre>Std_ReturnType Dcm_DemTriggerOnDTCStatus (uint32 DTC, Dem_UdsStatusByteType DTCStatusOld, Dem_UdsStatusByteType DTCStatusNew)</pre>	
Service ID [hex]	0x2B	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	DTC	This is the DTC the change trigger is assigned to.
	DTCStatusOld	DTC status before change
	DTCStatusNew	DTC status after change
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: this value is always returned.
	Description	
Description		Triggers on changes of the UDS status byte. Allows to trigger on ROE Event for subservice OnDTCStatusChanged.
Available via	Dcm_Dem.h	

}]()

8.3.1.4 Dcm_GetVin

[SWS_Dcm_00950] [

Service Name	Dcm_GetVin	
Syntax	<pre>Std_ReturnType Dcm_GetVin (uint8* Data)</pre>	
Service ID [hex]	0x07	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	Data	Pointer to where to store the VIN
	Return value	E_OK: The Data pointer has been filled with valid VIN E_NOT_OK: The default VIN will be used in the DoIP
Description		Function to get the VIN (as defined in SAE J1979-DA)
Available via	Dcm.h	

}]()

Note: After fetching the VIN, the [Dcm](#) can offer the data to all users without worrying that the data is unavailable if a user asks for it. This is necessary because the VIN could not be fetched synchronously for all settings of DcmDspDidDataUsePort.

8.3.1.5 Dcm_BndMWriteBlockFinish

[SWS_Dcm_91088]{DRAFT} [

Service Name	Dcm_BndMWriteBlockFinish (draft)	
Syntax	<pre>void Dcm_BndMWriteBlockFinish (BndM_BlockIdType BlockId, BndM_ResultType result)</pre>	
Service ID [hex]	0x7a	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	BlockId	–
	result	–
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Called by BndM to indicate that a block write operation has finished. Tags: atp.Status=draft	
Available via	Dcm.h	

] ([RS_Diag_04243](#))

8.3.2 Functions provided to BSW modules and to SW-Cs

The functions defined in this section can also be used by SW-Cs through the RTE.

8.3.2.1 Dcm_SetDeauthenticatedRole

[SWS_Dcm_91069] [

Service Name	Dcm_SetDeauthenticatedRole	
Syntax	<pre>Std_ReturnType Dcm_SetDeauthenticatedRole (uint16 connectionId, Dcm_AuthenticationRoleType deauthenticatedRole)</pre>	
Service ID [hex]	0x79	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	connectionId	Unique connection identifier identifying the connection for which a deauthenticated roles is set.
	deauthenticatedRole	New deauthenticated role that is assigned to that connection
Parameters (inout)	None	





Parameters (out)	None	
Return value	Std_ReturnType	E_OK: this value is always returned.
Description	Sets a new role used in deauthenticated state for that connection. The set role is valid until the connection switches into authenticated state or the ECU is reset.	
Available via	Dcm.h	

]()

8.3.2.2 Dcm_GetSecurityLevel

[SWS_Dcm_00338] [

Service Name	Dcm_GetSecurityLevel	
Syntax	Std_ReturnType Dcm_GetSecurityLevel (Dcm_SecLevelType* SecLevel)	
Service ID [hex]	0x0d	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	SecLevel	Active Security Level value Conversion formula to calculate SecurityLevel out of tester requested SecurityAccessType parameter: $SecurityLevel = (SecurityAccessType + 1) / 2$ Content of SecurityAccessType is according to "securityAccessType" parameter of SecurityAccess request (see [11])
Return value	Std_ReturnType	E_OK: this value is always returned.
Description	This function provides the active security level value.	
Available via	Dcm.h	

] ([RS_Diag_04005](#), [RS_Diag_04011](#))

8.3.2.3 Dcm_GetSesCtrlType

[SWS_Dcm_00339] [

Service Name	Dcm_GetSesCtrlType	
Syntax	Std_ReturnType Dcm_GetSesCtrlType (Dcm_SesCtrlType* SesCtrlType)	
Service ID [hex]	0x06	



△

Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	SesCtrlType	Active Session Control Type value Content is according to "diagnosticSessionType" parameter of DiagnosticSessionControl request (see [11])
Return value	Std_ReturnType	E_OK: this value is always returned.
Description	This function provides the active session control type value.	
Available via	Dcm.h	

](RS_Diag_04006, RS_Diag_04011)

8.3.2.4 Dcm_GetActiveProtocol

[SWS_Dcm_00340] [

Service Name	Dcm_GetActiveProtocol	
Syntax	Std_ReturnType Dcm_GetActiveProtocol (Dcm_ProtocolType* ActiveProtocolType, uint16* ConnectionId, uint16* TesterSourceAddress)	
Service ID [hex]	0x0f	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	ActiveProtocolType	Active protocol type value
	ConnectionId	Unique connection identifier
	TesterSourceAddress	source address of the tester
Return value	Std_ReturnType	E_OK: this value is always returned.
Description	This function returns the active UDS protocol details	
Available via	Dcm.h	

](RS_Diag_04011)

8.3.2.5 Dcm_ResetToDefaultSession

[SWS_Dcm_00520] [

Service Name	Dcm_ResetToDefaultSession	
Syntax	Std_ReturnType Dcm_ResetToDefaultSession (void)	
Service ID [hex]	0x2a	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: this value is always returned.
Description	The call to this function allows the application to reset the current session to Default session. Example: Automatic termination of an extended diagnostic session upon exceeding of a speed limit.	
Available via	Dcm.h	

}]()

8.3.2.6 Dcm_TriggerOnEvent

[SWS_Dcm_00521] [

Service Name	Dcm_TriggerOnEvent	
Syntax	Std_ReturnType Dcm_TriggerOnEvent (uint8 RoeEventId)	
Service ID [hex]	0x2D	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	RoeEventId	Identifier of the event that is triggered
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: RoeEventId value is valid E_NOT_OK: RoeEventId value is not valid
Description	The call to this function allows to trigger an event linked to a ResponseOnEvent request. On the function call, the DCM will execute the associated service if the corresponding Mode of the Roe EventId is 'ROE started'.	
Available via	Dcm.h	

}]()

8.3.2.7 Dcm_SetActiveDiagnostic

[SWS_Dcm_01068] [

Service Name	Dcm_SetActiveDiagnostic	
Syntax	Std_ReturnType Dcm_SetActiveDiagnostic (boolean active)	
Service ID [hex]	0x56	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	active	If false Dcm shall not call ComM_DCM_ActiveDiagnostic(). If true Dcm will call ComM_DCM_ActiveDiagnostic().
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: this value is always returned.
Description	Allows to activate and deactivate the call of ComM_DCM_ActiveDiagnostic() function.	
Available via	Dcm.h	

}]()

8.4 Callback notifications

This section defines the functions provided for lower layer BSW modules.

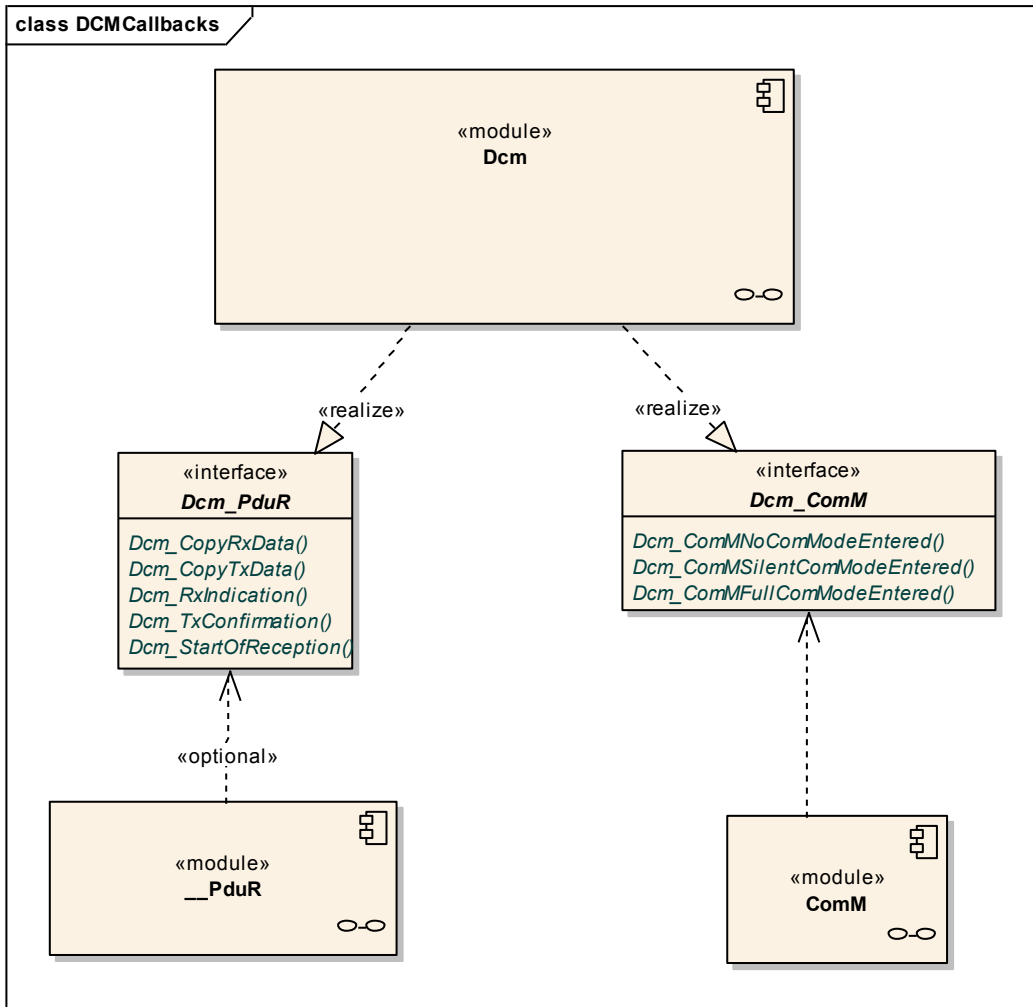


Figure 8.1: Overview of the callbacks provided by the DCM

8.4.1 Dcm_StartOfReception

[SWS_Dcm_00094] [

Service Name	Dcm_StartOfReception
Syntax	BufReq_ReturnType Dcm_StartOfReception (PduIdType id, const PduInfoType* info, PduLengthType TpSduLength, PduLengthType* bufferSizePtr)
Service ID [hex]	0x46
Sync/Async	Synchronous





Reentrancy	Reentrant	
Parameters (in)	id	Identification of the I-PDU.
	info	Pointer to a PduInfoType structure containing the payload data (without protocol information) and payload length of the first frame or single frame of a transport protocol I-PDU reception, and the MetaData related to this PDU. If neither first/single frame data nor MetaData are available, this parameter is set to NULL_PTR.
	TpSduLength	Total length of the N-SDU to be received.
Parameters (inout)	None	
Parameters (out)	bufferSizePtr	Available receive buffer in the receiving module. This parameter will be used to compute the Block Size (BS) in the transport protocol module.
Return value	BufReq_ReturnType	BUFREQ_OK: Connection has been accepted. bufferSizePtr indicates the available receive buffer; reception is continued. If no buffer of the requested size is available, a receive buffer size of 0 shall be indicated by bufferSizePtr. BUFREQ_E_NOT_OK: Connection has been rejected; reception is aborted. bufferSizePtr remains unchanged. BUFREQ_E_OVFL: No buffer of the required length can be provided; reception is aborted. bufferSizePtr remains unchanged.
Description	This function is called at the start of receiving an N-SDU. The N-SDU might be fragmented into multiple N-PDUs (FF with one or more following CFs) or might consist of a single N-PDU (SF). The service shall provide the currently available maximum buffer size when invoked with TpSdu Length equal to 0.	
Available via	Dcm.h	

]()

By the function `Dcm_StartOfReception` the receiver (e.g. DCM) is also informed implicitly about a first frame reception or a single frame reception. If the function `Dcm_StartOfReception` returns a return value not equal to `BUFREQ_OK`, the values of the out parameters are not specified and should not be evaluated by the caller.

8.4.2 Dcm_CopyRxData

[SWS_Dcm_00556] [

Service Name	Dcm_CopyRxData	
Syntax	<pre>BufReq_ReturnType Dcm_CopyRxData (PduIdType id, const PduInfoType* info, PduLengthType* bufferSizePtr)</pre>	
Service ID [hex]	0x44	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	id	Identification of the received I-PDU.



△

	info	Provides the source buffer (SduDataPtr) and the number of bytes to be copied (SduLength). An SduLength of 0 can be used to query the current amount of available buffer in the upper layer module. In this case, the SduDataPtr may be a NULL_PTR.
Parameters (inout)	None	
Parameters (out)	bufferSizePtr	Available receive buffer after data has been copied.
Return value	BufReq_ReturnType	BUFREQ_OK: Data copied successfully BUFREQ_E_NOT_OK: Data was not copied because an error occurred.
Description	This function is called to provide the received data of an I-PDU segment (N-PDU) to the upper layer. Each call to this function provides the next part of the I-PDU data. The size of the remaining buffer is written to the position indicated by bufferSizePtr.	
Available via	Dcm.h	

]()

8.4.3 Dcm_TpRxIndication

[SWS_Dcm_00093] [

Service Name	Dcm_TpRxIndication	
Syntax	<pre>void Dcm_TpRxIndication (PduIdType id, Std_ReturnType result)</pre>	
Service ID [hex]	0x45	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	id	Identification of the received I-PDU.
	result	Result of the reception.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Called after an I-PDU has been received via the TP API, the result indicates whether the transmission was successful or not.	
Available via	Dcm.h	

]()

8.4.4 Dcm_CopyTxData

[SWS_Dcm_00092] [

Service Name	Dcm_CopyTxData	
Syntax	<pre>BufReq_ReturnType Dcm_CopyTxData (PduIdType id, const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)</pre>	
Service ID [hex]	0x43	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	id	Identification of the transmitted I-PDU.
	info	Provides the destination buffer (SduDataPtr) and the number of bytes to be copied (SduLength). If not enough transmit data is available, no data is copied by the upper layer module and BUFREQ_E_BUSY is returned. The lower layer module may retry the call. An SduLength of 0 can be used to indicate state changes in the retry parameter or to query the current amount of available data in the upper layer module. In this case, the SduDataPtr may be a NULL_PTR.
	retry	<p>This parameter is used to acknowledge transmitted data or to retransmit data after transmission problems.</p> <p>If the retry parameter is a NULL_PTR, it indicates that the transmit data can be removed from the buffer immediately after it has been copied. Otherwise, the retry parameter must point to a valid RetryInfoType element.</p> <p>If TpDataState indicates TP_CONFPENDING, the previously copied data must remain in the TP buffer to be available for error recovery. TP_DATACONF indicates that all data that has been copied before this call is confirmed and can be removed from the TP buffer. Data copied by this API call is excluded and will be confirmed later. TP_DATARETRY indicates that this API call shall copy previously copied data in order to recover from an error. In this case TxTpDataCnt specifies the offset in bytes from the current data copy position.</p>
Parameters (inout)	None	
Parameters (out)	availableDataPtr	Indicates the remaining number of bytes that are available in the upper layer module's Tx buffer. availableDataPtr can be used by TP modules that support dynamic payload lengths (e.g. FrlsoTp) to determine the size of the following CFs.
Return value	BufReq_ReturnType	<p>BUFREQ_OK: Data has been copied to the transmit buffer completely as requested.</p> <p>BUFREQ_E_BUSY: Request could not be fulfilled, because the required amount of Tx data is not available. The lower layer module may retry this call later on. No data has been copied.</p> <p>BUFREQ_E_NOT_OK: Data has not been copied. Request failed.</p>
Description	This function is called to acquire the transmit data of an I-PDU segment (N-PDU). Each call to this function provides the next part of the I-PDU data unless retry->TpDataState is TP_DATARETRY. In this case the function restarts to copy the data beginning at the offset from the current position indicated by retry->TxTpDataCnt. The size of the remaining data is written to the position indicated by availableDataPtr.	
Available via	Dcm.h	

]()

8.4.5 Dcm_TpTxConfirmation

[SWS_Dcm_00351] [

Service Name	Dcm_TpTxConfirmation	
Syntax	<pre>void Dcm_TpTxConfirmation (PduIdType id, Std_ReturnType result)</pre>	
Service ID [hex]	0x48	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	id	Identification of the transmitted I-PDU.
	result	Result of the transmission of the I-PDU.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	This function is called after the I-PDU has been transmitted on its network, the result indicates whether the transmission was successful or not.	
Available via	Dcm.h	

]()

8.4.6 Dcm_TxConfirmation

[SWS_Dcm_01092] [

Service Name	Dcm_TxConfirmation	
Syntax	<pre>void Dcm_TxConfirmation (PduIdType TxPduId, Std_ReturnType result)</pre>	
Service ID [hex]	0x40	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	TxPduId	ID of the PDU that has been transmitted.
	result	E_OK: The PDU was transmitted. E_NOT_OK: Transmission of the PDU failed.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU.	
Available via	Dcm.h	

]()

8.4.7 Dcm_ComM_NoComModeEntered

[SWS_Dcm_00356] [

Service Name	Dcm_ComM_NoComModeEntered	
Syntax	<pre>void Dcm_ComM_NoComModeEntered (uint8 NetworkId)</pre>	
Service ID [hex]	0x21	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	NetworkId	Identifier of the network concerned by the mode change
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	This call informs the Dcm module about a ComM mode change to COMM_NO_COMMUNICATION.	
Available via	Dcm_ComM.h	

]()

8.4.8 Dcm_ComM_SilentComModeEntered

[SWS_Dcm_00358] [

Service Name	Dcm_ComM_SilentComModeEntered	
Syntax	<pre>void Dcm_ComM_SilentComModeEntered (uint8 NetworkId)</pre>	
Service ID [hex]	0x22	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	NetworkId	Identifier of the network concerned by the mode change
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	This call informs the Dcm module about a ComM mode change to COMM_SILENT_COMMUNICATION.	
Available via	Dcm_ComM.h	

]()

8.4.9 Dcm_ComM_FullComModeEntered

[SWS_Dcm_00360] [

Service Name	Dcm_ComM_FullComModeEntered	
Syntax	<pre>void Dcm_ComM_FullComModeEntered (uint8 NetworkId)</pre>	
Service ID [hex]	0x23	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	NetworkId	Identifier of the network concerned by the mode change
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	This call informs the Dcm module about a ComM mode change to COMM_FULL_COMMUNICATION.	
Available via	Dcm_ComM.h	

}]()

8.4.10 Dcm_CsmAsyncJobFinished

[SWS_Dcm_91076] [

Service Name	Dcm_CsmAsyncJobFinished	
Syntax	<pre>Std_ReturnType Dcm_CsmAsyncJobFinished (Csm_ResultType result)</pre>	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	result	Return value of the asynchronous job
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: this value is always returned.
Description	Can be called from Csm upon finishing an asynchronous job processing. The integrator will configure this name as callback function within the Csm ECUC configuration for asynchronous jobs. Only one such callback is available, the Dcm detects the job that has finished by evaluating the job parameter. API availability: This API will be available only if ({ecuc(Dcm/DcmDsp/DcmDspAuthentication/DcmDspAuthenticationConnection) != null}).	
Available via		

}]()

8.4.11 Dcm_KeyMAsyncCertificateVerifyFinished

[SWS_Dcm_91077] [

Service Name	Dcm_KeyMAsyncCertificateVerifyFinished	
Syntax	<pre>Std_ReturnType Dcm_KeyMAsyncCertificateVerifyFinished (KeyM_ResultType result, uint32 certId)</pre>	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	result	Return value of the asynchronous job
	certId	Certificate identifier that has finished the verification
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: this value is always returned
Description	<p>Can be called from Key upon finishing an asynchronous certificate verification. The integrator will configure this name as callback function within the KeyM ECUC configuration for asynchronous jobs. Only one such callback is available, the Dcm detects the certificate that has finished by evaluating the certId parameter. API availability: This API will be available only if ((ecuc(Dcm/DcmDsp/DcmDspAuthentication/DcmDspAuthenticationConnection) != null).</p>	
Available via		

]()

8.5 Callout Definitions

Callouts are pieces of code that have to be added to the [Dcm](#) during ECU integration. The content of most callouts is hand-written code, for some callouts the [Dcm](#) configuration tool shall generate a default implementation that is manually edited by the integrator. Conceptually, these callouts belong to the ECU Firmware.

Since callouts are no services of the [Dcm](#) they do not have an assigned Service ID. Note: The Autosar architecture doesn't provide the possibility to access the ECU memory using a physical address. This realized using BlockId which identified a memory block.

According to that, the [Dcm](#) is not able to fully support the implementation of ISO14229-1 [1]services which request a physical memory access. Therefore, the [Dcm](#) define callout to realize this kind of memory access. This callout implementation could be simply realized by defining a mapping between the BlockId and the physical memory address.

8.5.1 Dcm_ReadMemory

[SWS_Dcm_00539] [

Service Name	Dcm_ReadMemory	
Syntax	<pre>Dcm_ReturnReadMemoryType Dcm_ReadMemory (Dcm_OpStatusType OpStatus, uint8 MemoryIdentifier, uint32 MemoryAddress, uint32 MemorySize, uint8* MemoryData, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x26	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL DCM_PENDING DCM_CANCEL DCM_FORCE_RCRRP_OK
	MemoryIdentifier	Identifier of the Memory Block (e.g. used if memory section distinguishing is needed) Note: If it's not used this parameter shall be set to 0.
	MemoryAddress	Starting address of server memory from which data is to be retrieved.
	MemorySize	Number of bytes in the MemoryData
Parameters (inout)	None	
Parameters (out)	MemoryData	Data read (Points to the diagnostic buffer in DCM)
	ErrorCode	If the operation Dcm_ReadMemory returns value DCM_READ_FAILED, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Dcm_ReturnReadMemoryType	DCM_READ_OK: read was successful DCM_READ_FAILED: read was not successful DCM_READ_PENDING: read is not yet finished DCM_READ_FORCE_RCRRP: reading is pending, the Response pending transmission starts immediately
Description	<p>The Dcm_ReadMemory callout is used to request memory data identified by the parameter memoryAddress and memorySize from the UDS request message. This service is needed for the implementation of UDS services:</p> <ul style="list-style-type: none"> • ReadMemoryByAddress • RequestUpload • ReadDataByIdentifier (in case of Dynamical DID defined by memory address) • TransferData 	
Available via	Dcm_Externals.h	

]()

8.5.2 Dcm_WriteMemory

[SWS_Dcm_00540] [

Service Name	Dcm_WriteMemory	
Syntax	<pre>Dcm_ReturnWriteMemoryType Dcm_WriteMemory (Dcm_OpStatusType OpStatus, uint8 MemoryIdentifier, uint32 MemoryAddress, uint32 MemorySize, const uint8* MemoryData, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x27	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL DCM_PENDING DCM_CANCEL DCM_FORCE_RCRRP_OK
	MemoryIdentifier	Identifier of the Memory Block (e.g. used by WriteDataByIdentifier service). Note: If it's not used this parameter shall be set to 0.
	MemoryAddress	Starting address of server memory in which data is to be copied. Note: If it's not used (e.g. if the data is compressed) this parameter shall be set to 0.
	MemorySize	Number of bytes in MemoryData
	MemoryData	Data to write (Points to the diagnostic buffer in DCM)
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Dcm_WriteMemory returns value DCM_WRITE_FAILED, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Dcm_ReturnWriteMemoryType	DCM_WRITE_OK: write was successful DCM_WRITE_FAILED: write was not successful DCM_WRITE_PENDING: write is not yet finished DCM_WRITE_FORCE_RCRRP: writing is pending, the Response pending transmission starts immediately
Description	The Dcm_WriteMemory callout is used to write memory data identified by the parameter memoryAddress and memorySize. This service is needed for the implementation of UDS services : <ul style="list-style-type: none"> • WriteMemoryByAddress • RequestDownload • TransferData 	
Available via	Dcm_Externals.h	

]()

Note : The callout implementation shall take care of the following points :

- When writing data in NVRAM, take care to keep the consistency with data in the mirror RAM
- When writing data in memory, take care that a [SW-C](#) won't overwrite the data. Maybe the [SW-C](#) should be informed of this writing

8.5.3 Dcm_SetProgConditions

[SWS_Dcm_00543] [

Service Name	Dcm_SetProgConditions	
Syntax	<pre>Std_ReturnType Dcm_SetProgConditions (Dcm_OpStatusType OpStatus, const Dcm_ProgConditionsType* ProgConditions)</pre>	
Service ID [hex]	0x61	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	OpStatus DCM_INITIAL DCM_PENDING DCM_CANCEL DCM_FORCE_RCRRP_OK
	ProgConditions	Conditions on which the jump to bootloader has been requested
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Conditions have correctly been set E_NOT_OK: Conditions cannot be set DCM_E_PENDING: Conditions set is in progress, a further call to this API is needed to end the setting DCM_E_FORCE_RCRRP: Application requests the transmission of a response Response Pending (NRC 0x78)
Description	The Dcm_SetProgConditions callout allows the integrator to store relevant information prior to jumping to bootloader / jump due to ECUReset request. The context parameter are defined in Dcm_ProgConditionsType.	
Available via	Dcm_Externals.h	

⌋()

Note: In case the SecurityAccess AttemptCounter needs to be shared between application and bootloader in addition to the ProgConditionStructure the current value can be retrieved via the [API Xxx_GetSecurityAttemptCounter](#) (see chapter 7.5.4 Interaction)

8.5.4 Dcm_GetProgConditions

[SWS_Dcm_00544] [

Service Name	Dcm_GetProgConditions	
Syntax	<pre>Dcm_EcuStartModeType Dcm_GetProgConditions (Dcm_ProgConditionsType * ProgConditions)</pre>	
Service ID [hex]	0x62	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	ProgConditions	Conditions on which the jump from the bootloader has been requested





Return value	Dcm_EcuStartModeType	-
Description	The Dcm_GetProgConditions callout is called upon Dcm initialization and allows to determine if a response (\$50 or \$51) has to be sent. The context parameters are defined in Dcm_ProgConditionsType.	
Available via	Dcm_Externals.h	

⌋()

8.5.5 Dcm_ProcessRequestAddFile

[SWS_Dcm_91078] [

Service Name	Dcm_ProcessRequestAddFile	
Syntax	<pre>Std_ReturnType Dcm_ProcessRequestAddFile (Dcm_OpStatusType OpStatus, uint16 filePathAndNameLength, const uint8* filePathAndName, uint8 dataFormatIdentifier, uint64 fileSizeUncompressed, uint64 fileSizeCompressed, uint64* maxNumberOfBlockLength, Dcm_NegativeResponseType* ErrorCode)</pre>	
Service ID [hex]	0x72	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL: All In-parameters are valid. DCM_PENDING: All In-parameters are set to 0x00. DCM_CANCEL: All In-parameters are set to 0x00. DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x00.
	filePathAndNameLength	Defines the length in bytes for the parameter filePathAndName.
	filePathAndName	Defines the file system location of the server where the file which shall be added, deleted, replaced or read from depending on the parameter modeOfOperation parameter. In addition this parameter includes the file name of the file which shall be added, deleted, replaced or read as part of the file path.
	dataFormatIdentifier	This data-parameter is a one byte value with each nibble encoded separately. The high nibble specifies the "compression Method", and the low nibble specifies the "encryptingMethod". The value 0x00 specifies that neither compressionMethod nor encryptingMethod is used. Values other than 0x00 are vehicle manufacturer specific.
	fileSizeUncompressed	Defines the size of the uncompressed file to be download in bytes.
	fileSizeCompressed	Defines the size of the compressed file to be downloaded in bytes.
Parameters (inout)	None	
Parameters (out)	maxNumberOfBlock Length	Max number of bytes to be included in each TransferData request excluding the SID and the blockSequenceCounter.





	ErrorCode	If the operation Dcm_ProcessRequestAddFile returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful E_NOT_OK: Request was not successful DCM_E_PENDING: Request is not yet finished DCM_E_FORCE_RCRRP: Application request the transmission of a response Response Pending (NRC 0x78)
Description	Callout function. DCM shall call this function to start a RequestFileTransfer process with mode OfOperation equal to 0x01 (AddFile).	
Available via	Dcm_Externals.h	

⌋()

8.5.6 Dcm_ProcessRequestDeleteFile

[SWS_Dcm_91079] ⌈

Service Name	Dcm_ProcessRequestDeleteFile	
Syntax	<pre>Std_ReturnType Dcm_ProcessRequestDeleteFile (Dcm_OpStatusType OpStatus, uint16 filePathAndNameLength, const uint8* filePathAndName, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x73	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL: All In-parameters are valid. DCM_PENDING: All In-parameters are set to 0x00. DCM_CANCEL: All In-parameters are set to 0x00. DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x00.
	filePathAndNameLength	Defines the length in bytes for the parameter filePathAndName.
	filePathAndName	Defines the file system location of the server where the file which shall be added, deleted, replaced or read from depending on the parameter modeOfOperation parameter. In addition this parameter includes the file name of the file which shall be added, deleted, replaced or read as part of the file path.
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Dcm_ProcessRequestDeleteFile returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful E_NOT_OK: Request was not successful DCM_E_PENDING: Request is not yet finished DCM_E_FORCE_RCRRP: Application request the transmission of a response Response Pending (NRC 0x78)
Description	Callout function. DCM shall call this function to start a RequestFileTransfer process with mode OfOperation equal to 0x02 (DeleteFile).	





Available via	Dcm_Externals.h
----------------------	-----------------

]()

8.5.7 Dcm_ProcessRequestReplaceFile

[SWS_Dcm_91080] [

Service Name	Dcm_ProcessRequestReplaceFile	
Syntax	<pre>Std_ReturnType Dcm_ProcessRequestReplaceFile (Dcm_OpStatusType OpStatus, uint16 filePathAndNameLength, const uint8* filePathAndName, uint8 dataFormatIdentifier, uint64 fileSizeUncompressed, uint64 fileSizeCompressed, uint64* maxNumberOfBlockLength, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x74	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL: All In-parameters are valid. DCM_PENDING: All In-parameters are set to 0x00. DCM_CANCEL: All In-parameters are set to 0x00. DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x00.
	filePathAndNameLength	Defines the length in bytes for the parameter filePathAndName.
	filePathAndName	Defines the file system location of the server where the file which shall be added, deleted, replaced or read from depending on the parameter modeOfOperation parameter. In addition this parameter includes the file name of the file which shall be added, deleted, replaced or read as part of the file path.
	dataFormatIdentifier	This data-parameter is a one byte value with each nibble encoded separately. The high nibble specifies the "compression Method", and the low nibble specifies the "encryptingMethod". The value 0x00 specifies that neither compressionMethod nor encryptingMethod is used. Values other than 0x00 are vehicle manufacturer specific.
	fileSizeUncompressed	Defines the size of the uncompressed file to be download in bytes.
	fileSizeCompressed	Defines the size of the compressed file to be downloaded in bytes.
Parameters (inout)	None	
Parameters (out)	maxNumberOfBlock Length	Max number of bytes to be included in each TransferData request excluding the SID and the blockSequenceCounter.
	ErrorCode	If the operation Dcm_ProcessRequestReplaceFile returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.





Return value	Std_ReturnType	E_OK: Request was successful E_NOT_OK: Request was not successful DCM_E_PENDING: Request is not yet finished DCM_E_FORCE_RCRRP: Application request the transmission of a response Response Pending (NRC 0x78)
Description	Callout function. DCM shall call this function to start a RequestFileTransfer process with mode OfOperation equal to 0x03 (ReplaceFile).	
Available via	Dcm_Externals.h	

]()

8.5.8 Dcm_ProcessRequestReadFile

[SWS_Dcm_91081] [

Service Name	Dcm_ProcessRequestReadFile	
Syntax	<pre>Std_ReturnType Dcm_ProcessRequestReadFile (Dcm_OpStatusType OpStatus, uint16 filePathAndNameLength, const uint8* filePathAndName, uint8 dataFormatIdentifier, uint64 fileSizeUncompressed, uint64 fileSizeCompressed, uint64* maxNumberOfBlockLength, Dcm_NegativeResponseType* ErrorCode)</pre>	
Service ID [hex]	0x75	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL: All In-parameters are valid. DCM_PENDING: All In-parameters are set to 0x00. DCM_CANCEL: All In-parameters are set to 0x00. DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x00.
	filePathAndNameLength	Defines the length in bytes for the parameter filePathAndName.
	filePathAndName	Defines the file system location of the server where the file which shall be added, deleted, replaced or read from depending on the parameter modeOfOperation parameter. In addition this parameter includes the file name of the file which shall be added, deleted, replaced or read as part of the file path.
	dataFormatIdentifier	This data-parameter is a one byte value with each nibble encoded separately. The high nibble specifies the "compressionMethod", and the low nibble specifies the "encryptingMethod". The value 0x00 specifies that neither compressionMethod nor encryptingMethod is used. Values other than 0x00 are vehicle manufacturer specific.
Parameters (inout)	None	
Parameters (out)	fileSizeUncompressed	Defines the size of the uncompressed file to be uploaded in bytes.
	fileSizeCompressed	Defines the size of the compressed file to be uploaded in bytes.
	maxNumberOfBlockLength	Max number of bytes to be included in each TransferData response excluding the SID and the blockSequenceCounter.





	ErrorCode	If the operation Dcm_ProcessRequestReadFile returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful E_NOT_OK: Request was not successful DCM_E_PENDING: Request is not yet finished DCM_E_FORCE_RCRRP: Application request the transmission of a response Response Pending (NRC 0x78)
Description	Callout function. DCM shall call this function to start a RequestFileTransfer process with mode OfOperation equal to 0x04 (ReadFile).	
Available via	Dcm_Externals.h	

⌋()

8.5.9 Dcm_ProcessRequestReadDir

[SWS_Dcm_91082] ⌈

Service Name	Dcm_ProcessRequestReadDir	
Syntax	<pre>Std_ReturnType Dcm_ProcessRequestReadDir (Dcm_OpStatusType OpStatus, uint16 filePathAndNameLength, const uint8* filePathAndName, uint64* dirInfoLength, uint64* maxNumberOfBlockLength, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x76	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL: All In-parameters are valid. DCM_PENDING: All In-parameters are set to 0x00. DCM_CANCEL: All In-parameters are set to 0x00. DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x00.
	filePathAndNameLength	Defines the length in bytes for the parameter filePathAndName.
	filePathAndName	Defines the file system location of the server where the file which shall be added, deleted, replaced or read from depending on the parameter modeOfOperation parameter. In addition this parameter includes the file name of the file which shall be added, deleted, replaced or read as part of the file path.
Parameters (inout)	None	
Parameters (out)	dirInfoLength	Defines the size of directory information to be uploaded in bytes.
	maxNumberOfBlockLength	Max number of bytes to be included in each TransferData request excluding the SID and the blockSequenceCounter.
	ErrorCode	If the operation Dcm_ProcessRequestReadDir returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.





Return value	Std_ReturnType	E_OK: Request was successful E_NOT_OK: Request was not successful DCM_E_PENDING: Request is not yet finished DCM_E_FORCE_RCRRP: Application request the transmission of a response Response Pending (NRC 0x78)
Description	Callout function. DCM shall call this function to start a RequestFileTransfer process with mode OfOperation equal to 0x05 (ReadDir).	
Available via	Dcm_Externals.h	

}]()

8.5.10 Dcm_WriteFile

[SWS_Dcm_91083] [

Service Name	Dcm_WriteFile	
Syntax	<pre>Std_ReturnType Dcm_WriteFile (Dcm_OpStatusType OpStatus, uint64 DataLength, uint8* Data, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL: All In-parameters are valid. DCM_PENDING: All In-parameters are set to 0x00. DCM_CANCEL: All In-parameters are set to 0x00. DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x00.
	DataLength	Defines the length in bytes for the parameter Data. The value will not exceed, but might be less, compared to the value of max NumberOfBlockLength return in Dcm_ProcessRequestFile Transfer.
	Data	Pointer to the data to be written.
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Dcm_WriteFile returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful E_NOT_OK: Request was not successful DCM_E_PENDING: Request is not yet finished DCM_E_FORCE_RCRRP: Application request the transmission of a response Response Pending (NRC 0x78)
Description	Callout function. DCM shall call this function when data is received using UDS service Transfer Data if there's an ongoing RequestFileTransfer process started with 0x01 (AddFile) or 0x03 (ReplaceFile).	
Available via	Dcm_Externals.h	

}]()

8.5.11 Dcm_ReadFileOrDir

[SWS_Dcm_91085] [

Service Name	Dcm_ReadFileOrDir	
Syntax	<pre>Std_ReturnType Dcm_ReadFileOrDir (Dcm_OpStatusType OpStatus, uint64* DataLength, uint8* Data, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x78	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL: All In-parameters are valid. DCM_PENDING: All In-parameters are set to 0x00. DCM_CANCEL: All In-parameters are set to 0x00. DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x00.
	Data	Pointer to the data to be written.
Parameters (inout)	DataLength	As in, the parameter defines the maximum block length to be used, i.e. the value of maxNumberOfBlockLength sent to the client in the response of RequestFileTransfer. As out, the parameter defines the actual length in bytes for the parameter Data. The value shall not exceed, but might be less, the value provided as in parameter.
	ErrorCode	If the operation Dcm_ReadFileOrDir returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful E_NOT_OK: Request was not successful DCM_E_PENDING: Request is not yet finished DCM_E_FORCE_RCRRP: Application request the transmission of a response Response Pending (NRC 0x78)
Description	Callout function. DCM shall call this function when data shall be sent as a response to UDS service TransferData if there's an ongoing RequestFileTransfer process started with 0x04 (ReadFile) or 0x05 (ReadDir).	
Available via	Dcm_Externals.h	

]()

8.6 Scheduled functions

These functions are directly called by Basic Software Scheduler. The following functions shall have no return value and no parameter. All functions shall be non reentrant.

8.6.1 Dcm_MainFunction

[SWS_Dcm_00053] [

Service Name	Dcm_MainFunction
Syntax	void Dcm_MainFunction (void)
Service ID [hex]	0x25
Description	This service is used for processing the tasks of the main loop.
Available via	SchM_Dcm.h

]([SRS_BSW_00424](#), [SRS_BSW_00373](#))

8.7 Expected interfaces

In this chapter all interfaces required from other modules are listed.

8.7.1 Mandatory interfaces

This section defines all interfaces, which are required to fulfill the core functionality of the module.

[SWS_Dcm_91001] [

API Function	Header File	Description
ComM_DCM_ActiveDiagnostic	ComM_Dcm.h	Indication of active diagnostic by the DCM.
ComM_DCM_InactiveDiagnostic	ComM_Dcm.h	Indication of inactive diagnostic by the DCM.
PduR_DcmTransmit	PduR_Dcm.h	Requests transmission of a PDU.

]()

8.7.2 Optional interfaces

This section defines all interfaces, which are required to fulfill an optional functionality of the module.

[SWS_Dcm_91002] [

API Function	Header File	Description
BndM_GetBlockPtr_<Block Id.Shortname>	BndM_Externals.h	– Tags: atp.Status=draft
BndM_WriteBlock_<Block Id.Shortname>	BndM_Externals.h	Will persist the data in flash, that it can later directly accessed via BndM_GetBlockPtr API. The writing take a while and is finished after the successful callback xxx_WriteBlockFinish. Tags: atp.Status=draft
BndM_WriteCancel	BndM.h	Cancels the writing Tags: atp.Status=draft
BndM_WriteFinalize	BndM.h	Will trigger the finalization of writing phase. The finish of asynchronous processing will trigger the callback xxx_WriteFinalizeFinish including the result of this operation. Tags: atp.Status=draft
BndM_WriteStart	BndM.h	Will trigger the start of writing phase. The finish of asynchronous processing will trigger the callback xxx_WriteFinalizeFinish including the result of this operation. Tags: atp.Status=draft
BswM_Dcm_ApplicationUpdated	BswM_Dcm.h	This function is called by the DCM in order to report an updated application.
BswM_Dcm_CommunicationMode_CurrentState	BswM_Dcm.h	Function called by DCM to inform the BswM about the current state of the communication mode.
Dem_ClearDTC	Dem.h	Clears single DTCs, as well as groups of DTCs.
Dem_DcmGetAvailableOBDMIDs	Dem_Dcm.h	Reports the value of a requested "availability-OBDMID" to the DCM upon a Service \$06 request. Derived from that the tester displays the supported tests a mechanic can select from. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if $\{\{ecuc(Dem/DemGeneral.DemOBDSupport)\} \neq DEM_OBD_NO_OBD_SUPPORT\}$
Dem_DcmGetDTCofOBDFreezeFrame	Dem_Dcm.h	Gets DTC by freeze frame record number. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if $\{\{ecuc(Dem/DemGeneral.DemOBDSupport)\} \neq DEM_OBD_NO_OBD_SUPPORT\}$
Dem_DcmGetDTRData	Dem_Dcm.h	Reports a DTR data along with TID-value, UaSID, test result with lower and upper limit. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if $\{\{ecuc(Dem/DemGeneral.DemOBDSupport)\} \neq DEM_OBD_NO_OBD_SUPPORT\}$
Dem_DcmGetNumTIDsOfOBDMID	Dem_Dcm.h	Gets the number of TIDs per (functional) OBDMID. This can be used by the DCM to iteratively request for OBD/TID result data within a loop from 0...numberOfTIDs-1 API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if $\{\{ecuc(Dem/DemGeneral.DemOBDSupport)\} \neq DEM_OBD_NO_OBD_SUPPORT\}$





API Function	Header File	Description
Dem_DcmReadDataOfOBDFreezeFrame	Dem_Dcm.h	Gets data element per PID and index of the most important freeze frame being selected for the output of service \$02. The function stores the data in the provided DestBuffer. API is needed in OBD-relevant ECUs only. API Availability: This API will be available only if $\{\{ecuc(Dem/DemGeneral.DemOBDSupport)\} \neq DEM_OBD_NO_OBD_SUPPORT\}$
Dem_DisableDTCRecordUpdate	Dem.h	Disables the event memory update of a specific DTC (only one at one time).
Dem_DisableDTCSetting	Dem.h	Disables the DTC setting for all DTCs assigned to the DemEventMemorySet of the addressed client.
Dem_EnableDTCRecordUpdate	Dem.h	Enables the event memory update of the DTC disabled by Dem_DisableDTCRecordUpdate() before.
Dem_EnableDTCSetting	Dem.h	(Re)-Enables the DTC setting for all DTCs assigned to the DemEventMemorySet of the addressed client.
Dem_GetDTCByOccurrenceTime	Dem.h	Gets the DTC by occurrence time. There is no explicit parameter for the DTC-origin as the origin always is DEM_DTC_ORIGIN_PRIMARY_MEMORY.
Dem_GetDTCSeverityAvailabilityMask	Dem.h	Gets the DTC Severity availability mask.
Dem_GetDTCStatusAvailabilityMask	Dem.h	Gets the DTC Status availability mask of the selected fault memory.
Dem_GetFunctionalUnitOfDTC	Dem.h	Gets the functional unit of the requested DTC.
Dem_GetNextExtendedDataRecord	Dem.h	Gets extended data record for the DTC selected by Dem_SelectExtendedDataRecord. The function stores the data in the provided DestBuffer.
Dem_GetNextFilteredDTC	Dem.h	Gets the next filtered DTC matching the filter criteria. For UDS services, the interface has an asynchronous behavior, because a large number of DTCs has to be processed.
Dem_GetNextFilteredDTCAndFDC	Dem.h	Gets the next filtered DTC and its associated Fault Detection Counter (FDC) matching the filter criteria. The interface has an asynchronous behavior, because a large number of DTCs has to be processed and the FDC might be received asynchronously from a SW-C, too.
Dem_GetNextFilteredDTCAndSeverity	Dem.h	Gets the next filtered DTC and its associated Severity matching the filter criteria. The interface has an asynchronous behavior, because a large number of DTCs has to be processed.
Dem_GetNextFilteredRecord	Dem.h	Gets the next freeze frame record number and its associated DTC stored in the event memory. The interface has an asynchronous behavior, because NvRAM access might be required.
Dem_GetNextFreezeFrameData	Dem.h	Gets freeze frame data by the DTC selected by Dem_SelectFreezeFrameData. The function stores the data in the provided DestBuffer.
Dem_GetNumberOfFilteredDTC	Dem.h	Gets the number of a filtered DTC.
Dem_GetNumberOfFreezeFrameRecords	Dem.h	This function returns the number of all freeze frame records currently stored in the primary event memory





API Function	Header File	Description
Dem_GetSeverityOfDTC	Dem.h	Gets the severity of the requested DTC. For large configurations and DTC-calibration, the interface behavior can be asynchronous (splitting the DTC-search into segments).
Dem_GetSizeOfExtendedDataRecord Selection	Dem.h	Gets the size of Extended Data Record by DTC selected by the call of Dem_SelectExtendedData Record.
Dem_GetSizeOfFreezeFrameSelection	Dem.h	Gets the size of freeze frame data by DTC selected by the call of Dem_SelectFreezeFrameData.
Dem_GetStatusOfDTC	Dem.h	Gets the status of a DTC. For large configurations and DTC-calibration, the interface behavior can be asynchronous (splitting the DTC-search into segments). The DTCs of OBD Events Suppression shall be reported as Dem_WRONG_DTC.
Dem_GetTranslationType	Dem.h	Gets the supported DTC formats of the ECU. The supported formats are configured via DemTypeOf DTCsupported.
Dem_SetDTCFilter	Dem.h	Sets the DTC Filter. The server shall perform a bit-wise logical AND-ing operation between the parameter DTCStatusMask and the current UDS status in the server. In addition to the DTCStatusAvailabilityMask, the server shall return all DTCs for which the result of the AND-ing operation is non-zero [i.e. (statusOfDTC & DTCStatusMask) != 0]. The server shall process only the DTC Status bits that it is supporting. OBD Events Suppression shall be ignored for this computation. If no DTCs within the server match the masking criteria specified in the clients request, no DTC or status information shall be provided following the DTCStatusAvailabilityMask byte in the positive response message (((statusOfDTC & DTCStatusMask) != 0) && ((severity & DTCSeverityMask) != 0)) == TRUE
Dem_SetFreezeFrameRecordFilter	Dem.h	Sets a freeze frame record filter.
Det_ReportError	Det.h	Service to report development errors.
IoHwAb_Dcm_<EcuSignalName>	IoHwAb_Dcm.h	This function provides control access to a certain ECU Signal to the DCM module (<EcuSignalname> is the symbolic name of an ECU Signal). The ECU signal can be locked and unlocked by this function. Locking 'freezes' the ECU signal to the current value, the configured default value or a value given by the parameter 'signal'.
IoHwAb_Dcm_Read<EcuSignal Name>	IoHwAb_Dcm.h	This function provides read access to a certain ECU Signal to the DCM module (<EcuSignalname> is the symbolic name of an ECU Signal).
NvM_ReadBlock	NvM.h	Service to copy the data of the NV block to its corresponding RAM block.
NvM_SetBlockLockStatus	NvM.h	Service for setting the lock status of a permanent RAM block or of the explicit synchronization of a NVRAM block.
NvM_SetRamBlockStatus	NvM.h	Service for setting the RAM block status of a permanent RAM block or the status of the explicit synchronization of a NVRAM block.





API Function	Header File	Description
NvM_WriteBlock	NvM.h	Service to copy the data of the RAM block to its corresponding NV block.
PduR_DcmCancelReceive	PduR_Dcm.h	Requests cancellation of an ongoing reception of a PDU in a lower layer transport protocol module.
PduR_DcmCancelTransmit	PduR_Dcm.h	Requests cancellation of an ongoing transmission of a PDU in a lower layer communication module.
SchM_ActMainFunction_Dcm	<none>	Invokes the SchM_ActMainFunction function to trigger the activation of a corresponding main processing function.

⌋()

Dem_DcmReadDataOfOBDFreezeFrame is only required when OBD Service \$02 is configured (see configuration parameter [DcmDsdSidTabServiceId](#)).

8.7.3 Configurable interfaces

This section defines the interfaces where the [Dcm](#) configuration defines the actual functions that the [Dcm](#) will use. Depending on the configuration, an implementation of these functions could be provided by other BSW-modules (typically the DEM) or by software-components (through the RTE).

8.7.3.1 SecurityAccess

From the point of view of the DCM, the operation has the following signature:

8.7.3.1.1 GetSeed

If [DcmDspDataUsePort](#) is set to [USE_DATA_ASYNC_CLIENT_SERVER](#) or [USE_DATA_ASYNC_CLIENT_SERVER_ERROR](#), the following definition is used:

If [DcmDspSecurityADRSize](#) is present:

[SWS_Dcm_01151] [

Service Name	Xxx_GetSeed





Syntax	<pre>Std_ReturnType Xxx_GetSeed (const uint8* SecurityAccessDataRecord, Dcm_OpStatusType OpStatus, uint8* Seed, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x44	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SecurityAccessData Record	This data record contains additional data to calculate the seed value; the size of this parameter is DcmDspSecurityADRSIZE which is at least "1".
	OpStatus	Status of the current operation
Parameters (inout)	None	
Parameters (out)	Seed	Pointer for provided seed
	ErrorCode	If the operation Xxx_GetSeed returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	Request to application for asynchronous provision of seed value	
Available via	Dcm_Externals.h	

]()

If [DcmDspSecurityADRSIZE](#) is not present:

[SWS_Dcm_91003] [

Service Name	Xxx_GetSeed	
Syntax	<pre>Std_ReturnType Xxx_GetSeed (Dcm_OpStatusType OpStatus, uint8* Seed, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x45	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
Parameters (inout)	None	
Parameters (out)	Seed	Pointer for provided seed
	ErrorCode	If the operation Xxx_GetSeed returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.





Description	Request to application for asynchronous provision of seed value
Available via	Dcm_Externals.h

]()

8.7.3.1.2 CompareKey

[SWS_Dcm_91004] [

Service Name	Xxx_CompareKey	
Syntax	<pre>Std_ReturnType Xxx_CompareKey (const uint8* Key, Dcm_OpStatusType OpStatus, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x47	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	Key	Key, which needs to be compared
	OpStatus	Status of the current operation
Parameters (inout)	None	
Parameters (out)	ErrorCode	NRC to be sent if E_NOT_OK is returned
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish. DCM_E_COMPARE_KEY_FAILED: Key did not match.
Description	Request to application for asynchronous comparing key (DcmDspSecurityUsePort = USE_ASYNC_CLIENT_SERVER)	
Available via	Dcm_Externals.h	

]()

8.7.3.1.3 GetSecurityAttemptCounter

[SWS_Dcm_01152] [

Service Name	Xxx_GetSecurityAttemptCounter	
Syntax	<pre>Std_ReturnType Xxx_GetSecurityAttemptCounter (Dcm_OpStatusType OpStatus, uint8* AttemptCounter)</pre>	



△

Service ID [hex]	0x59	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL DCM_PENDING DCM_CANCEL
Parameters (inout)	None	
Parameters (out)	AttemptCounter	The attempt counter for this security level
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	Read the attempt counter for a specific security level from the application	
Available via	Dcm_Externals.h	

]()

Note: In case the Security Access AttemptCounter needs to be shared between application and bootloader, the application needs to consider this in the API-call [Xxx_GetSecurityAttemptCounter](#) (see chapter 7.5.4 Interaction). Further this has also impact on the security delay timer which needs to be considered.

8.7.3.1.4 SetSecurityAttemptCounter

[SWS_Dcm_01153] [

Service Name	Xxx_SetSecurityAttemptCounter	
Syntax	Std_ReturnType Xxx_SetSecurityAttemptCounter (Dcm_OpStatusType OpStatus, uint8 AttemptCounter)	
Service ID [hex]	0x5a	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL DCM_PENDING DCM_CANCEL
	AttemptCounter	The attempt counter for this security level
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	Set the attempt counter for a specific security level in the application	
Available via	Dcm_Externals.h	

]()

8.7.3.2 DataServices

From the point of view of the DCM, the operations have the following signatures:

Note : The OpStatus parameter shall only exist for asynchronous operations (if `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_CLIENT_SERVER` or `USE_DATA_ASYNC_CLIENT_SERVER_ERROR` or `USE_DATA_ASYNC_FNC` or `USE_DATA_ASYNC_FNC_ERROR`). In case of synchronous operations (`DcmDspDataUsePort` is set to `USE_DATA_SYNC_CLIENT_SERVER` or `USE_DATA_SYNC_FNC`), the OpStatus parameter shall not exist.

8.7.3.2.1 ReadData

If `DcmDspDataUsePort` is set to `USE_DATA_SYNC_CLIENT_SERVER` or `USE_DATA_SYNC_FNC`, the following definition is used:

[SWS_Dcm_00793] [

Service Name	Xxx_ReadData	
Syntax	Std_ReturnType Xxx_ReadData (uint8* Data)	
Service ID [hex]	0x34	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	Data	Buffer where the requested data shall be copied to
Return value	Std_ReturnType	E_OK: this value is always returned.
Description	This function requests to the application a data value of a DID/PID if <code>DcmDspDataUsePort</code> is set to <code>USE_DATA_SYNC_CLIENT_SERVER</code> .	
Available via	Dcm_Externals.h	

]()

If `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_CLIENT_SERVER` or `USE_DATA_ASYNC_FNC`, the following definition is used:

[SWS_Dcm_91006] [

Service Name	Xxx_ReadData	
Syntax	Std_ReturnType Xxx_ReadData (Dcm_OpStatusType OpStatus, uint8* Data)	



△

Service ID [hex]	0x3b	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
Parameters (inout)	None	
Parameters (out)	Data	Buffer where the requested data shall be copied to
Return value	Std_ReturnType	E_OK: Request was successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests to the application a data value of a DID/PID if DcmDspDataUsePort is set to USE_DATA_ASYNC_CLIENT_SERVER.	
Available via	Dcm_Externals.h	

}]()

If `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_CLIENT_SERVER_ERROR` or `USE_DATA_ASYNC_FNC_ERROR`, the following definition is used:

[SWS_Dcm_91005] [

Service Name	Xxx_ReadData	
Syntax	<pre>Std_ReturnType Xxx_ReadData (Dcm_OpStatusType OpStatus, uint8* Data, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x58	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
Parameters (inout)	None	
Parameters (out)	Data	Buffer where the requested data shall be copied to
	ErrorCode	If the operation <code>Xxx_ReadData</code> returns value <code>E_NOT_OK</code> , the Dcm module shall send a negative response with NRC code equal to the parameter <code>ErrorCode</code> parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests to the application a data value of a DID/PID if <code>DcmDspDataUsePort</code> is set to <code>USE_DATA_ASYNC_CLIENT_SERVER</code> .	
Available via	Dcm_Externals.h	

}]()

8.7.3.2.2 WriteData

If `DcmDspDataUsePort` is set to `USE_DATA_SYNC_CLIENT_SERVER` or `USE_DATA_SYNC_FNC`, the following definition is used:

If `DcmDspDataType` is NOT set to `UINT8_DYN`, the following definition is used:

[SWS_Dcm_00794] [

Service Name	Xxx_WriteData	
Syntax	<pre>Std_ReturnType Xxx_WriteData (const uint8* Data, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x51	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	Data	Buffer containing the data to be written
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation <code>Xxx_WriteData</code> returns value <code>E_NOT_OK</code> , the DCM module shall send a negative response with NRC code equal to the parameter <code>ErrorCode</code> parameter value.
Return value	Std_ReturnType	<code>E_OK</code> : Request was successful. <code>E_NOT_OK</code> : Request was not successful.
Description	This function requests the application to write a data value of a DID.	
Available via	Dcm_Externals.h	

]()

If `DcmDspDataType` is set to `UINT8_DYN`, the following definition is used:

[SWS_Dcm_91007] [

Service Name	Xxx_WriteData	
Syntax	<pre>Std_ReturnType Xxx_WriteData (const uint8* Data, uint16 DataLength, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x52	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	Data	Buffer containing the data to be written
	DataLength	Length in byte of the data to be written
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation <code>Xxx_WriteData</code> returns value <code>E_NOT_OK</code> , the DCM module shall send a negative response with NRC code equal to the parameter <code>ErrorCode</code> parameter value.
Return value	Std_ReturnType	<code>E_OK</code> : Request was successful. <code>E_NOT_OK</code> : Request was not successful.
Description	This function requests the application to write a data value of a DID.	
Available via	Dcm_Externals.h	

]()

If `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_CLIENT_SERVER` or `USE_DATA_ASYNC_FNC` or `USE_DATA_ASYNC_CLIENT_SERVER_ERROR` or `USE_DATA_ASYNC_FNC_ERROR`, the following definition is used:

If `DcmDspDataType` is NOT set to `UINT8_DYN`, the following definition is used:

[SWS_Dcm_91008] [

Service Name	Xxx_WriteData	
Syntax	<pre>Std_ReturnType Xxx_WriteData (const uint8* Data, Dcm_OpStatusType OpStatus, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x35	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	Data	Buffer containing the data to be written
	OpStatus	Status of the current operation
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation <code>Xxx_WriteData</code> returns value <code>E_NOT_OK</code> , the DCM module shall send a negative response with NRC code equal to the parameter <code>ErrorCode</code> parameter value.
Return value	Std_ReturnType	<code>E_OK</code> : Request was successful. <code>E_NOT_OK</code> : Request was not successful. <code>DCM_E_PENDING</code> : Request is not yet finished. Further call(s) required to finish.
Description	This function requests the application to write a data value of a DID.	
Available via	<code>Dcm_Externals.h</code>	

]()

If `DcmDspDataType` is set to `UINT8_DYN`, the following definition is used:

[SWS_Dcm_91009] [

Service Name	Xxx_WriteData	
Syntax	<pre>Std_ReturnType Xxx_WriteData (const uint8* Data, uint16 DataLength, Dcm_OpStatusType OpStatus, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x3e	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	Data	Buffer containing the data to be written
	DataLength	Length in byte of the data to be written
	OpStatus	Status of the current operation





Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_WriteData returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests the application to write a data value of a DID.	
Available via	Dcm_Externals.h	

}]()

8.7.3.2.3 ReadDataLength

If `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_CLIENT_SERVER` or `USE_DATA_SYNCH_FNC`, the following definition is used:

[SWS_Dcm_00796] [

Service Name	Xxx_ReadDataLength	
Syntax	Std_ReturnType Xxx_ReadDataLength (uint16* DataLength)	
Service ID [hex]	0x36	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	DataLength	Length in byte of the data to be read
Return value	Std_ReturnType	E_OK: this value is always returned.
Description	This function requests the application to return the data length in byte of a Data.	
Available via	Dcm_Externals.h	

}]()

If `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_CLIENT_SERVER` or `USE_DATA_ASYNC_CLIENT_SERVER_ERROR` or `USE_DATA_ASYNC_FNC` or `USE_DATA_ASYNC_FNC_ERROR`, the following definition is used:

[SWS_Dcm_91010] [

Service Name	Xxx_ReadDataLength	
Syntax	Std_ReturnType Xxx_ReadDataLength (Dcm_OpStatusType OpStatus, uint16* DataLength)	
Service ID [hex]	0x4c	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
Parameters (inout)	None	
Parameters (out)	DataLength	Length in byte of the data to be read
Return value	Std_ReturnType	E_OK: this value is always returned. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests the application to return the data length in byte of a Data.	
Available via	Dcm_Externals.h	

]()

8.7.3.2.4 ConditionCheckRead

If `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_CLIENT_SERVER` or `USE_DATA_SYNCH_FNC`, the following definition is used:

[SWS_Dcm_00797] [

Service Name	Xxx_ConditionCheckRead	
Syntax	Std_ReturnType Xxx_ConditionCheckRead (Dcm_NegativeResponseCodeType* ErrorCode)	
Service ID [hex]	0x49	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation <code>Xxx_ConditionCheckRead</code> returns value <code>E_NOT_OK</code> , the DCM module shall send a negative response with NRC code equal to the parameter <code>ErrorCode</code> parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests to the application if the conditions to read the Data are correct.	
Available via	Dcm_Externals.h	

]()

If `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_CLIENT_SERVER` or `USE_DATA_ASYNC_FNC` or `USE_DATA_ASYNC_CLIENT_SERVER_ERROR` or `USE_DATA_ASYNC_FNC_ERROR`, the following definition is used:

[SWS_Dcm_91011] [

Service Name	Xxx_ConditionCheckRead	
Syntax	Std_ReturnType Xxx_ConditionCheckRead (Dcm_OpStatusType OpStatus, Dcm_NegativeResponseCodeType* ErrorCode)	
Service ID [hex]	0x37	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_ConditionCheckRead returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests to the application if the conditions to read the Data are correct.	
Available via	Dcm_Externals.h	

]()

8.7.3.2.5 GetScalingInformation

This function requests to the application for the scaling information of a Data (scalingByte and scalingByteExtension).

If `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_CLIENT_SERVER` or `USE_DATA_SYNCH_FNC`, the following definition is used:

[SWS_Dcm_00798] [

Service Name	Xxx_GetScalingInformation	
Syntax	Std_ReturnType Xxx_GetScalingInformation (uint8* ScalingInfo, Dcm_NegativeResponseCodeType* ErrorCode)	
Service ID [hex]	0x4b	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	ScalingInfo	Scaling information (scalingByte and scalingByteExtension)



△

	ErrorCode	If the operation Xxx_GetScalingInformation returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests to the application for the scaling information of a Data.	
Available via	Dcm_Externals.h	

⌋()

If `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_CLIENT_SERVER` or `USE_DATA_ASYNC_FNC` or `USE_DATA_ASYNC_CLIENT_SERVER_ERROR` or `USE_DATA_ASYNC_FNC_ERROR`, the following definition is used:

[SWS_Dcm_91012] [

Service Name	Xxx_GetScalingInformation	
Syntax	<pre>Std_ReturnType Xxx_GetScalingInformation (Dcm_OpStatusType OpStatus, uint8* ScalingInfo, Dcm_NegativeResponseType* ErrorCode)</pre>	
Service ID [hex]	0x38	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
Parameters (inout)	None	
Parameters (out)	ScalingInfo	Scaling information (scalingByte and scalingByteExtension)
	ErrorCode	If the operation Xxx_GetScalingInformation returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests to the application for the scaling information of a Data.	
Available via	Dcm_Externals.h	

⌋()

8.7.3.2.6 ReturnControlToECU

[SWS_Dcm_01285] [

Service Name	Xxx_ReturnControlToECU	
Syntax	<pre>Std_ReturnType Xxx_ReturnControlToECU ([Dcm_ControlMask_{DID}Type controlMask], [uint8* controlMask2], Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x4f	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Variation	<pre>{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort))==(USE_DATA_ SYNCH_FNC USE_DATA_ASYNCH_FNC USE_DATA_ASYNCH_FNC_ERROR) && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidFreeze CurrentState)} == TRUE) {(ecuc(Dcm/Dc</pre>	
Parameters (in)	controlMask	–
	Variation	<pre>{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData UsePort)) == (USE_DATA_SYNCH_FNC USE_DATA_ ASYNCH_FNC USE_DATA_ASYNCH_FNC_ERROR) && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDid Control/DcmDspDidFreezeCurrentState)} == TRUE) {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDid Control/DcmDspDidResetToDefault)} == TRUE) {(ecuc(Dcm/ DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/ DcmDspDidShortTermAdjustment))== TRUE) && {(ecuc(Dcm/ DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> Dcm DspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_CONTROLMASK_EXTERNAL) && {(ecuc(Dcm/DcmConfig Set/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/ DcmDspDidControl/DcmDspDidControlMaskSize)) <= 0x04)</pre>
	controlMask2	–
	Variation	<pre>{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData UsePort)) == (USE_DATA_SYNCH_FNC USE_DATA_ ASYNCH_FNC USE_DATA_ASYNCH_FNC_ERROR) && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDid Control/DcmDspDidFreezeCurrentState)} == TRUE) {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDid Control/DcmDspDidResetToDefault)} == TRUE) {(ecuc(Dcm/ DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/ DcmDspDidShortTermAdjustment))== TRUE) && {(ecuc(Dcm/ DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> Dcm DspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_CONTROLMASK_EXTERNAL)&& {(ecuc(Dcm/DcmConfig Set/DcmDsp/DcmDspDid/?DcmDspDidInfoRef->DcmDspDidInfo/ DcmDspDidControl/DcmDspDidControlMaskSize)) => 0x05)</pre>
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_ReturnControlToECU returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests to the application to return control to ECU of an IOControl.	
Available via	Dcm_Externals.h	

]()

Note: Square brackets [] indicate that an argument is optional.

8.7.3.2.7 ResetToDefault

8.7.3.2.7.1 Synchronous interface

[SWS_Dcm_01286] [

Service Name	Xxx_ResetToDefault	
Syntax	<pre>Std_ReturnType Xxx_ResetToDefault ([Dcm_ControlMask_{DID}Type controlMask], [uint8* controlMask2], Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x4d	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	controlMask	–
	Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData UsePort)) == (USE_DATA_SYNCH_FNC)) && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidResetToDefault)) == TRUE} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) == DCM_CONTROLMASK_EXTERNAL}&& {(ecuc(Dcm/?DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)) <= 0x04}
	controlMask2	–
Variation		{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData UsePort)) == (USE_DATA_SYNCH_FNC)) && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidResetToDefault)) == TRUE} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef -> DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) == DCM_CONTROLMASK_EXTERNAL}&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)) => 0x05}
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_ResetToDefault returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests to the application to reset an IOControl to default value.	
Available via	Dcm_Externals.h	

]()

Note: Square brackets [] indicate that an argument is optional.

8.7.3.2.7.2 Asynchronous interface

[SWS_Dcm_01314] [

Service Name	Xxx_ResetToDefault	
Syntax	<pre>Std_ReturnType Xxx_ResetToDefault (Dcm_OpStatusType OpStatus, [Dcm_ControlMask_{DID}Type controlMask], [uint8* controlMask2], Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x3c	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
	controlMask	–
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData UsePort)} == (USE_DATA_ASYNC_FNC USE_DATA_ASYNC_FNC_ERROR)&& {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidResetToDefault)} == TRUE && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef -> DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_CONTROLMASK_EXTERNAL}&& {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)} <= 0x04)
	controlMask2	–
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData UsePort)} == (USE_DATA_ASYNC_FNC USE_DATA_ASYNC_FNC_ERROR)&& {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidResetToDefault)} == TRUE && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef -> DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_CONTROLMASK_EXTERNAL}&& {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)} => 0x05)
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_ResetToDefault returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests to the application to reset an IOControl to default value.	
Available via	Dcm_Externals.h	

]()

Note: Square brackets [] indicate that an argument is optional.

8.7.3.2.8 FreezeCurrentState

8.7.3.2.8.1 Synchronous interface

[SWS_Dcm_01290] [

Service Name	Xxx_FreezeCurrentState	
Syntax	<pre>Std_ReturnType Xxx_FreezeCurrentState ([Dcm_ControlMask_{DID}Type controlMask], [uint8* controlMask2], Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x4a	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	controlMask	–
	Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData UsePort)) == USE_DATA_SYNCH_FNC}&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidFreezeCurrentState)) == TRUE} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef -> DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) == DCM_CONTROLMASK_EXTERNAL}&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)) <= 0x04}
	controlMask2	–
Variation		{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData UsePort)) == USE_DATA_SYNCH_FNC}&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidFreezeCurrentState)) == TRUE} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef -> DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) == DCM_CONTROLMASK_EXTERNAL}&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)) => 0x05}
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_FreezeCurrentState returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests to the application to freeze the current state of an IOControl.	
Available via	Dcm_Externals.h	

]()

Note: Square brackets [] indicate that an argument is optional.

8.7.3.2.8.2 Asynchronous interface

[SWS_Dcm_01315] [

Service Name	Xxx_FreezeCurrentState
---------------------	------------------------





Syntax	<pre>Std_ReturnType Xxx_FreezeCurrentState (Dcm_OpStatusType OpStatus, [Dcm_ControlMask_{DID}]Type controlMask], [uint8* controlMask2], Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]		
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
	controlMask	–
	Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData UsePort)) == (USE_DATA_ASYNC_FNC USE_DATA_ASYNC_FNC_ERROR)}&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidFreezeCurrentState)) == TRUE} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef -> DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) == DCM_CONTROLMASK_EXTERNAL}&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)) <= 0x04}
	controlMask2	–
Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData UsePort)) == (USE_DATA_ASYNC_FNC USE_DATA_ASYNC_FNC_ERROR)}&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidFreezeCurrentState)) == TRUE} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef -> DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) == DCM_CONTROLMASK_EXTERNAL}&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)) => 0x05}	
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_FreezeCurrentState returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests to the application to freeze the current state of an IOControl.	
Available via	Dcm_Externals.h	

]()

8.7.3.2.9 ShortTermAdjustment

8.7.3.2.9.1 Synchronous interface

[SWS_Dcm_00802] [

Service Name	Xxx_ShortTermAdjustment	
Syntax	<pre>Std_ReturnType Xxx_ShortTermAdjustment (const uint8* ControlStateInfo, [Dcm_ControlMask_{DID}]Type controlMask], [uint8* controlMask2], Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x50	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ControlStateInfo	ControlState information contained in the ControlOptionRecord parameter of the InputOutputControlByIdentifier diagnostic request
	controlMask	–
	Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData.UsePort)) == USE_DATA_SYNCH_FNC} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)) == TRUE} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType) != UINT8_DYN)} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef -> DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) == DCM_CONTROLMASK_EXTERNAL}&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)) <= 0x04}
	controlMask2	–
Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData.UsePort)) == USE_DATA_SYNCH_FNC} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)) == TRUE} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType) != UINT8_DYN)} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef -> DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) == DCM_CONTROLMASK_EXTERNAL}&& {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)) >= 0x05}	
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_ShortTermAdjustment returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests to the application to adjust the IO signal.	
Available via	Dcm_Externals.h	

]()

Note: Square brackets [] indicate that an argument is optional.

8.7.3.2.9.2 Asynchronous interface

[SWS_Dcm_01316] [

Service Name	Xxx_ShortTermAdjustment	
Syntax	<pre>Std_ReturnType Xxx_ShortTermAdjustment (const uint8* ControlStateInfo, uint16 DataLength, Dcm_OpStatusType OpStatus, [Dcm_ControlMask_{DID}Type controlMask], [uint8* controlMask2], Dcm_NegativeResponseType* ErrorCode)</pre>	
Service ID [hex]	0x55	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ControlStateInfo	ControlState information contained in the ControlOptionRecord parameter of the InputOutputControlByIdentifier diagnostic request
	DataLength	Length in byte of the data to be written
	OpStatus	Status of the current operation
	controlMask	–
	Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData UsePort)} == (USE_DATA_ASYNC_FNC USE_DATA_ASYNC_FNC_ERROR)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)} == TRUE) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType) == UINT8_DYN) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef -> DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_CONTROLMASK_EXTERNAL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)} <= 0x04)
controlMask2	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData UsePort)} == (USE_DATA_ASYNC_FNC USE_DATA_ASYNC_FNC_ERROR)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)} == TRUE) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType) == UINT8_DYN) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef -> DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_CONTROLMASK_EXTERNAL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)} >= 0x05)	
Parameters (inout)	None	
Parameters (out)	ErrorCode	NRC to be sent in the negative response in case of failure (E_NOT_OK)
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests to the application to adjust the IO signal.	
Available via	Dcm_Externals.h	

10

8.7.3.3 DataServices_DIDRange

From the point of view of the DCM, the operations have the following signatures:

Note : The OpStatus parameter should only be used for asynchronous operations (if `DcmDspDataUsePort` is set to `USE_DATA_ASYNC_CLIENT_SERVER` or `USE_DATA_ASYNC_FNC` or `USE_DATA_ASYNC_CLIENT_SERVER_ERROR` or `USE_DATA_ASYNC_FNC_ERROR`). In case of synchronous operations (`DcmDspDataUsePort` is set to `USE_DATA_SYNC_CLIENT_SERVER` or `USE_DATA_SYNC_FNC`), the OpStatus parameter should not be used.

8.7.3.3.1 IsDidAvailable

[SWS_Dcm_00803] [

Service Name	Xxx_IsDidAvailable	
Syntax	<pre>Std_ReturnType Xxx_IsDidAvailable (uint16 DID, Dcm_OpStatusType OpStatus, Dcm_DidSupportedType* supported)</pre>	
Service ID [hex]	0x53	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	DID	DID value
	OpStatus	Status of the current operation
Parameters (inout)	None	
Parameters (out)	supported	Indicate if the DID is available within the range. Returning DCM_DID_SUPPORTED means it is supported within the range, Returning DCM_DID_NOT_SUPPORTED means it is not supported within the range
Return value	Std_ReturnType	E_OK: This value is returned when the Did is finally available. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests if a specific DID is available within the range or not.	
Available via	Dcm_Externals.h	

]()

8.7.3.3.2 ReadDidData

[SWS_Dcm_00804] [

Service Name	Xxx_ReadDidData	
Syntax	<pre>Std_ReturnType Xxx_ReadDidData (uint16 DID, uint8* Data, Dcm_OpStatusType OpStatus, uint16 DataLength, Dcm_NegativeResponseCodeType ErrorCode)</pre>	
Service ID [hex]	0x40	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	DID	Data ID value
	OpStatus	Status of the current operation
Parameters (inout)	None	
Parameters (out)	Data	Buffer where the requested data shall be copied to
	DataLength	Length of the data to be read
	ErrorCode	If the operation Xxx_ReadDidData returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests to the application a data value of a DID	
Available via	Dcm_Externals.h	

]()

8.7.3.3.3 WriteDidData

[SWS_Dcm_00805] [

Service Name	Xxx_WriteDidData	
Syntax	<pre>Std_ReturnType Xxx_WriteDidData (uint16 DID, const uint8* Data, Dcm_OpStatusType OpStatus, uint16 DataLength, Dcm_NegativeResponseCodeType ErrorCode)</pre>	
Service ID [hex]	0x41	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	DID	Data ID value
	Data	Buffer containing the data to be written
	OpStatus	Status of the current operation
	DataLength	Length of the data to be written





Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_WriteDidData returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests the application to write a data value of a DID.	
Available via	Dcm_Externals.h	

}]()

8.7.3.3.4 ReadDidRangeDataLength

ReadDidRangeDataLength requests the application to return the data length of a DID range. This interface is used for UDS Service ReadDataByIdentifier.

[SWS_Dcm_01271] [

Service Name	Xxx_ReadDidRangeDataLength	
Syntax	Std_ReturnType Xxx_ReadDidRangeDataLength (uint16 DID, Dcm_OpStatusType OpStatus, uint16* DataLength)	
Service ID [hex]	0x5e	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	DID	Data ID value
	OpStatus	Status of the current operation
Parameters (inout)	None	
Parameters (out)	DataLength	Length of the data to be written/read
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	This function requests the application to return the data length of a DID Range.	
Available via	Dcm_Externals.h	

}]()

8.7.3.4 InfoTypesServices

8.7.3.4.1 GetInfotypeValueData

[SWS_Dcm_91014] [

Service Name	Xxx_GetInfotypeValueData	
Syntax	<pre>Std_ReturnType Xxx_GetInfotypeValueData (Dcm_OpStatusType OpStatus, uint8* DataValueBuffer, uint8* DataValueBufferSize)</pre>	
Service ID [hex]	0x60	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
Parameters (inout)	DataValueBufferSize	When the function is called this parameter contains the maximum number of data bytes that can be written to the buffer. The callee fills in the number of written data bytes in DataValueBuffer.
Parameters (out)	DataValueBuffer	Buffer containing the Infotype information
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish.
Description	The function provides the data related to the requested Infotype.	
Available via	Dcm_Externals.h	

]()

8.7.3.5 RoutineServices

The operations mentioned in the following sub-chapters are only general examples, because the number of In and OUT parameters can be variable from 0 to an arbitrary number. It is therefore not possible to list all variations of operation prototypes.

8.7.3.5.1 Xxx_Start Operation

[SWS_Dcm_01203] [

Service Name	Xxx_Start	
Syntax	<pre>Std_ReturnType Xxx_Start ([DcmDspRoutineSignalType dataIn_1, ... DcmDspRoutineSignalType dataIn_n], [const uint8* dataInVar], Dcm_OpStatusType OpStatus, [DcmDspRoutineSignalType dataOut_1, ... DcmDspRoutineSignalType dataOut_n], [uint8* dataOutVar], [uint16* currentDataLength], Dcm_NegativeResponseCodeType ErrorCode)</pre>	
Service ID [hex]	0x5b	





Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	dataIn_1	Fixed-length input data provided in the routine control request

	dataIn_n	Fixed-length input data provided in the routine control request
	dataInVar	Variable-length input data provided in the routine control request
	OpStatus	Status of the current operation
Parameters (inout)	currentDataLength	If variable length routine input data is used, this parameter contains the length in bytes of the dataInVar array. If variable length routine output data is used, this parameter contains the length in bytes of the dataOutVar parameter.
Parameters (out)	dataOut_1	Fixed-length output data to provide in the routine control response

	dataOut_n	Fixed-length output data to provide in the routine control response
	dataOutVar	Variable-length output data to provide in the routine control response
	ErrorCode	If the operation Xxx_Start returns value E_NOT_OK, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish. DCM_E_FORCE_RCRRP: application requests the transmission of a response Pending (NRC 0x78)
Description	This function requests to the application to start the execution of a routine.	
Available via	Dcm_Externals.h	

⌋()

Note: Square brackets [] indicate that an argument is optional.

8.7.3.5.2 Xxx_StartConfirmation Operation

[SWS_Dcm_91016] ⌈

Service Name	Xxx_StartConfirmation	
Syntax	Std_ReturnType Xxx_StartConfirmation (Dcm_ConfirmationStatusType ConfirmationStatus)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ConfirmationStatus	Confirmation status of a StartRoutine request
Parameters (inout)	None	
Parameters (out)	None	





Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function indicates the transmission of a response to a StartRoutine request	
Available via	Dcm_Externals.h	

⌋()

Note: Square brackets [] indicate that an argument is optional.

8.7.3.5.3 Xxx_Stop Operation

[SWS_Dcm_01204] [

Service Name	Xxx_Stop	
Syntax	<pre>Std_ReturnType Xxx_Stop ([DcmDspRoutineSignalType dataIn_1, ... DcmDspRoutineSignalType dataIn_n], [const uint8* dataInVar], [DcmDspRoutineSignalType dataOut_1, ... DcmDspRoutineSignalType dataOut_n], [uint8* dataOutVar], [uint16* currentDataLength], Dcm_NegativeResponseType* ErrorCode)</pre>	
Service ID [hex]	0x5c	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	dataIn_1	Fixed-length input data provided in the routine control request

	dataIn_n	Fixed-length input data provided in the routine control request
	dataInVar	Variable-length input data provided in the routine control request
Parameters (inout)	currentDataLength	If variable length routine input data is used, this parameter contains the length in bytes of the dataInVar array. If variable length routine output data is used, this parameter contains the length in bytes of the dataOutVar parameter.
Parameters (out)	dataOut_1	Fixed-length output data to provide in the routine control response

	dataOut_n	Fixed-length output data to provide in the routine control response
	dataOutVar	Variable-length output data to provide in the routine control response
	ErrorCode	If the operation Xxx_Stop returns value E_NOT_OK, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.



△

Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish DCM_E_FORCE_RCRRP: application requests the transmission of a response Pending (NRC 0x78)
Description	This function requests to the application to stop the execution of a routine	
Available via	Dcm_Externals.h	

]()

Note: Square brackets [] indicate that an argument is optional.

8.7.3.5.4 Xxx_StopConfirmation Operation

[SWS_Dcm_91017] [

Service Name	Xxx_StopConfirmation	
Syntax	Std_ReturnType Xxx_StopConfirmation (Dcm_ConfirmationStatusType ConfirmationStatus)	
Service ID [hex]	0x69	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ConfirmationStatus	Dcm_ConfirmationStatus Confirmation status of a StopRoutine request
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function indicates the transmission of a response to a StopRoutine request	
Available via	Dcm_Externals.h	

]()

Note: Square brackets [] indicate that an argument is optional.

8.7.3.5.5 Xxx_RequestResults Operation

[SWS_Dcm_91013] [

Service Name	Xxx_RequestResults	
Syntax	<pre>Std_ReturnType Xxx_RequestResults (Dcm_OpStatusType OpStatus, [DcmDspRoutineSignalType* dataIn_1, ... DcmDspRoutineSignalType* dataIn_n], [const uint8* dataInVar], [DcmDspRoutineSignalType* dataOut_1, ... DcmDspRoutineSignalType* dataOut_n], [uint8* dataOutVar], [uint16* variableDataLength], Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x71	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	Status of the current operation
	dataIn_1	Fixed-length input data provided in the routine control request

	dataIn_n	Fixed-length input data provided in the routine control request
	dataInVar	Variable-length input data provided in the routine control request
Parameters (inout)	variableDataLength	Length in bytes of the dataOutVar parameter.
Parameters (out)	dataOut_1	Fixed-length Output data to provide in the routine control response

	dataOut_n	Fixed-length Output data to provide in the routine control response
	dataOutVar	Variable-length Output data to provide in the routine control response
	ErrorCode	If the operation Xxx_RequestResults returns value E_NOT_OK, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. DCM_E_PENDING: Request is not yet finished. Further call(s) required to finish DCM_E_FORCE_RCRRP: application requests the transmission of a response Pending (NRC 0x78)
Description	This function requests to the application the result of a routine execution	
Available via	Dcm_Externals.h	

⌋()

Note: Square brackets [] indicate that an argument is optional.

8.7.3.5.6 Xxx_RequestResultsConfirmation Operation

[SWS_Dcm_91018] [

Service Name	Xxx_RequestResultsConfirmation	
Syntax	Std_ReturnType Xxx_RequestResultsConfirmation (Dcm_ConfirmationStatusType ConfirmationStatus)	
Service ID [hex]	0x70	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ConfirmationStatus	Confirmation status of a RequestRoutineResults request
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function indicates the transmission of a response to a RequestRoutineResults request	
Available via	Dcm_Externals.h	

]()

Note: Square brackets [] indicate that an argument is optional.

8.7.3.6 RequestControlServices

From the point of view of the DCM, the operation has the following signature:

8.7.3.6.1 RequestControl callout

[SWS_Dcm_01338] [

Service Name	Xxx_RequestControl	
Syntax	Std_ReturnType Xxx_RequestControl (uint8* OutBuffer, const uint8* InBuffer)	
Service ID [hex]	0x63	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	OutBuffer	Output buffer in which the RequestControl function can store its result
	InBuffer	Input buffer containing the data of the OBD Service 0x08 request
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	Invokes a TID-specific function taking a configured number of bytes as input and returning a fixed number of bytes as output. This is typically used to implement OBD Service \$08	





Available via	Dcm_Externals.h
----------------------	-----------------

}]()

8.7.3.7 CallbackDCMRequestServices

From the point of view of the DCM, the operations have the following signatures:

8.7.3.7.1 StartProtocol

[SWS_Dcm_01339] [

Service Name	Xxx_StartProtocol	
Syntax	<pre>Std_ReturnType Xxx_StartProtocol (Dcm_ProtocolType ProtocolType, uint16 TesterSourceAddress, uint16 ConnectionId)</pre>	
Service ID [hex]	0x67	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ProtocolType	Type of the protocol to be started
	TesterSourceAddress	source address of the tester
	ConnectionId	Unique connection identifier
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. E_PROTOCOL_NOT_ALLOWED: Protocol not allowed
Description	This function allows the application to examine the environment conditions and enable/disable further processing of the protocol.	
Available via	Dcm_Externals.h	

}]()

8.7.3.7.2 StopProtocol

[SWS_Dcm_01340] [

Service Name	Xxx_StopProtocol	
Syntax	<pre>Std_ReturnType Xxx_StopProtocol (Dcm_ProtocolType ProtocolType, uint16 TesterSourceAddress, uint16 ConnectionId)</pre>	
Service ID [hex]	0x64	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ProtocolType	Type of the protocol to be stopped
	TesterSourceAddress	source address of the tester
	ConnectionId	Unique connection identifier
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
	Description	
Description		This function informs the application of the protocol stop.
Available via	Dcm_Externals.h	

]()

8.7.3.8 ServiceRequestNotification

From the point of view of the DCM, the operations has the following signatures:

8.7.3.8.1 Indication

[SWS_Dcm_01341] [

Service Name	Xxx_Indication	
Syntax	<pre>Std_ReturnType Xxx_Indication (uint8 SID, const uint8* RequestData, uint32 DataSize, uint8 ReqType, uint16 ConnectionId, Dcm_NegativeResponseCodeType* ErrorCode, Dcm_ProtocolType ProtocolType, uint16 TesterSourceAddress)</pre>	
Service ID [hex]	0x65	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SID	-





	RequestData	Complete request data (diagnostic buffer), except the service ID
	DataSize	Number of valid bytes in the RequestData parameter
	ReqType	Addressing type of the request(0=physical request,1=functional request)
	ConnectionId	Unique connection identifier
	ProtocolType	Type of the protocol to be indicated
	TesterSourceAddress	source address of the tester
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_Indication returns value E_NOT_OK, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. E_REQUEST_NOT_ACCEPTED : Request not accepted
Description	This function indicates to the application that a service is about to be executed and allows the application to reject the execution of the service request	
Available via	Dcm_Externals.h	

⌋()

8.7.3.8.2 Confirmation

[SWS_Dcm_01342] [

Service Name	Xxx_Confirmation	
Syntax	<pre>Std_ReturnType Xxx_Confirmation (uint8 SID, uint8 ReqType, uint16 ConnectionId, Dcm_ConfirmationStatusType ConfirmationStatus, Dcm_ProtocolType ProtocolType, uint16 TesterSourceAddress)</pre>	
Service ID [hex]	0x66	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	SID	Value of service identifier
	ReqType	Addressing type of the request(0=physical request,1=functional request)
	ConnectionId	Unique connection identifier
	ConfirmationStatus	Confirmation of a successful transmission or a transmission error of a diagnostic service.
	ProtocolType	Type of Dcm Protocol
	TesterSourceAddress	source address of the tester
Parameters (inout)	None	





Parameters (out)	None	
Return value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function confirms to the application the successful transmission or a transmission error of a diagnostic service.	
Available via	Dcm_Externals.h	

}]()

8.7.3.9 ClearDTCCheckFnc

From the point of view of the Dcm, the operation has the following signature:

[SWS_Dcm_01270] [

Service Name	Xxx_ClearDTCCheckFnc	
Syntax	Std_ReturnType Xxx_ClearDTCCheckFnc (uint32 GoDTC, Dcm_NegativeResponseCodeType* ErrorCode)	
Service ID [hex]	0x5f	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	GoDTC	requested groupOfDTC
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Xxx_ClearDTCCheckFnc returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: application allows to clear the requested groupOfDTC E_NOT_OK: application does not allow to clear the requested groupOfDTC. Dcm shall send a negative response with the NRC returned in the ErrorCode
Description	Callout function for condition check, manufacturer / supplier specific checks on the groupOfDTC, which is requested to clear.	
Available via	Dcm_Externals.h	

}]()

8.7.3.10 UploadDownloadServices

From the point of view of the DCM, the operations has the following signatures:

8.7.3.10.1 ProcessRequestDownload

[SWS_Dcm_00754] [

Service Name	Xxx_ProcessRequestDownload	
Syntax	<pre>Std_ReturnType Xxx_ProcessRequestDownload (Dcm_OpStatusType OpStatus, uint8 DataFormatIdentifier, uint8 MemoryIdentifier, uint32 MemoryAddress, uint32 MemorySize, uint32* BlockLength, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x30	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL DCM_PENDING DCM_CANCEL
	DataFormatIdentifier	Bit 7 - 4: Compression Method <ul style="list-style-type: none"> • 0x0: not compressed • 0x1..F: vehicle-manufacturer-specific Bit 3 - 0: Encrypting method <ul style="list-style-type: none"> • 0x0: not encrypted • 0x1..F: vehicle-manufacturer-specific
	MemoryIdentifier	Identifier of the Memory Block, if the parameter is not used it shall be set to 0.
	MemoryAddress	Starting address of server memory to which data is to be written
	MemorySize	Uncompressed memory size in bytes
Parameters (inout)	BlockLength	Max. Number of bytes for one Dcm_WriteMemory
Parameters (out)	ErrorCode	If the operation Dcm_ProcessRequestDownload returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful E_NOT_OK: Request was not successful DCM_E_PENDING: Request is not yet finished
Description	Callout function. DCM shall call this callout function to start a download process. This service is needed for the implementation of UDS service RequestDownload.	
Available via	Dcm_Externals.h	

]()

8.7.3.10.2 ProcessRequestTransferExit

[SWS_Dcm_00755] [

Service Name	Xxx_ProcessRequestTransferExit	
Syntax	<pre>Std_ReturnType Xxx_ProcessRequestTransferExit (Dcm_OpStatusType OpStatus, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	





Service ID [hex]	0x32	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL DCM_PENDING DCM_CANCEL
Parameters (inout)	None	
Parameters (out)	ErrorCode	see below
Return value	Std_ReturnType	E_OK: Transfer was successful E_NOT_OK: Transfer was not successful or the response buffer is too small DCM_E_PENDING: Transfer is not yet finished
Description	Callout function. DCM shall call this callout function to terminate a download or upload process. This callout is needed for the implementation of UDS service RequestTransferExit.	
Available via	Dcm_Externals.h	

⌋()

8.7.3.10.3 ProcessRequestUpload

[SWS_Dcm_00756] [

Service Name	Xxx_ProcessRequestUpload	
Syntax	<pre>Std_ReturnType Xxx_ProcessRequestUpload (Dcm_OpStatusType OpStatus, uint8 DataFormatIdentifier, uint8 MemoryIdentifier, uint32 MemoryAddress, uint32 MemorySize, uint32* BlockLength, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x31	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL DCM_PENDING DCM_CANCEL DCM_FORCE_RCRRP_OK
	DataFormatIdentifier	Bit 7 - 4: Compression Method <ul style="list-style-type: none"> • 0x0: not compressed • 0x1..F: vehicle-manufacturer-specific Bit 3 - 0: Encrypting method <ul style="list-style-type: none"> • 0x0: not encrypted • 0x1..F: vehicle-manufacturer-specific
	MemoryIdentifier	Identifier of the Memory Block, if the parameter is not used it shall be set to 0.
	MemoryAddress	Starting address of server memory from which data are to be copied





	MemorySize	Uncompressed memory size in bytes
Parameters (inout)	BlockLength	Max. Number of bytes for one Dcm_ReadMemory
Parameters (out)	ErrorCode	If the operation Dcm_ProcessRequestUpload returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful E_NOT_OK: Request was not successful DCM_E_PENDING: Request is not yet finished
Description	Callout function. DCM shall call this callout function to start an upload process. This service is needed for the implementation of UDS service RequestUpload.	
Available via	Dcm_Externals.h	

⌋()

8.7.3.10.4 ProcessTransferDataRead

[SWS_Dcm_91070] [

Service Name	Xxx_ProcessTransferDataRead	
Syntax	<pre>Dcm_ReturnReadMemoryType Xxx_ProcessTransferDataRead (Dcm_OpStatusType OpStatus, uint8 MemoryIdentifier, uint32 MemoryAddress, uint32 MemorySize, Dcm_RequestDataArrayType MemoryData, Dcm_NegativeResponseCodeType* ErrorCode)</pre>	
Service ID [hex]	0x26	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL DCM_PENDING DCM_CANCEL DCM_FORCE_RCRRP_OK
	MemoryIdentifier	Identifier of the Memory Block (e.g. used if memory section distinguishing is needed) Note: If it's not used this parameter shall be set to 0.
	MemoryAddress	Starting address of server memory from which data is to be retrieved.
	MemorySize	Number of bytes in the MemoryData
Parameters (inout)	None	
Parameters (out)	MemoryData	Data read (Points to the diagnostic buffer in DCM)
	ErrorCode	If the operation Dcm_ReadMemory returns value DCM_READ_FAILED, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Dcm_ReturnReadMemoryType	DCM_READ_OK: read was successful DCM_READ_FAILED: read was not successful DCM_READ_PENDING: read is not yet finished DCM_READ_FORCE_RCRRP: reading is pending, the Response pending transmission starts immediately





Description	<p>The ProcessTransferDataRead callout is used to request memory data identified by the parameter memoryAddress and memorySize from the UDS request message. This service is needed for the implementation of UDS services:</p> <ul style="list-style-type: none"> • ReadMemoryByAddress • RequestUpload • ReadDataByIdentifier (in case of Dynamical DID defined by memory address) • TransferData
Available via	Dcm_Externals.h

]()

8.7.3.10.5 ProcessTransferDataWrite

[SWS_Dcm_91071] [

Service Name	Xxx_ProcessTransferDataWrite	
Syntax	<pre>Dcm_ReturnWriteMemoryType Xxx_ProcessTransferDataWrite (Dcm_OpStatusType OpStatus, uint8 MemoryIdentifier, uint32 MemoryAddress, uint32 MemorySize, const Dcm_RequestDataArrayType MemoryData, Dcm_NegativeResponseType* ErrorCode)</pre>	
Service ID [hex]	0x27	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL DCM_PENDING DCM_CANCEL DCM_FORCE_RCRRP_OK
	MemoryIdentifier	Identifier of the Memory Block (e.g. used by WriteDataByIdentifier service). Note: If it's not used this parameter shall be set to 0.
	MemoryAddress	Starting address of server memory in which data is to be copied. Note: If it's not used (e.g. if the data is compressed) this parameter shall be set to 0.
	MemorySize	Number of bytes in MemoryData
	MemoryData	Data to write (Points to the diagnostic buffer in DCM)
Parameters (inout)	None	
Parameters (out)	ErrorCode	If the operation Dcm_WriteMemory returns value DCM_WRITE_FAILED, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Dcm_ReturnWriteMemoryType	DCM_WRITE_OK: write was successful DCM_WRITE_FAILED: write was not successful DCM_WRITE_PENDING: write is not yet finished DCM_WRITE_FORCE_RCRRP: writing is pending, the Response pending transmission starts immediately





Description	The ProcessTransferDataWrite callout is used to write memory data identified by the parameter memoryAddress and memorySize. This service is needed for the implementation of UDS services : <ul style="list-style-type: none"> • WriteMemoryByAddress • RequestDownload • TransferData
Available via	Dcm_Externals.h

}]()

8.8 Dcm as Service-Component

8.8.1 Implementation Data Types

8.8.1.1 Dcm_OpStatusType

[SWS_Dcm_00984] [

Name	Dcm_OpStatusType		
Kind	Type		
Derived from	uint8		
Range	DCM_INITIAL	0x00	Indicates the initial call to the operation
	DCM_PENDING	0x01	Indicates that a pending return has been done on the previous call of the operation
	DCM_CANCEL	0x02	Indicates that the DCM requests to cancel the pending operation
	DCM_FORCE_RCRRP_OK	0x03	Confirm a response pending transmission
Description	-		
Variation	-		
Available via	Rte_Dcm_Type.h		

}]()

8.8.1.2 Dcm_ConfirmationStatusType

[SWS_Dcm_00983] [

Name	Dcm_ConfirmationStatusType		
Kind	Type		
Derived from	uint8		
Range	DCM_RES_POS_OK	0x00	–
	DCM_RES_POS_NOT_OK	0x01	–
	DCM_RES_NEG_OK	0x02	–
	DCM_RES_NEG_NOT_OK	0x03	–
Description	–		
Variation	–		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.3 Dcm_SecLevelType

[SWS_Dcm_00977] [

Name	Dcm_SecLevelType		
Kind	Type		
Derived from	uint8		
Range	DCM_SEC_LEV_LOCKED	0x00	–
	configuration dependent	0x01...0x3F	–
	Reserved by Document	0x40...0xFF	–
Description	Security Level type definition		
Variation	–		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.4 Dcm_SesCtrlType

[SWS_Dcm_00978] [

Name	Dcm_SesCtrlType		
Kind	Type		
Derived from	uint8		
Range	DCM_DEFAULT_SESSION	0x01	–
	DCM_PROGRAMMING_SESSION	0x02	–
	DCM_EXTENDED_DIAGNOSTIC_SESSION	0x03	–





	DCM_SAFETY_SYSTEM_DIAGNOSTIC_SESSION	0x04	–
	configuration dependent	0x40...0x7E	(according to "diagnosticSessionType" parameter of Diagnostic SessionControl request)
Description	Session type definition. 0, 127 and all values above 127 are reserved by ISO.		
Variation	–		
Available via	Rte_Dcm_Type.h		

⌋()

8.8.1.5 Dcm_ProtocolType

[SWS_Dcm_00979] [

Name	Dcm_ProtocolType		
Kind	Type		
Derived from	uint8		
Range	DCM_OBD_ON_CAN	0x00	OBD on CAN (ISO15765-4; ISO15031-5)
	DCM_OBD_ON_FLEXRAY	0x01	(OBD on Flexray (Manufacturer specific; ISO15031-5))
	DCM_OBD_ON_IP	0x02	(OBD on Internet Protocol (Manufacturer specific; ISO15031-5))
	DCM_UDS_ON_CAN	0x03	UDS on CAN (ISO15765-3; ISO14229-1)
	DCM_UDS_ON_FLEXRAY	0x04	UDS on FlexRay (Manufacturer specific; ISO14229-1)
	DCM_UDS_ON_IP	0x05	(UDS on Internet Protocol (Manufacturer specific; ISO14229-1))
	DCM_ROE_ON_CAN	0x06	Response On Event on CAN
	DCM_ROE_ON_FLEXRAY	0x07	Response On Event on FlexRay
	DCM_ROE_ON_IP	0x08	(Response on Event on Internet Protocol)
	DCM_PERIODICTRANS_ON_CAN	0x09	Periodic Transmission on CAN
	DCM_PERIODICTRANS_ON_FLEXRAY	0x0A	Periodic Transmission on FlexRay
	DCM_PERIODICTRANS_ON_IP	0x0B	(Periodic Transmission on Internet Protocol)
	DCM_NO_ACTIVE_PROTOCOL	0x0C	No protocol has been started
DCM_UDS_ON_LIN	0x0D	UDS on LIN (ISO14229-1; ISO14229-7)	





	Reserved for further AUTOSAR implementation	0x0E..0xEF	–
	DCM_SUPPLIER_1	0xF0	Reserved for SW supplier specific.
	DCM_SUPPLIER_2	0xF1	Reserved for SW supplier specific.
	DCM_SUPPLIER_3	0xF2	Reserved for SW supplier specific.
	DCM_SUPPLIER_4	0xF3	Reserved for SW supplier specific.
	DCM_SUPPLIER_5	0xF4	Reserved for SW supplier specific.
	DCM_SUPPLIER_6	0xF5	Reserved for SW supplier specific.
	DCM_SUPPLIER_7	0xF6	Reserved for SW supplier specific.
	DCM_SUPPLIER_8	0xF7	Reserved for SW supplier specific.
	DCM_SUPPLIER_9	0xF8	Reserved for SW supplier specific.
	DCM_SUPPLIER_10	0xF9	Reserved for SW supplier specific.
	DCM_SUPPLIER_11	0xFA	Reserved for SW supplier specific.
	DCM_SUPPLIER_12	0xFB	Reserved for SW supplier specific.
	DCM_SUPPLIER_13	0xFC	Reserved for SW supplier specific.
	DCM_SUPPLIER_14	0xFD	Reserved for SW supplier specific.
	DCM_SUPPLIER_15	0xFE	Reserved for SW supplier specific.
Description	Protocol type definition		
Variation	–		
Available via	Rte_Dcm_Type.h		

⌋()

8.8.1.6 Dcm_NegativeResponseCodeType

[SWS_Dcm_00980] ⌈

Name	Dcm_NegativeResponseCodeType
Kind	Type
Derived from	uint8





Range	DCM_POS_RESP	0x00	PR
	range of values 0x01..0x0F reserved by ISO 14229	0x01..0x0F	ISOSAERESRVD
	DCM_E_GENERALREJECT	0x10	GR
	DCM_E_SERVICENOTSUPPORTED	0x11	SNS
	DCM_E_SUBFUNCTIONNOTSUPPORTED	0x12	SFNS
	DCM_E_INCORRECTMESSAGELENGTHORINVALIDFORMAT	0x13	IMLOIF
	DCM_E_RESPONSETOOLONG	0x14	RTL
	range of values 0x15..0x20 reserved by ISO 14229	0x15..0x20	ISOSAERESRVD
	DCM_E_BUSYREPEATREQUEST	0x21	BRR
	DCM_E_CONDITIONSNOTCORRECT	0x22	CNC
	value 0x23 reserved by ISO 14229	0x23	ISOSAERESRVD
	DCM_E_REQUESTSEQUENCEERROR	0x24	RSE
	DCM_E_NORESPONSEFROMSUBNETCOMPONENT	0x25	NRFSC
	DCM_E_FAILUREPREVENTSEXTIONOFREQUESTEDACTION	0x26	FPEORA
	range of values 0x27..0x30 reserved by ISO 14229	0x27..0x30	ISOSAERESRVD
	DCM_E_REQUESTOUTOFRANGE	0x31	ROOR
	value 0x32 reserved by ISO 14229	0x32	ISOSAERESRVD
	DCM_E_SECURITYACCESSDENIED	0x33	SAD
	value 0x34 reserved by ISO 14229	0x34	ISOSAERESRVD
	DCM_E_INVALIDKEY	0x35	IK
	DCM_E_EXCEEDNUMBEROFATTEMPTS	0x36	ENOA
	DCM_E_REQUIREDTIMEDELAYNOTEXPIRED	0x37	RTDNE
	range of values 0x38..0x4F reserved by ISO 15764	0x38..0x4F	RBEDLSD
	range of values 0x50..0x6F reserved by ISO 14229	0x50..0x6F	ISOSAERESRVD
	DCM_E_UPLOADDOWNLOADNOTACCEPTED	0x70	UDNA
	DCM_E_TRANSFERDATASUSPENDED	0x71	TDS
	DCM_E_GENERALPROGRAMMINGFAILURE	0x72	GPF





DCM_E_WRONGBLOCKSEQUENCECOUNTER	0x73	WBSC
range of values 0x74..0x77 reserved by ISO 14229	0x74..0x77	ISOSAERESRVD
range of values 0x79..0x7D reserved by ISO 14229	0x79..0x7D	ISOSAERESRVD
DCM_E_SUBFUNCTIONNOTSUPPORTEDINACTIVESSESSION	0x7E	SFNSIAS
DCM_E_SERVICENOTSUPPORTEDINACTIVESSESSION	0x7F	SNSIAS
value 0x80 reserved by ISO 14229	0x80	ISOSAERESRVD
DCM_E_RPMTOOHIGH	0x81	RPMT
DCM_E_RPMTOOLOW	0x82	RPMTL
DCM_E_ENGINEISRUNNING	0x83	EIR
DCM_E_ENGINEISNOTRUNNING	0x84	EINR
DCM_E_ENGINERUNTIMETOLOW	0x85	ERTTL
DCM_E_TEMPERATURETOOHIGH	0x86	TEMPH
DCM_E_TEMPERATURETOOLOW	0x87	TEMPTL
DCM_E_VEHICLESPEEDTOOHIGH	0x88	VSTH
DCM_E_VEHICLESPEEDTOOLOW	0x89	VSTL
DCM_E_THROTTLE_PEDALTOOHIGH	0x8A	TPTH
DCM_E_THROTTLE_PEDALTOOLOW	0x8B	TPTL
DCM_E_TRANSMISSION-RANGENOTINNEUTRAL	0x8C	TRNIN
DCM_E_TRANSMISSION-RANGENOTINGEAR	0x8D	TRNIG
value 0x8E reserved by ISO 14229	0x8E	ISOSAERESRVD
DCM_E_BRAKESWITCH_NOTCLOSED	0x8F	BSNC
DCM_E_SHIFTERLEVERNOTINPARK	0x90	SLNIP
DCM_E_TORQUECONVERTER-CLUTCHLOCKED	0x91	TCCL
DCM_E_VOLTAGETOOHIGH	0x92	VTH
DCM_E_VOLTAGETOLOW	0x93	VTL
range of values 0x94..0xEF reserved by ISO 14229	0x94..0xEF	RFSCNC
DCM_E_VMSCNC_0	0xF0	VMSCNC
DCM_E_VMSCNC_1	0xF1	VMSCNC1
DCM_E_VMSCNC_2	0xF2	VMSCNC2





	DCM_E_VMSCNC_3	0xF3	VMSCNC3
	DCM_E_VMSCNC_4	0xF4	VMSCNC4
	DCM_E_VMSCNC_5	0xF5	VMSCNC5
	DCM_E_VMSCNC_6	0xF6	VMSCNC6
	DCM_E_VMSCNC_7	0xF7	VMSCNC7
	DCM_E_VMSCNC_8	0xF8	VMSCNC8
	DCM_E_VMSCNC_9	0xF9	VMSCNC9
	DCM_E_VMSCNC_A	0xFA	VMSCNCA
	DCM_E_VMSCNC_B	0xFB	VMSCNCB
	DCM_E_VMSCNC_C	0xFC	VMSCNCC
	DCM_E_VMSCNC_D	0xFD	VMSCNCD
	DCM_E_VMSCNC_E	0xFE	VMSCNCE
	value 0xFF reserved by ISO 14229	0xFF	ISOSAERESRVD
Description	This Table of available Negative Response Codes represents the allowed Response Codes an AUTOSAR SW Component shall return after a function call. For the allowed NRC of the executed Service ID please refer to the specification of the service in ISO14229-1 (UDS) and ISO15031-5 (OBD/CARB) (see chapter 4.2.4 Response code parameter definition Table 12).		
Variation	-		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.7 Dcm_DataElementType_{Data}Type

[SWS_Dcm_91051] [

Name	Dcm_DataElement_{Data}Type	
Kind	Type	
Derived from	Basetype	Variation
	Dcm_DataElement_{Data}_Array Type	(({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == ([S U]INT[8 16 32] FLOAT)_N) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == ([S U]INT[8 16 32] FLOAT)_N))
	Dcm_DataElement_{Data}_Primitive Type	(({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == (BOOLEAN [S U]INT[8 16 32] FLOAT)) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == (BOOLEAN [S U]INT[8 16 32] FLOAT)))
Description	Common description for S/R and C/S data elements.	





Variation	((({Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort == USE_DATA_ELEMENT_SPECIFIC_INTERFACES}) && (({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort) == USE_DATA_SENDER_RECEIVER} ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort) == USE_DATA_SENDER_RECEIVER} ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort) == USE_DATA_SENDER_RECEIVER_AS_SERVICE} ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort) == USE_DATA_SENDER_RECEIVER_AS_SERVICE} ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort) == USE_DATA_SYNCH_CLIENT_SERVER} ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort) == USE_DATA_ASYNC_CLIENT_SERVER} ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort) == USE_DATA_ASYNC_CLIENT_SERVER_ERROR} ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01.DcmDspPidDataUsePort) == USE_DATA_ASYNC_CLIENT_SERVER}))) Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)}))
Available via	Rte_Dcm_Type.h

]()

8.8.1.8 Dcm_DataElement_{Data}ArrayType

[SWS_Dcm_91063] [

Name	Dcm_DataElement_{Data}_ArrayType	
Kind	Array	
Element type	Type	Variation
	float32	((({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspData Type) == FLOAT_N} ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidData Type) == FLOAT_N}))
	sint16	((({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspData Type) == SINT16_N} ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidData Type) == SINT16_N}))
	sint32	((({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspData Type) == SINT32_N} ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidData Type) == SINT32_N}))
	sint8	((({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspData Type) == SINT8_N} ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidData Type) == SINT8_N}))
	uint16	((({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspData Type) == UINT16_N} ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidData Type) == UINT16_N}))
	uint32	((({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspData Type) == UINT32_N} ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidData Type) == UINT32_N}))





	uint8	{{{ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType) == UINT8_N) (ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType) == UINT8_N)}}
Size	–	
Description	–	
Variation	((({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType) == (S U)INT[8 16 32] FLOAT_N) (ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType) == (S U)INT[8 16 32] FLOAT_N)) &&({Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort == USE_DATA_ELEMENT_SPECIFIC_INTERFACES}) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort) == USE_DATA_SENDER_RECEIVER) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort) == USE_DATA_SENDER_RECEIVER) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort) == USE_DATA_SENDER_RECEIVER_AS_SERVICE) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort) == USE_DATA_SENDER_RECEIVER_AS_SERVICE) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort) == USE_DATA_SYNCH_CLIENT_SERVER) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort) == USE_DATA_ASYNC_CLIENT_SERVER) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort) == USE_DATA_ASYNC_CLIENT_SERVER_ERROR) (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01.DcmDspPidDataUsePort) == USE_DATA_SYNCH_CLIENT_SERVER)))) Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)})	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.9 Dcm_DataElement_{Data}PrimitiveType

[SWS_Dcm_91062] [

Name	Dcm_DataElement_{Data}_PrimitiveType	
Kind	Type	
Derived from	Basetype	Variation
	boolean	{{{ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType) == BOOLEAN) (ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType) == BOOLEAN)}}
	float32	{{{ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType) == FLOAT) (ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType) == FLOAT)}}
	sint16	{{{ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType) == SINT16) (ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType) == SINT16)}}





	sint32	((({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == SINT32) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == SINT32))
	sint8	((({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == SINT8) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == SINT8))
	uint16	((({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == UINT16) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == UINT16))
	uint32	((({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == UINT32) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == UINT32))
	uint8	((({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == UINT8) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == UINT8))
Description	–	
Variation	((({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspData.?DcmDspDataType)} == (BOOLEAN [S U]INT[8 16 32] FLOAT)) ({ecuc(Dcm/?DcmConfigSet/?DcmDsp/?DcmDspPid/?DcmDspPidData/?DcmDspPidService01.?DcmDspPidDataType)} == (BOOLEAN [S U]INT[8 16 32] FLOAT))) &&({Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort == USE_DATA_ELEMENT_SPECIFIC_INTERFACES}) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == USE_DATA_SENDER_RECEIVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort)} == USE_DATA_SENDER_RECEIVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort)} == USE_DATA_SENDER_RECEIVER_AS_SERVICE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == USE_DATA_SENDER_RECEIVER_AS_SERVICE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_ASYNC_CLIENT_SERVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_ASYNC_CLIENT_SERVER_ERROR) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01.DcmDspPidDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER))) Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)})	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.10 Dcm_DataArrayTypeUint8_{Data}Type

[SWS_Dcm_01121] [

Name	Dcm_DataArrayTypeUint8_{Data}Type		
Kind	Array	Element type	uint8
Size	((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataByteSize)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.DcmDspPidDataByteSize)})) Elements		





Description	-
Variation	(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_DATA_ELEMENT_SPECIFIC_INTERFACES) && (({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType)} == UINT8_N) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType)} == UINT8_DYN) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01.DcmDspPidDataType)} == UINT8_N)) Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)})
Available via	Rte_Dcm_Type.h

]()

8.8.1.11 {DID}_Struct_DataType

This data type has a different modeling as other data types. The {DID}_Struct_DataType datatype is modeled as prosa text only. At the time this specification was created there are no means to visualize this datatype with existing AUTOSAR tooling as table as all the other data types. Still AUTOSAR allows modeling such data types. Simply that they cannot be shown here as table.

[SWS_Dcm_91056] [

Name	{DID}_Struct_DataType
Kind	Structure
Description	The elements of this structure data type is a composition of all DcmDspDataElement of the DcmDspDid. Example: A DID with the 3 data elements uint32 data1, sint8 data2 and sint16 data 3, has a structure definition of struct { uint32 data1, sint8 data2, sint16 data 3}.
Variation	(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == ((USE_ATOMIC_SENDER_RECEIVER_INTERFACE USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE) USE_ATOMIC_NV_DATA_INTERFACE)) DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})
Available via	Rte_Dcm_Type.h

]()

8.8.1.12 Dcm_RangeArray_{Range}Type

[SWS_Dcm_01012] [

Name	Dcm_RangeArray_{Range}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRangeMaxDataLength)} Elements		
Description	-		



△

Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRange.DcmDspDidRangeUsePort)} == TRUE Range = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRange.SHORT-NAME)}
Available via	Rte_Dcm_Type.h

⌋()

8.8.1.13 Dcm_InfoTypeServicesArray_{VehInfoData}Type

[SWS_Dcm_01013] ⌈

Name	Dcm_InfoTypeServicesArray_{VehInfoData}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData.DcmDspVehInfoData Size)} Elements		
Description	-		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData/DcmDspVehInfoData UsePort)} == TRUE VehInfoData = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfo Data.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

⌋()

8.8.1.14 Dcm_RequestControlServicesInArray_{Tid}Type

[SWS_Dcm_01014] ⌈

Name	Dcm_RequestControlServicesInArray_{Tid}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl.DcmDspRequestControlInBuffer Size)} Elements		
Description	-		
Variation	Tid = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

⌋()

8.8.1.15 Dcm_RequestControlServicesOutArray_{Tid}Type

[SWS_Dcm_01015] ⌈

Name	Dcm_RequestControlServicesOutArray_{Tid}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl.DcmDspRequestControlOutBufferSize)} Elements		
Description	-		
Variation	Tid = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.16 Dcm_ScalingInfoArray_{Data}Type

[SWS_Dcm_01017] [

Name	Dcm_ScalingInfoArray_{Data}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDataInfo.DcmDspDataScalingInfoSize)} Elements		
Description	-		
Variation	(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_ ASYNCH_CLIENT_SERVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_SYNC_CLIENT_SERVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_ASYNC_CLIENT_SERVER_ERROR)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData->DcmDspDataInfoRef.DcmDspDataScalingInfoSize)} != NULL) Data = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.17 Dcm_RequestDataOut_{Routine}_{Signal}PrimitivType

[SWS_Dcm_01018] [

Name	Dcm_RequestDataOut_{Routine}_{Signal}PrimitivType		
Kind	Type		
Derived from	Basetype	Variation	
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == FLOAT	
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == SINT16	





	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == SINT32
	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == SINT8
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == UINT16
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == UINT32
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == UINT8
Description	-	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == [U S]INT[8 16 32]FLOAT) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.18 Dcm_RequestDataIn_{Routine}_{Signal}PrimitiveType

[SWS_Dcm_91054] [

Name	Dcm_RequestDataIn_{Routine}_{Signal}PrimitiveType	
Kind	Type	
Derived from	Basetype	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == FLOAT
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == SINT16
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == SINT32



△

	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == SINT8
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == UINT16
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == UINT32
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == UINT8
Description	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == [U S]INT[8 16 32] FLOAT)&& {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.19 Dcm_RequestDataOut_{Routine}_{Signal}Type

[SWS_Dcm_91040] [

Name	Dcm_RequestDataOut_{Routine}_{Signal}Type	
Kind	Type	
Derived from	Basetype	Variation
	Dcm_RequestDataOut_{Routine}_{Signal}PrimitivType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT))
	Dcm_RequestDataOut_{Routine}_{Signal}ArrayType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N
Description	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.20 Dcm_RequestDataIn_{Routine}_{Signal}Type

[SWS_Dcm_91052] [

Name	Dcm_RequestDataIn_{Routine}_{Signal}Type	
Kind	Type	
Derived from	Basetype	Variation
	Dcm_RequestDataIn_{Routine}_{Signal}PrimitiveType	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32 FLOAT])}}
	Dcm_RequestDataIn_{Routine}_{Signal}ArrayType	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32 FLOAT]_N)}
Description	–	
Variation	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE} Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.21 Dcm_RequestDataOut_{Routine}_{Signal}ArrayType

[SWS_Dcm_91041] [

Name	Dcm_RequestDataOut_{Routine}_{Signal}ArrayType	
Kind	Array	
Element type	Type	Variation
	float32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == FLOAT_N}}
	sint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == SINT16_N}}
	sint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == SINT32_N}}
	sint8	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == SINT8_N}}
	uint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == UINT16_N}}
uint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == UINT32_N}}	





	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == UINT8_N)
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineParameterSize)} Elements	
Description	-	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N) && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.22 Dcm_RequestDataIn_{Routine}_{Signal}ArrayType

[SWS_Dcm_91055] [

Name	Dcm_RequestDataIn_{Routine}_{Signal}ArrayType	
Kind	Array	
Element type	Type	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == FLOAT_N
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == SINT16_N
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == SINT32_N
	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == SINT8_N
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == UINT16_N
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == UINT32_N
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == UINT8_N
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineParameterSize)} Elements	





Description	–
Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)) == ([U S]INT[8 16 32] FLOAT)_N)&& ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)) == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
Available via	Rte_Dcm_Type.h

⌋()

8.8.1.23 Dcm_RequestFlexibleOutArrayData_{Routine}_{Signal}Type

[SWS_Dcm_01019] ⌈

Name	Dcm_RequestFlexibleOutArrayData_{Routine}_{Signal}Type		
Kind	Array	Element type	uint8
Size	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineParameterSize)} Elements		
Description	–		
Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)) == VARIABLE_LENGTH) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)) == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

⌋()

8.8.1.24 Dcm_RequestFlexibleInArrayData_{Routine}_{Signal}Type

[SWS_Dcm_91053] ⌈

Name	Dcm_RequestFlexibleInArrayData_{Routine}_{Signal}Type		
Kind	Array	Element type	uint8
Size	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineParameterSize)} Elements		
Description	–		





Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)) == VARIABLE_LENGTH) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)) == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
Available via	Rte_Dcm_Type.h

]()

8.8.1.25 Dcm_StartDataIn_{Routine}_{Signal}PrimitivType

[SWS_Dcm_01020] [

Name	Dcm_StartDataIn_{Routine}_{Signal}PrimitivType	
Kind	Type	
Derived from	Basetype	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == FLOAT
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == SINT16
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == SINT32
	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == SINT8
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == UINT16
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == UINT32
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == UINT8
Description	-	
Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == [U S]INT[8 16 32] FLOAT) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)) == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.26 Dcm_StartDataIn_{Routine}_{Signal}Type

[SWS_Dcm_91042] [

Name	Dcm_StartDataIn_{Routine}_{Signal}Type	
Kind	Type	
Derived from	Basetype	Variation
	Dcm_StartDataIn_{Routine}_{Signal}PrimitivType	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == (U S)INT[8 16 32] FLOAT}}
	Dcm_StartDataIn_{Routine}_{Signal}ArrayType	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == (U S)INT[8 16 32] FLOAT}_N}}
Description	-	
Variation	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.27 Dcm_StartDataIn_{Routine}_{Signal}ArrayType

[SWS_Dcm_91043] [

Name	Dcm_StartDataIn_{Routine}_{Signal}ArrayType	
Kind	Array	
Element type	Type	Variation
	float32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == FLOAT}_N}}
	sint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == SINT16}_N}}
	sint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == SINT32}_N}}
	sint8	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == SINT8}_N}}
	uint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == UINT16}_N}}
uint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == UINT32}_N}}	





	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == UINT8_N)
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineParameterSize)} Elements	
Description	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.28 Dcm_StartDataOut_{Routine}_{Signal}PrimitivType

[SWS_Dcm_01021] [

Name	Dcm_StartDataOut_{Routine}_{Signal}PrimitivType	
Kind	Type	
Derived from	Basetype	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == FLOAT
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == SINT16
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == SINT32
	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == SINT8
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == UINT16
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == UINT32
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == UINT8
Description	–	





Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == [U S]INT[8 16 32]FLOAT && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
Available via	Rte_Dcm_Type.h

]()

8.8.1.29 Dcm_StartDataOut_{Routine}_{Signal}Type

[SWS_Dcm_91044] [

Name	Dcm_StartDataOut_{Routine}_{Signal}Type	
Kind	Type	
Derived from	Basetype	Variation
	Dcm_StartDataOut_{Routine}_{Signal}PrimitivType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32]FLOAT)
	Dcm_StartDataOut_{Routine}_{Signal}ArrayType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32]FLOAT)_N
Description	-	
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.30 Dcm_StartDataOut_{Routine}_{Signal}ArrayType

[SWS_Dcm_91045] [

Name	Dcm_StartDataOut_{Routine}_{Signal}ArrayType	
Kind	Array	
Element type	Type	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == FLOAT_N



△

	sint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == SINT16_N)
	sint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == SINT32_N)
	sint8	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == SINT8_N)
	uint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == UINT16_N)
	uint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == UINT32_N)
	uint8	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == UINT8_N)
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineParameterSize)} Elements	
Description	-	
Variation	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N) && {{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.31 Dcm_StartFlexibleInArrayData_{Routine}_{Signal}Type

[SWS_Dcm_01022] [

Name	Dcm_StartFlexibleInArrayData_{Routine}_{Signal}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineParameterSize)} Elements		
Description	-		
Variation	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH) && {{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.32 Dcm_StartFlexibleOutArrayData_{Routine}_{Signal}Type

[SWS_Dcm_01023] [

Name	Dcm_StartFlexibleOutArrayData_{Routine}_{Signal}Type		
Kind	Array	Element type	uint8
Size	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineParameterSize) Elements		
Description	-		
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.33 Dcm_StopDataIn_{Routine}_{Signal}PrimitivType

[SWS_Dcm_01024] [

Name	Dcm_StopDataIn_{Routine}_{Signal}PrimitivType	
Kind	Type	
Derived from	Basetype	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == FLOAT
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == SINT16
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == SINT32
	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == SINT8
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == UINT16
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == UINT32
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == UINT8
Description	-	





Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32]FLOAT) && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
Available via	Rte_Dcm_Type.h

]()

8.8.1.34 Dcm_StopDataIn_{Routine}_{Signal}Type

[SWS_Dcm_91046] [

Name	Dcm_StopDataIn_{Routine}_{Signal}Type	
Kind	Type	
Derived from	Basetype	Variation
	Dcm_StopDataIn_{Routine}_{Signal}PrimitivType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32]FLOAT)
	Dcm_StopDataIn_{Routine}_{Signal}ArrayType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32]FLOAT)_N
Description	-	
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.35 Dcm_StopDataIn_{Routine}_{Signal}ArrayType

[SWS_Dcm_91047] [

Name	Dcm_StopDataIn_{Routine}_{Signal}ArrayType	
Kind	Array	
Element type	Type	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == FLOAT_N





	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == SINT16_N)
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == SINT32_N)
	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == SINT8_N)
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == UINT16_N)
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == UINT32_N)
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == UINT8_N)
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineParameterSize)} Elements	
Description	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N) && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.36 Dcm_StopDataOut_{Routine}_{Signal}PrimitivType

[SWS_Dcm_01025] [

Name	Dcm_StopDataOut_{Routine}_{Signal}PrimitivType	
Kind	Type	
Derived from	Basetype	Variation
	float32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == FLOAT
	sint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == SINT16
	sint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == SINT32



△

	sint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == SINT8
	uint16	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == UINT16
	uint32	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == UINT32
	uint8	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == UINT8
Description	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == [U S]INT[8 16 32] FLOAT) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.37 Dcm_StopDataOut_{Routine}_{Signal}Type

[SWS_Dcm_91048] [

Name	Dcm_StopDataOut_{Routine}_{Signal}Type	
Kind	Type	
Derived from	Basetype	Variation
	Dcm_StopDataOut_{Routine}_{Signal}PrimitivType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)
	Dcm_StopDataOut_{Routine}_{Signal}ArrayType	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N
Description	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.38 Dcm_StopDataOut_{Routine}_{Signal}ArrayType

[SWS_Dcm_91049] [

Name	Dcm_StopDataOut_{Routine}_{Signal}ArrayType	
Kind	Array	
Element type	Type	Variation
	float32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == FLOAT_N}
	sint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == SINT16_N}
	sint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == SINT32_N}
	sint8	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == SINT8_N}
	uint16	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == UINT16_N}
	uint32	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == UINT32_N}
	uint8	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == UINT8_N}
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineParameterSize)} Elements	
Description	–	
Variation	{{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == ([U S]INT[8 16 32] FLOAT)_N} && {{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)} == TRUE} Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.39 Dcm_StopFlexibleInArrayData_{Routine}_{Signal}Type

[SWS_Dcm_01026] [

Name	Dcm_StopFlexibleInArrayData_{Routine}_{Signal}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineParameterSize)} Elements		
Description	–		





Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)) == VARIABLE_LENGTH) && (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)) == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
Available via	Rte_Dcm_Type.h

]()

8.8.1.40 Dcm_StopFlexibleOutArrayData_{Routine}_{Signal}Type

[SWS_Dcm_01027] [

Name	Dcm_StopFlexibleOutArrayData_{Routine}_{Signal}Type		
Kind	Array	Element type	uint8
Size	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineParameterSize) Elements		
Description	-		
Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)) == VARIABLE_LENGTH) && (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRoutineUsePort)) == TRUE) Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.41 Dcm_KeyArray_{SecurityLevel}Type

[SWS_Dcm_01028] [

Name	Dcm_KeyArray_{SecurityLevel}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecurityKeySize)} Elements		
Description	-		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecurityUsePort)) == USE_ASYNC_CLIENT_SERVER SecurityLevel = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.42 Dcm_SeedArray_{SecurityLevel}Type

[SWS_Dcm_01029] [

Name	Dcm_SeedArray_{SecurityLevel}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecuritySeed Size)} Elements		
Description	–		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecurityUse Port)} == USE_ASYNC_CLIENT_SERVER SecurityLevel = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurity Row.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.43 Dcm_SecurityAccessDataRecordArray_{SecurityLevel}Type

[SWS_Dcm_01159] [

Name	Dcm_SecurityAccessDataRecordArray_{SecurityLevel}Type		
Kind	Array	Element type	uint8
Size	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow/DcmDsp SecurityADRSIZE)} Elements		
Description	–		
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecurityUse Port)} == USE_ASYNC_CLIENT_SERVER) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDsp Security/DcmDspSecurityRow.DcmDspSecurityADRSIZE)} != NULL) SecurityLevel = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurity Row.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.44 Dcm_RequestDataArrayType

[SWS_Dcm_01165] [

Name	Dcm_RequestDataArrayType		
Kind	Array	Element type	uint8
Size	(MAX({ecuc(Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocol RxBufferID->DcmDslBuffer.DcmDslBufferSize)}) - 1) Elements		



△

Description	–
Variation	–
Available via	Rte_Dcm_Type.h

]()

8.8.1.45 Dcm_ControlMask_{DID}Type

[SWS_Dcm_01320] [

Name	Dcm_ControlMask_{DID}Type	
Kind	Type	
Derived from	Basetype	Variation
	Dcm_ControlMask_{Data}Array Type	((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)} >= 0x05) && (({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == (USE_DATA_SYNCH_CLIENT_SERVER USE_DATA_ASYNC_CLIENT_SERVER USE_DATA_ASYNC_CLIENT_SERVER_ERROR))))
	uint16	(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)} == 0x02)
	uint32	((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)}== 0x03) {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)}== 0x04))
	uint8	(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)} == 0x01)
Description	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == (USE_DATA_SYNCH_CLIENT_SERVER USE_DATA_ASYNC_CLIENT_SERVER USE_DATA_ASYNC_CLIENT_SERVER_ERROR)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_CONTROLMASK_EXTERNAL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlEnableMask)} == NULL) DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})	
Available via	Rte_Dcm_Type.h	

]()

8.8.1.46 Dcm_inputOutputControlParameterType

[SWS_Dcm_01305] [

Name	Dcm_InputOutputControlParameterType		
Kind	Type		
Derived from	uint8		
Range	DCM_RETURN_CONTROL_TO_ECU	0x00	returnControlToECU
	DCM_RESET_TO_DEFAULT	0x01	resetToDefault
	DCM_FREEZE_CURRENT_STATE	0x02	freezeCurrentState
	DCM_SHORT_TERM_ADJUSTMENT	0x03	shortTermAdjustment
	DCM_IDLE	0xff	Idle state, no request in processing (initial value)
Description	–		
Variation	–		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.47 Dcm_IOOperationRequest_{DID}Type

[SWS_Dcm_01306] [

Name	Dcm_IOOperationRequest_{DID}Type		
Kind	Structure		
Elements	inputOutputControlParameter		
	Type	Dcm_InputOutputControlParameterType	
	Comment	–	
	controlEnableMask		
	Type	Dcm_Cemr_{DID}Type	
	Comment	–	
Description	–		
Variation	<pre> {{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE)} {{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE)}} && {{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidControl)} != NULL} DID = {{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)}} </pre>		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.48 Dcm_IOOperationResponseType

[SWS_Dcm_01307] [

Name	Dcm_IOOperationResponseType		
Kind	Type		
Derived from	uint8		
Range	DCM_POSITIVE_RESPONSE	0x00	positive response (similar to E_OK)
	DCM_GENERAL_REJECT	0x10	NRC generalReject
	DCM_BUSY_REPEAT_REQUEST	0x21	NRC busyRepeatRequest
	DCM_CONDITIONS_NOT_CORRECT	0x22	NRC conditionsNotCorrect
	DCM_FAILURE_PREVENTS_EXECUTION	0x26	NRC FailurePreventsExecutionOfRequestedAction
	DCM_REQUEST_OUT_OF_RANGE	0x31	NRC requestOutOfRange
	DCM_RESPONSE_PENDING	0x78	ResponsePending (similar to E_PENDING)
Description	–		
Variation	–		
Available via	Rte_Dcm_Type.h		

}]()

8.8.1.49 Dcm_DidSupportedType

[SWS_Dcm_01138] [

Name	Dcm_DidSupportedType		
Kind	Type		
Derived from	uint8		
Range	DCM_DID_SUPPORTED	0x00	–
	DCM_DID_NOT_SUPPORTED	0x01	–
Description	–		
Variation	–		
Available via	Rte_Dcm_Type.h		

}]()

8.8.1.50 Dcm_FileAndDirNameType

[SWS_Dcm_91066] [

Name	Dcm_FileAndDirNameType		
Kind	Array	Element type	uint8
Size	((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRequestFileTransfer.DcmRequestFileTransferMaxFileAndDirName))) Elements		
Description	–		
Variation	((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRequestFileTransfer/DcmRequestFileTransferUsePort)) == TRUE)		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.51 Dcm_ResponseDataArrayType

[SWS_Dcm_91064] [

Name	Dcm_ResponseDataArrayType		
Kind	Array	Element type	uint8
Size	(MAX({ecuc(Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolTxBufferRef->DcmDslBuffer.DcmDslBufferSize)}) - 1) Elements		
Description	–		
Variation	((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRequestFileTransfer/DcmRequestFileTransferUsePort)) == TRUE)		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.52 Dcm_AuthenticationRoleType

[SWS_Dcm_91068] [

Name	Dcm_AuthenticationRoleType		
Kind	Array	Element type	uint8
Size	((ecuc(Dcm/DcmDsp/DcmDspAuthentication.DcmDspAuthenticationRoleSize))) Elements		
Description	–		
Variation	–		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.53 Dcm_ControlMask_{Data}ArrayType

[SWS_Dcm_91050] [

Name	Dcm_ControlMask_{Data}ArrayType		
Kind	Array	Element type	uint8
Size	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)}) Elements		
Description	-		
Variation	(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)} >= 0x05) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == (USE_DATA_SYNCH_CLIENT_SERVER USE_DATA_ASYNC_CLIENT_SERVER USE_DATA_ASYNC_CLIENT_SERVER_ERROR))) Data = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

]()

8.8.1.54 Dcm_Cemr_{DID}Type

[SWS_Dcm_91087] [

Name	Dcm_Cemr_{DID}Type		
Kind	Bitfield		
Derived from	Dcm_ControlMask_{DID}Type		
Elements	Variable bit		
	Kind	bit	
	Name	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidControl.DcmDspDidControlEnableMask.SHORT-NAME)})	
	Mask	2 ^{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMaskSize)} * 8 - 1) - {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlEnableMask/DcmDspDidControlMaskBitPosition)})	
Description	Bitmask of data element in control enable mask record.		
Description	-		
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidControl/DcmDspDidControlEnableMask)} != NULL) DID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)}		
Available via	Rte_Dcm_Type.h		

]()

The [Dcm_Cemr_Type](#) data type defines a bitfield text table with a single mask entry for each valid bit in the control enable mask record. A value of 0 in the [DcmDspDidControlMaskBitPosition](#) represents the most significant bit in the most significant byte of the control enable mask. Therefore it is mapped to Bit 7 (0x80) of the first byte of the control enable mask. In a two byte [CEMR](#) the bit position 0 would map to 0x8000, the bit position 1 would map to 0x4000, Also this is a unusual representation it helps to keep the bits stable if the size of the [CEMR](#) is changed. The definition of the bits is directly defined by [DcmDspDidControlEnableMask](#).

8.8.2 Sender-Receiver-Interfaces

Using the concepts of the [SW-C](#) template, the interface is defined as follows if Sender-Receiver interface is used ([DcmDspDataUsePort](#) set to [USE_DATA_SENDER_RECEIVER](#) || [USE_DATA_SENDER_RECEIVER_AS_SERVICE](#)):

8.8.2.1 DataServices_{DID}

[SWS_Dcm_91057] [

Name	DataServices_{DID}	
Comment	-	
IsService	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE)	
Variation	((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE))) DID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})	
Data Elements	data	
	Type	{DID}_Struct_DataType
	Variation	DID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)}

]([RS_Diag_04218](#))

8.8.2.2 DataServices_{Data}

[SWS_Dcm_00687] [

Name	DataServices_{Data}	
Comment	-	
IsService	((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == USE_DATA_SENDER_RECEIVER_AS_SERVICE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort)} == USE_DATA_SENDER_RECEIVER_AS_SERVICE))	
Variation	((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_DATA_ELEMENT_SPECIFIC_INTERFACES)&&({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == (USE_DATA_SENDER_RECEIVER USE_DATA_SENDER_RECEIVER_AS_SERVICE))) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort)} == (USE_DATA_SENDER_RECEIVER USE_DATA_SENDER_RECEIVER_AS_SERVICE))) Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)})	
Data Elements	data	
	Type	Dcm_DataElement_{Data}Type
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)})





--	--	--

]()

8.8.2.3 IOControlRequest_{DID}

[SWS_Dcm_01308] [

Name	IOControlRequest_{DID}	
Comment	Attention: controlState is only valid in case of IOOperationRequest is set to shortTerm Adjustment. The DCM provides a byte stream which could be transformed via transformer into an complex type.	
IsService	true	
Variation	<pre>(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)} == True) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_CONTROLMASK_EXTERNAL))) DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})</pre>	
Data Elements	underControl	
	Type	Dcm_Cemr_{DID}Type
	Variation	-
	IOOperationRequest	
	Type	Dcm_IOOperationRequest_{DID}Type
	Variation	DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})
	controlState	
	Type	{DID}_Struct_DataType
Variation	<pre>({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)} == True) DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})</pre>	

]()

8.8.2.4 IOControlResponse_{DID}

[SWS_Dcm_01309] [

Name	IOControlResponse_{DID}
Comment	-
IsService	true





Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidControl)} != NULL) DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})	
Data Elements	IOOperationResponse	
	Type	Dcm_IOOperationResponseType
	Variation	–

]()

8.8.3 Client-Server-Interfaces

8.8.3.1 SecurityAccess_{SecurityLevel}

Provides functions required for the UDS Service SecurityAccess (see [\[SWS_Dcm_00323\]](#), [\[SWS_Dcm_00862\]](#) and [\[SWS_Dcm_00863\]](#)).

Using the concepts of the SW-C template, the interface is defined as follows if ClientServer interface is used ([DcmDspSecurityUsePort](#) set to or USE_ASYNC_CLIENT_SERVER):

[SWS_Dcm_00685] [

Name	SecurityAccess_{SecurityLevel}		
Comment	–		
IsService	true		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecurityUsePort)} == USE_ASYNC_CLIENT_SERVER SecurityLevel = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.SHORT-NAME)}		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	10	DCM_E_PENDING	Request is not yet finished. Further call(s) required to finish.
	11	DCM_E_COMPARE_KEY_FAILED	Key did not match.

Operation	CompareKey	
Comment	–	
Variation	–	
Parameters	Key	
	Type	Dcm_KeyArray_{SecurityLevel}Type





	Direction	IN
	Comment	Key, which needs to be compared
	Variation	SecurityLevel = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.SHORT-NAME)}
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	return Error Code
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_COMPARE_KEY_FAILED	

Operation	GetSecurityAttemptCounter	
Comment	Restore the attempt counter from the application	
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecurityAttemptCounterEnabled)} == TRUE	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	AttemptCounter	
	Type	uint8
	Direction	OUT
Comment	–	
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	GetSeed	
Comment	–	
Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecurityADRSize)) == NULL}	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–



△

	Variation	–
	Seed	
	Type	Dcm_SeedArray_{SecurityLevel}Type
	Direction	OUT
	Comment	Pointer for provided seed
	Variation	SecurityLevel = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.SHORT-NAME)}
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	GetSeed	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecurityADRSize)} != NULL)	
Parameters	SecurityAccessDataRecord	
	Type	Dcm_SecurityAccessDataRecordArray_{SecurityLevel}Type
	Direction	IN
	Comment	–
	Variation	SecurityLevel = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.SHORT-NAME)}
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	Seed	
	Type	Dcm_SeedArray_{SecurityLevel}Type
	Direction	OUT
	Comment	Pointer for provided seed
	Variation	SecurityLevel = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.SHORT-NAME)}
	ErrorCode	
Type	Dcm_NegativeResponseCodeType	
Direction	OUT	
Comment	–	
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	SetSecurityAttemptCounter	
Comment	Store the attempt counter in the application	
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecurityAttemptCounterEnabled)} == TRUE	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	AttemptCounter	
	Type	uint8
	Direction	IN
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

]()

8.8.3.2 DataServices_{Data}

Using the concepts of the SW-C template, the interface is defined as follows if ClientServer interface is used (DcmDspDataUsePort set to USE_DATA_SYNCH_CLIENT_SERVER or USE_DATA_ASYNCH_CLIENT_SERVER or USE_DATA_ASYNCH_CLIENT_SERVER_ERROR):

[SWS_Dcm_00686] [

Name	DataServices_{Data}		
Comment	–		
IsService	true		
Variation	((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_DATA_ELEMENT_SPECIFIC_INTERFACES)) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_ASYNCH_CLIENT_SERVER) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_ASYNCH_CLIENT_SERVER_ERROR) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01.DcmDspPidDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER)) Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)})		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	10	DCM_E_PENDING	Request is not yet finished. Further call(s) required to finish.

Operation	ConditionCheckRead	
Comment	–	
Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == (USE_DATA_ ASYNCH_CLIENT_SERVER USE_DATA_ASYNCH_CLIENT_SERVER_ERROR)) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidRead)) != NULL) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataConditionCheckReadFnc Used)) == TRUE)</pre>	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
Possible Errors	E_OK	
	E_NOT_OK	
	DCM_E_PENDING	

Operation	ConditionCheckRead	
Comment	–	
Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == USE_DATA_ SYNCH_CLIENT_SERVER) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDsp DidRead)) != NULL) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspData ConditionCheckReadFncUsed)) == TRUE)</pre>	
Parameters	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK	
	E_NOT_OK	

Operation	FreezeCurrentState	
Comment	–	
Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == (USE_DATA_ ASYNCH_CLIENT_SERVER USE_DATA_ASYNCH_CLIENT_SERVER_ERROR)) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidFreeze CurrentState)) == TRUE) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo Ref->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) != DCM_ CONTROLMASK_EXTERNAL)</pre>	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–





	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	FreezeCurrentState	
Comment	–	
Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == (USE_DATA_ ASYNCH_CLIENT_SERVER USE_DATA_ ASYNCH_CLIENT_SERVER_ERROR)) && (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidFreeze CurrentState)) == TRUE) && (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfo Ref->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) == DCM_ CONTROLMASK_EXTERNAL)</pre>	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	controlMask	
	Type	Dcm_ControlMask_{DID}Type
	Direction	IN
	Comment	–
	Variation	Data = (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME))
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
Comment	–	
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	FreezeCurrentState	
Comment	–	
Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == USE_DATA_ SYNCH_CLIENT_SERVER) && (ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDsp DidControl/DcmDspDidFreezeCurrentState)) == TRUE) && (ecuc(Dcm/DcmConfigSet/Dcm Dsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControl Mask)) != DCM_CONTROLMASK_EXTERNAL)</pre>	
Parameters	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–





	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	FreezeCurrentState	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidFreezeCurrentState)} == TRUE) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_CONTROLMASK_EXTERNAL)	
Parameters	controlMask	
	Type	Dcm_ControlMask_{DID}Type
	Direction	IN
	Comment	–
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)})
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	GetScalingInformation	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == (USE_DATA_ASYNCH_CLIENT_SERVER USE_DATA_ASYNCH_CLIENT_SERVER_ERROR)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataInfoRef->DcmDspDataScalingInfoSize)} != NULL)	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	ScalingInfo	
	Type	Dcm_ScalingInfoArray_{Data}Type
	Direction	OUT
	Comment	–
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)})
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–





Possible Errors	E_OK E_NOT_OK DCM_E_PENDING
------------------------	-----------------------------------

Operation	GetScalingInformation	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataInfoRef->DcmDspDataScalingInfoSize)} != NULL)	
Parameters	ScalingInfo	
	Type	Dcm_ScalingInfoArray_{Data}Type
	Direction	OUT
	Comment	–
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)})
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
Comment	–	
Variation	–	
Possible Errors	E_OK E_NOT_OK	

Operation	ReadData	
Comment	The server is not allowed to return E_NOT_OK, but shall always provide a valid data value (e.g. a default/replacement value in an error-case) to Dcm/Dem nevertheless the signature of the operation includes E_NOT_OK to ensure compatibility between server runnable and RTE Call API, since the RTE may return negative Std_Return values in certain cases (e.g. partition of server stopped)	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_ASYNCH_CLIENT_SERVER) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidRead)} != NULL)	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	Data	
	Type	Dcm_DataElement_{Data}Type
	Direction	OUT
	Comment	–
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)})
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	ReadData	
Comment	The server is not allowed to return E_NOT_OK, but shall always provide a valid data value (e.g. a default/replacement value in an error-case) to Dcm/Dem nevertheless the signature of the operation includes E_NOT_OK to ensure compatibility between server runnable and RTE Call API, since the RTE may return negative Std_Return values in certain cases (e.g. partition of server stopped)	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_ASYNCH_CLIENT_SERVER_ERROR) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidRead)} != NULL)	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	Data	
	Type	Dcm_DataElement_{Data}Type
	Direction	OUT
	Comment	–
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)})
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
Comment	–	
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	ReadData	
Comment	The server is not allowed to return E_NOT_OK, but shall always provide a valid data value (e.g. a default/replacement value in an error-case) to Dcm/Dem nevertheless the signature of the operation includes E_NOT_OK to ensure compatibility between server runnable and RTE Call API, since the RTE may return negative Std_Return values in certain cases (e.g. partition of server stopped)	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER) && (({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidRead)} != NULL) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01.DcmDspPidDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER))	
Parameters	Data	
	Type	Dcm_DataElement_{Data}Type
	Direction	OUT
	Comment	–
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)})
Possible Errors	E_OK E_NOT_OK	

Operation	ReadDataLength	
Comment	–	
Variation	<pre>{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == (USE_DATA_ ASYNCH_CLIENT_SERVER USE_DATA_ASYNCH_CLIENT_SERVER_ERROR)) && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidRead)) != NULL} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType) == UINT8_DYN)}</pre>	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataLength	
	Type	uint16
	Direction	OUT
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	ReadDataLength	
Comment	–	
Variation	<pre>{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == USE_DATA_ SYNCH_CLIENT_SERVER) && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDsp DidRead)) != NULL} && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType) == UINT8_DYN)}</pre>	
Parameters	DataLength	
	Type	uint16
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	ResetToDefault	
Comment	–	
Variation	<pre>{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == (USE_DATA_ ASYNCH_CLIENT_SERVER USE_DATA_ASYNCH_CLIENT_SERVER_ERROR)) && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidResetTo Default)) == TRUE) && {(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo Ref->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) != DCM_ CONTROLMASK_EXTERNAL)}</pre>	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–





	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	ResetToDefault	
Comment	–	
Variation	<pre>{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == (USE_DATA_ ASYNCH_CLIENT_SERVER USE_DATA_ASYNCH_CLIENT_SERVER_ERROR) && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidResetTo Default)} == TRUE) && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfo Ref->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_ CONTROLMASK_EXTERNAL)</pre>	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	controlMask	
	Type	Dcm_ControlMask_{DID}Type
	Direction	IN
	Comment	–
	Variation	Data = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	ResetToDefault	
Comment	–	
Variation	<pre>{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_ SYNCH_CLIENT_SERVER) && {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDsp DidControl/DcmDspDidResetToDefault)} == TRUE) && {ecuc(Dcm/DcmConfigSet/DcmDsp/ DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControl Mask)} != DCM_CONTROLMASK_EXTERNAL)</pre>	
Parameters	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–





	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	ResetToDefault	
Comment	–	
Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == USE_DATA_ SYNCH_CLIENT_SERVER) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDsp DidControl/DcmDspDidResetToDefault)) == TRUE) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/ DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControl Mask)) == DCM_CONTROLMASK_EXTERNAL)</pre>	
Parameters	controlMask	
	Type	Dcm_ControlMask_{DID}Type
	Direction	IN
	Comment	–
	Variation	Data = ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)))
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
Comment	–	
Variation	–	
Possible Errors	E_OK E_NOT_OK	

Operation	ReturnControlToECU	
Comment	–	
Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == (USE_DATA_ SYNCH_CLIENT_SERVER USE_DATA_ASYNCH_CLIENT_SERVER USE_DATA_ ASYNCH_CLIENT_SERVER_ERROR)) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid Info/DcmDspDidControl/DcmDspDidFreezeCurrentState)) == TRUE) ((ecuc(Dcm/DcmConfig Set/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidResetToDefault)) == TRUE) ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShort TermAdjustment)) == TRUE) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDid InfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) != DCM_ CONTROLMASK_EXTERNAL)</pre>	
Parameters	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	ReturnControlToECU	
Comment	–	





Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidFreezeCurrentState)) == TRUE) ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidResetToDefault)) == TRUE) ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)) == TRUE) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) == DCM_CONTROLMASK_EXTERNAL)</pre>	
Parameters	controlMask	
	Type	Dcm_ControlMask_{DID}Type
	Direction	IN
	Comment	–
	Variation	Data = ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)))
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
Possible Errors	E_OK	
	E_NOT_OK	

Operation	ShortTermAdjustment	
Comment	–	
Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == (USE_DATA_ASYNC_CLIENT_SERVER USE_DATA_ASYNC_CLIENT_SERVER_ERROR)) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)) == TRUE) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType) != UINT8_DYN) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) != DCM_CONTROLMASK_EXTERNAL))</pre>	
Parameters	ControlStateInfo	
	Type	Dcm_DataArrayTypeUint8_{Data}Type
	Direction	IN
	Comment	–
	Variation	Data = ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)))
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
Variation	–	
Possible Errors	E_OK	
	E_NOT_OK DCM_E_PENDING	

Operation	ShortTermAdjustment	
Comment	–	
Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == (USE_DATA_ ASYNCH_CLIENT_SERVER USE_DATA_ASYNCH_CLIENT_SERVER_ERROR)) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShort TermAdjustment)) == TRUE) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDsp DataType) != UINT8_DYN) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDid InfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) == DCM_ CONTROLMASK_EXTERNAL)</pre>	
Parameters	ControlStateInfo	
	Type	Dcm_DataArrayTypeUint8_{Data}Type
	Direction	IN
	Comment	–
	Variation	Data = ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)))
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	controlMask	
	Type	Dcm_ControlMask_{DID}Type
	Direction	IN
	Comment	–
	Variation	Data = ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)))
	ErrorCode	
Type	Dcm_NegativeResponseCodeType	
Direction	OUT	
Comment	–	
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	ShortTermAdjustment	
Comment	–	
Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == (USE_DATA_ ASYNCH_CLIENT_SERVER USE_DATA_ASYNCH_CLIENT_SERVER_ERROR)) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShort TermAdjustment)) == TRUE) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDsp DataType) == UINT8_DYN) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDid InfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)) != DCM_ CONTROLMASK_EXTERNAL)</pre>	
Parameters	ControlStateInfo	
	Type	Dcm_DataArrayTypeUint8_{Data}Type
	Direction	IN
	Comment	–
	Variation	Data = ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)))
	DataLength	
Type	uint16	





	Direction	IN
	Comment	–
	Variation	–
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	ShortTermAdjustment	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == (USE_DATA_ ASYNCH_CLIENT_SERVER USE_DATA_ ASYNCH_CLIENT_SERVER_ERROR)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShort TermAdjustment)} == TRUE) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDsp Data Type)} == UINT8_DYN) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDid InfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_ CONTROLMASK_EXTERNAL)	
Parameters	ControlStateInfo	
	Type	Dcm_DataArrayTypeUint8_{Data}Type
	Direction	IN
	Comment	–
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)})
	DataLength	
	Type	uint16
	Direction	IN
	Comment	–
	Variation	–
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	controlMask	
	Type	Dcm_ControlMask_{DID}Type
	Direction	IN
Comment	–	
Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)})	





	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	ShortTermAdjustment	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)} == TRUE) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType) != UINT8_DYN) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} != DCM_CONTROLMASK_EXTERNAL)	
Parameters	ControlStateInfo	
	Type	Dcm_DataArrayTypeUint8_{Data}Type
	Direction	IN
	Comment	–
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)})
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
Possible Errors	E_OK E_NOT_OK	

Operation	ShortTermAdjustment	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)} == TRUE) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType) != UINT8_DYN) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_CONTROLMASK_EXTERNAL)	
Parameters	ControlStateInfo	
	Type	Dcm_DataArrayTypeUint8_{Data}Type
	Direction	IN
	Comment	–
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)})
	controlMask	
	Type	Dcm_ControlMask_{DID}Type
	Direction	IN
Comment	–	





	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)})
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	ShortTermAdjustment	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)} == TRUE) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType)} == UINT8_DYN) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} != DCM_CONTROLMASK_EXTERNAL)	
Parameters	ControlStateInfo	
	Type	Dcm_DataArrayTypeUint8_{Data}Type
	Direction	IN
	Comment	–
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)})
	DataLength	
	Type	uint16
	Direction	IN
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
Variation	–	
Possible Errors	E_OK E_NOT_OK	

Operation	ShortTermAdjustment	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_SYNCH_CLIENT_SERVER) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl/DcmDspDidShortTermAdjustment)} == TRUE) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType)} == UINT8_DYN) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl/DcmDspDidControlMask)} == DCM_CONTROLMASK_EXTERNAL)	
Parameters	ControlStateInfo	
	Type	Dcm_DataArrayTypeUint8_{Data}Type
	Direction	IN





	Comment	–
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)})
	DataLength	
	Type	uint16
	Direction	IN
	Comment	–
	Variation	–
	controlMask	
	Type	Dcm_ControlMask_{DID}Type
	Direction	IN
	Comment	–
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)})
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
Comment	–	
Variation	–	
Possible Errors	E_OK E_NOT_OK	

Operation	WriteData	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == (USE_DATA_ASYNC_CLIENT_SERVER USE_DATA_ASYNC_CLIENT_SERVER_ERROR)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidWrite)} != NULL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType)} != UINT8_DYN)	
Parameters	Data	
	Type	Dcm_DataElement_{Data}Type
	Direction	IN
	Comment	–
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)})
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
Comment	–	
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	WriteData	
Comment	–	
Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == (USE_DATA_ ASYNCH_CLIENT_SERVER USE_DATA_ASYNC_CLIENT_SERVER_ERROR)) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidWrite)) != NULL) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType) == UINT8_DYN)</pre>	
Parameters	Data	
	Type	Dcm_DataElement_{Data}Type
	Direction	IN
	Comment	–
	Variation	Data = ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)))
	DataLength	
	Type	uint16
	Direction	IN
	Comment	–
	Variation	–
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
Variation	–	
ErrorCode		
Type	Dcm_NegativeResponseCodeType	
Direction	OUT	
Comment	–	
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	WriteData	
Comment	–	
Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)) == USE_DATA_ SYNCH_CLIENT_SERVER) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDsp DidWrite)) != NULL) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType) != UINT8_DYN)</pre>	
Parameters	Data	
	Type	Dcm_DataElement_{Data}Type
	Direction	IN
	Comment	–
	Variation	Data = ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)))
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–





Possible Errors	E_OK E_NOT_OK
------------------------	------------------

Operation	WriteData	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataUsePort)} == USE_DATA_SYNC_CLIENT_SERVER) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidWrite)} != NULL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.DcmDspDataType)} == UINT8_DYN)	
Parameters	Data	
	Type	Dcm_DataElement_{Data}Type
	Direction	IN
	Comment	–
	Variation	Data = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)})
	DataLength	
	Type	uint16
	Direction	IN
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
Variation	–	
Possible Errors	E_OK E_NOT_OK	

⌋()

One DataServices interface will be generated for each Data of each DID/PID, with following possible operations:

8.8.3.2.1 ReadData

ReadData allows requesting to the application a data value of a DID/PID. A ReadData interface is defined for every data of each DID/PID with read access. The Data specific type is an array of uint8 which represents either the fix length of this Data or the maximum possible length of this Data. If the length is variable, the operation ReadDataLength has to provide the current valid data length of this Data.

This interface is used for [UDS Service ReadDataByIdentifier](#) and for [UDS Service ReadDataByPeriodicIdentifier \(0x2A\)](#) and for [UDS Service InputOutputControlByIdentifier \(0x2F\)](#).

The ReadData interface can be defined as synchronous or asynchronous according to configuration parameter [DcmDspDataUsePort](#). The synchronous mechanism of the

ReadData interface is compatible to the related DEM interface to allow the provider to use the same interface for both [Dcm](#) and DEM.

8.8.3.2.2 WriteData

WriteData requests the application to write a data value of a DID. The Data specific type is an array of uint8 which represent either the fix length of this Data or the maximum possible length of this Data. A WriteData interface is defined for every data of each [DID](#) with write access. This interface is used for the [UDS](#) Service WriteDataByIdentifier (0x2E).

8.8.3.2.3 ReadDataLength

ReadDataLength requests the application to return the data length of a Data. A ReadDataLength interface is defined for every data of each [DID](#) with variable data length. This interface is used for [UDS](#) Service ReadDataByIdentifier and for [UDS](#) Service ReadDataByPeriodicIdentifier (0x2A).

8.8.3.2.4 ConditionCheckRead

ConditionCheckRead requests to the application if the conditions (System state,...) to read the Data are correct. This operation is called for all requested DIDs before requesting the data of each DID. A ConditionCheckRead interface is defined for every data of each [DID](#) with read access.

8.8.3.2.5 GetScalingInformation

Request to the application for the scaling information of a Data (see [\[SWS_Dcm_00394\]](#)).

8.8.3.2.6 ReturnControlToEcu

Request to the application to return control to ECU of an IOControl (see [\[SWS_Dcm_00396\]](#)).

8.8.3.2.7 ResetToDefault

Request to the application to reset an IOControl to default value (see [\[SWS_Dcm_00397\]](#)).

8.8.3.2.8 FreezeCurrentState

Request to the application to freeze the current state of an IOControl (see [SWS_Dcm_00398]).

8.8.3.2.9 ShortTermAdjustment

Request to the application to adjust the IO signal (see [SWS_Dcm_00399]).

8.8.3.3 DataServices_DIDRange_{Range}

The following interface defines an operation needed to get the DID range. Using the concepts of the SW-C template, the interface is defined as follows:

[SWS_Dcm_00769] [

Name	DataServices_DIDRange_{Range}		
Comment	–		
IsService	true		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRange.DcmDspDidRangeUsePort)} == TRUE Range = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRange.SHORT-NAME)})		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	10	DCM_E_PENDING	Request is not yet finished. Further call(s) required to finish.

Operation	IsDidAvailable		
Comment	–		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRange.DcmDspDidRangeHasGaps)} == TRUE		
Parameters	DID		
	Type	uint16	
	Direction	IN	
	Comment	–	
	Variation	–	
	OpStatus		
	Type	Dcm_OpStatusType	
	Direction	IN	
	Comment	–	
	Variation	–	
	supported		
	Type	Dcm_DidSupportedType	
	Direction	OUT	





	Comment	–
	Variation	–
Possible Errors	E_OK DCM_E_PENDING	

Operation	ReadDidData	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRange/DcmDspDidRangeInfoRef->DcmDspDidRead)} != NULL)	
Parameters	DID	
	Type	uint16
	Direction	IN
	Comment	–
	Variation	–
	Data	
	Type	Dcm_RangeArray_{Range}Type
	Direction	OUT
	Comment	–
	Variation	Range = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRange.SHORT-NAME)})
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataLength	
	Type	uint16
	Direction	OUT
	Comment	–
	Variation	–
ErrorCode		
Type	Dcm_NegativeResponseCodeType	
Direction	OUT	
Comment	–	
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	



Operation	ReadDidRangeDataLength	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRange/DcmDspDidRangeInfoRef->DcmDspDidRead)} != NULL)	
Parameters	DID	



	Type	uint16
	Direction	IN
	Comment	–
	Variation	–
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataLength	
	Type	uint16
	Direction	OUT
	Comment	–
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	WriteDidData	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRange/DcmDspDidRangeInfoRef->DcmDspDidWrite)} != NULL)	
Parameters	DID	
	Type	uint16
	Direction	IN
	Comment	–
	Variation	–
	Data	
	Type	Dcm_RangeArray_{Range}Type
	Direction	IN
	Comment	–
	Variation	Range = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRange.SHORT-NAME)})
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataLength	
	Type	uint16
	Direction	IN
	Comment	–
Variation	–	
ErrorCode		





	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

⌋()

8.8.3.4 InfotypeServices_{VehInfoData}

The following interface defines an operation needed to get data from one or several SW-C in order to supply OBD Service \$09 (see [SWS_Dcm_00423]).

Using the concepts of the SW-C template, the interface is defined as follows:

[SWS_Dcm_00688] [

Name	InfotypeServices_{VehInfoData}		
Comment	–		
IsService	true		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData/DcmDspVehInfoDataUsePort)}==TRUE VehInfoData = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData.SHORT-NAME)}		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	10	DCM_E_PENDING	Request is not yet finished. Further call(s) required to finish.

Operation	GetInfotypeValueData	
Comment	–	
Variation	–	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataValueBuffer	
	Type	Dcm_InfoTypeServicesArray_{VehInfoData}Type
	Direction	OUT
	Variation	VehInfoData = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData.SHORT-NAME)}





	DataValueBufferSize	
	Type	uint8
	Direction	INOUT
	Comment	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 DataValueBuffer
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

]()

8.8.3.5 RoutineServices_{RoutineName}

The following interface defines operations needed for the [UDS Service RoutineControl \(0x31\)](#) (see [[SWS_Dcm_00400](#)], [[SWS_Dcm_00401](#)], [[SWS_Dcm_00402](#)], [[SWS_Dcm_00403](#)], [[SWS_Dcm_00404](#)], [[SWS_Dcm_00405](#)]).

Using the concepts of the [SW-C](#) template, the interface is defined as follows:

[[SWS_Dcm_00690](#)] [

Name	RoutineServices_{RoutineName}		
Comment	–		
IsService	true		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.DcmDspRoutineUsePort)} == TRUE RoutineName = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	10	DCM_E_PENDING	Request is not yet finished. Further call(s) required to finish.
	12	DCM_E_FORCE_RCRRP	application request the transmission of a response Response Pending (NRC 0x78)

Operation	RequestResults	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH) &&({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH)	
Parameters	DataIn_{Signal}	
	Type	Dcm_RequestDataIn_{Routine}_{Signal}Type
	Direction	IN





	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)}Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
	Type	Dcm_RequestDataOut_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)}Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	RequestResults	
Comment	–	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH) &&({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH)	
Parameters	DataIn_{Signal}	
	Type	Dcm_RequestDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)}Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–





	Variation	–
	DataOut_{Signal}	
	Type	Dcm_RequestDataOut_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataOut_{Signal}	
	Type	Dcm_RequestFlexibleOutArrayData_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	currentDataLength	
	Type	uint16
	Direction	OUT
	Comment	–
Variation	–	
ErrorCode		
Type	Dcm_NegativeResponseCodeType	
Direction	OUT	
Comment	–	
Variation	–	
Possible Errors	–	

Operation	RequestResults	
Comment	–	
Variation	{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH) &&{(ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH)}	
Parameters	DataIn_{Signal}	
	Type	Dcm_RequestDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–





	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)}Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataIn_{Signal}	
	Type	Dcm_RequestFlexibleInArrayData_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)}Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
	Type	Dcm_RequestDataOut_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)}Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors		–

Operation	RequestResults	
Comment	–	
Variation	(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH) &&({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH))	
Parameters	DataIn_{Signal}	
	Type	Dcm_RequestDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsIn/DcmDspRequestRoutineResultsInSignal.SHORT-NAME)}Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}



△

	DataIn_{Signal}
Type	Dcm_StartFlexibleInArrayData_{Routine}_{Signal}Type
Direction	IN
Comment	–
Variation	–
	OpStatus
Type	Dcm_OpStatusType
Direction	IN
Comment	–
Variation	–
	DataOut_{Signal}
Type	Dcm_RequestDataOut_{Routine}_{Signal}Type
Direction	OUT
Comment	–
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataOut_{Signal}
Type	Dcm_RequestFlexibleOutArrayData_{Routine}_{Signal}Type
Direction	OUT
Comment	–
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	currentDataLength
Type	uint16
Direction	OUT
Comment	–
Variation	–
	ErrorCode
Type	Dcm_NegativeResponseCodeType
Direction	OUT
Comment	–
Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP

Operation	RequestResultsConfirmation	
Comment	This operation indicates the transmission of a response to a RequestResultsRoutine request	
Variation	((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsConfirmationEnabled))==TRUE)	
Parameters	ConfirmationStatus	
	Type	Dcm_ConfirmationStatusType
	Direction	IN
	Comment	Confirmation status of a RequestResultsRoutinerequest
	Variation	–
Possible Errors	E_OK E_NOT_OK	

Operation	Start	
Comment	–	
Variation	((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)) != VARIABLE_LENGTH) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)) != VARIABLE_LENGTH)	
Parameters	DataIn_{Signal}	
	Type	Dcm_StartDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
	Type	Dcm_StartDataOut_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
Direction	OUT	
Comment	–	
Variation	–	





Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP
------------------------	--

Operation	Start	
Comment	–	
Variation	((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)) != VARIABLE_LENGTH) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)) == VARIABLE_LENGTH)	
Parameters	DataIn_{Signal}	
	Type	Dcm_StartDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)} Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)}
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
	Type	Dcm_StartDataOut_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataOut_{Signal}	
	Type	Dcm_StartFlexibleOutArrayData_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	–
currentDataLength		
Type	uint16	
Direction	OUT	
Comment	–	
Variation	–	
ErrorCode		
Type	Dcm_NegativeResponseCodeType	





	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	Start	
Comment	–	
Variation	((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)) == VARIABLE_LENGTH) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)) != VARIABLE_LENGTH)	
Parameters	DataIn_{Signal}	
	Type	Dcm_StartDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataIn_{Signal}	
	Type	Dcm_StartFlexibleInArrayData_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	–
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
	Type	Dcm_StartDataOut_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
currentDataLength		
Type	uint16	
Direction	IN	
Comment	–	





	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	Start		
Comment	–		
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH)		
Parameters	DataIn_{Signal}		
	Type	Dcm_StartDataIn_{Routine}_{Signal}Type	
	Direction	IN	
	Comment	–	
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
	DataIn_{Signal}		
	Type	Dcm_StartFlexibleInArrayData_{Routine}_{Signal}Type	
	Direction	IN	
	Comment	–	
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
	OpStatus		
	Type	Dcm_OpStatusType	
	Direction	IN	
Comment	–		
Variation	–		
DataOut_{Signal}			
Type	Dcm_StartDataOut_{Routine}_{Signal}Type		



△

	Direction	OUT
	Comment	–
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataOut_{Signal}	
	Type	Dcm_StartFlexibleOutArrayData_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	currentDataLength	
	Type	uint16
	Direction	INOUT
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
Direction	OUT	
Comment	–	
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	StartConfirmation	
Comment	This operation indicates the transmission of a response to a StartRoutine request	
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineConfirmationEnabled)}==TRUE	
Parameters	ConfirmationStatus	
	Type	Dcm_ConfirmationStatusType
	Direction	IN
	Comment	Confirmation status of a StartRoutine request
Variation	–	
Possible Errors	E_OK E_NOT_OK	

Operation	Stop	
Comment	–	
Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)) != VARIABLE_LENGTH) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)) != VARIABLE_LENGTH)</pre>	
Parameters	DataIn_{Signal}	
	Type	Dcm_StopDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	<pre>Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}</pre>
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
	Type	Dcm_StopDataOut_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
Variation	<pre>Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}</pre>	
ErrorCode		
Type	Dcm_NegativeResponseCodeType	
Direction	OUT	
Comment	–	
Variation	–	
Possible Errors	<pre>E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP</pre>	

Operation	Stop
Comment	–
Variation	<pre>((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)) != VARIABLE_LENGTH) && ((ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)) == VARIABLE_LENGTH)</pre>
Parameters	DataIn_{Signal}
	Type





	Direction	IN
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
	Type	Dcm_StopDataOut_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataOut_{Signal}	
	Type	Dcm_StopFlexibleOutArrayData_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	–
	currentDataLength	
	Type	uint16
	Direction	OUT
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRPP	

Operation	Stop
Comment	–
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH)





Parameters	DataIn_{Signal}	
	Type	Dcm_StopDataIn_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	DataIn_{Signal}	
	Type	Dcm_StopFlexibleInArrayData_{Routine}_{Signal}Type
	Direction	IN
	Comment	–
	Variation	–
	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	–
	Variation	–
	DataOut_{Signal}	
	Type	Dcm_StopDataOut_{Routine}_{Signal}Type
	Direction	OUT
	Comment	–
	Variation	Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	currentDataLength	
Type	uint16	
Direction	IN	
Comment	–	
Variation	–	
ErrorCode		
Type	Dcm_NegativeResponseCodeType	
Direction	OUT	
Comment	–	
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	Stop
Comment	–





Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH)		
Parameters	DataIn_{Signal}		
	Type	Dcm_StopDataIn_{Routine}_{Signal}Type	
	Direction	IN	
	Comment	–	
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
	DataIn_{Signal}		
	Type	Dcm_StopFlexibleInArrayData_{Routine}_{Signal}Type	
	Direction	IN	
	Comment	–	
	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}	
	OpStatus		
	Type	Dcm_OpStatusType	
	Direction	IN	
	Comment	–	
	Variation	–	
	DataOut_{Signal}		
Type	Dcm_StopDataOut_{Routine}_{Signal}Type		
Direction	OUT		
Comment	–		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} != VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}		
DataOut_{Signal}			
Type	Dcm_StopFlexibleOutArrayData_{Routine}_{Signal}Type		
Direction	OUT		
Comment	–		





	Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.DcmDspRoutineSignalType)} == VARIABLE_LENGTH Signal = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal.SHORT-NAME)} Routine = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}
	currentDataLength	
	Type	uint16
	Direction	INOUT
	Comment	–
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	–
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	StopConfirmation	
Comment	This operation indicates the transmission of a response to a StopRoutine request	
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineConfirmationEnabled)}==TRUE	
Parameters	ConfirmationStatus	
	Type	Dcm_ConfirmationStatusType
	Direction	IN
	Comment	Confirmation status of a StopRoutine request
Variation	–	
Possible Errors	E_OK E_NOT_OK	

]()

From the point of view of the DCM, the operations have the following signatures:

8.8.3.6 RequestControlServices_{Tid}

The following interface allows the [Dcm](#) to provide [OBD](#) Service \$08 (see [\[SWS_Dcm_00419\]](#)).

Using the concepts of the [SW-C](#) template, the interface is defined as follows:

[\[SWS_Dcm_00691\]](#) [

Name	RequestControlServices_{Tid}		
Comment	–		
IsService	true		
Variation	Tid = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl.SHORT-NAME)}		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	RequestControl		
Comment	–		
Variation	–		
Parameters	OutBuffer		
	Type	Dcm_RequestControlServicesOutArray_{Tid}Type	
	Direction	OUT	
	Comment	–	
	Variation	Tid = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl.SHORT-NAME)}	
	InBuffer		
	Type	Dcm_RequestControlServicesInArray_{Tid}Type	
	Direction	IN	
Possible Errors	E_OK		
	E_NOT_OK		

⌋()

8.8.3.7 CallbackDCMRequestServices

The following interface provides information on the status of the protocol communication and allows the Application to disallow a protocol (see [\[SWS_Dcm_00036\]](#), [\[SWS_Dcm_00144\]](#), [\[SWS_Dcm_00145\]](#), [\[SWS_Dcm_00146\]](#); [\[SWS_Dcm_00147\]](#), [\[SWS_Dcm_00459\]](#)).

Using the concepts of the *SW-C* template, the interface is defined as follows:

[SWS_Dcm_00692] [

Name	CallbackDCMRequestServices		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed





	5	E_PROTOCOL_NOT_ALLOWED	conditions in application allows no further procession of protocol
--	---	------------------------	--

Operation	StartProtocol		
Comment	–		
Variation	–		
Parameters	ProtocolType		
	Type	Dcm_ProtocolType	
	Direction	IN	
	Comment	–	
	Variation	–	
	TesterSourceAddress		
	Type	uint16	
	Direction	IN	
	Comment	–	
	Variation	–	
	ConnectionId		
	Type	uint16	
	Direction	IN	
Comment	–		
Variation	–		
Possible Errors	E_OK E_NOT_OK E_PROTOCOL_NOT_ALLOWED		

Operation	StopProtocol		
Comment	–		
Variation	–		
Parameters	ProtocolType		
	Type	Dcm_ProtocolType	
	Direction	IN	
	Comment	–	
	Variation	–	
	TesterSourceAddress		
	Type	uint16	
	Direction	IN	
	Comment	–	
	Variation	–	
	ConnectionId		
	Type	uint16	
	Direction	IN	
Comment	–		
Variation	–		





Possible Errors	E_OK E_NOT_OK
------------------------	------------------

}]()

8.8.3.8 ServiceRequestNotification

The following interface indicates to the Application that a service is about to be executed and allows the Application to reject the execution of the service request (see [SWS_Dcm_00218], [SWS_Dcm_00462], [SWS_Dcm_00463]).

Using the concepts of the SW-C template, the interface is defined as follows:

[SWS_Dcm_00694] [

Name	ServiceRequestNotification		
Comment	–		
IsService	true		
Variation	({{ecuc(Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceRequestManufacturerNotification)}} != NULL) ({{ecuc(Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceRequestSupplierNotification)}} != NULL)		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	8	E_REQUEST_NOT_ACCEPTED	no response will be sent

Operation	Confirmation	
Comment	–	
Variation	–	
Parameters	SID	
	Type	uint8
	Direction	IN
	Comment	Value of service identifier
	Variation	–
	ReqType	
	Type	uint8
	Direction	IN
	Comment	Addressing type of the request(0=physical request, 1=functional request)
	Variation	–
	ConnectionId	
	Type	uint16
	Direction	IN





	Comment	Unique connection identifier
	Variation	–
	ConfirmationStatus	
	Type	Dcm_ConfirmationStatusType
	Direction	IN
	Comment	Confirmation of a successful transmission or a transmission error of a diagnostic service.
	Variation	–
	ProtocolType	
	Type	Dcm_ProtocolType
	Direction	IN
	Comment	–
	Variation	–
	TesterSourceAddress	
	Type	uint16
	Direction	IN
Comment	–	
Variation	–	
Possible Errors	E_OK E_NOT_OK	

Operation	Indication	
Comment	–	
Variation	–	
Parameters	SID	
	Type	uint8
	Direction	IN
	Comment	Value of service identifier
	Variation	–
	RequestData	
	Type	Dcm_RequestDataArrayType
	Direction	IN
	Comment	This parameter contains the complete request data (diagnostic buffer), except the service ID
	Variation	–
	DataSize	
	Type	uint16
	Direction	IN
	Comment	This parameter defines how many bytes in the RequestData parameter are valid
	Variation	–
ReqType		
Type	uint8	
Direction	IN	



△

	Comment	Addressing type of the request(0=physical request, 1=functional request)		
	Variation	–		
	ConnectionId			
	Type	uint16		
	Direction	IN		
	Comment	Unique connection identifier		
	Variation	–		
	ErrorCode			
	Type	Dcm_NegativeResponseCodeType		
	Direction	OUT		
	Comment	–		
	Variation	–		
	ProtocolType			
	Type	Dcm_ProtocolType		
	Direction	IN		
	Comment	–		
	Variation	–		
	TesterSourceAddress			
	Type	uint16		
	Direction	IN		
Comment	–			
Variation	–			
Possible Errors	E_OK E_NOT_OK E_REQUEST_NOT_ACCEPTED			

⌋()

8.8.3.9 UploadDownloadServices

[SWS_Dcm_91065] [

Name	UploadDownloadServices		
Comment	–		
IsService	true		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspMemoryTransfer.DcmDspMemoryTransferUsePort)} == TRUE)		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	10	DCM_E_PENDING	Request is not yet finished. Further call(s) required to finish.
	12	DCM_E_FORCE_RCRRP	application request the transmission of a response Response Pending (NRC 0x78)

Operation	ProcessRequestDownload	
Comment	<p>Callout function.</p> <p>DCM shall call this callout function to start a download process.</p> <p>This service is needed for the implementation of UDS service RequestDownload.</p>	
Variation	–	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	DCM_INITIAL DCM_PENDING DCM_CANCEL
	Variation	–
	DataFormatIdentifier	
	Type	uint8
	Direction	IN
	Comment	<p>Bit 7 - 4: Compression Method</p> <ul style="list-style-type: none"> • 0x0: not compressed • 0x1..F: vehicle-manufacturer-specific <p>Bit 3 - 0: Encrypting method</p> <ul style="list-style-type: none"> • 0x0: not encrypted • 0x1..F: vehicle-manufacturer-specific
	Variation	–
	MemoryIdentifier	
	Type	uint8
	Direction	IN
	Comment	Identifier of the Memory Block, if the parameter is not used it shall be set to 0.
	Variation	–
	MemoryAddress	
	Type	uint32
	Direction	IN
	Comment	Starting address of server memory to which data is to be written
	Variation	–
	MemorySize	
Type	uint32	
Direction	IN	
Comment	Uncompressed memory size in bytes	
Variation	–	
BlockLength		
Type	uint32	
Direction	INOUT	





	Comment	Max. Number of bytes for one Dcm_WriteMemory
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	If the operation Dcm_ProcessRequestDownload returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	ProcessRequestTransferExit	
Comment	Callout function. DCM shall call this callout function to terminate a download or upload process. This callout is needed for the implementation of UDS service RequestTransferExit.	
Variation	–	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	DCM_INITIAL DCM_PENDING DCM_CANCEL
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	see below
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING	

Operation	ProcessRequestUpload	
Comment	Callout function. DCM shall call this callout function to start an upload process. This service is needed for the implementation of UDS service RequestUpload.	
Variation	–	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	DCM_INITIAL DCM_PENDING DCM_CANCEL DCM_FORCE_RCRRP_OK
	Variation	–





	Variation	–
	DataFormatIdentifier	
	Type	uint8
	Direction	IN
	Comment	<p>Bit 7 - 4: Compression Method</p> <ul style="list-style-type: none"> • 0x0: not compressed • 0x1..F: vehicle-manufacturer-specific <p>Bit 3 - 0: Encrypting method</p> <ul style="list-style-type: none"> • 0x0: not encrypted • 0x1..F: vehicle-manufacturer-specific
	Variation	–
	MemoryIdentifier	
	Type	uint8
	Direction	IN
	Comment	Identifier of the Memory Block, if the parameter is not used it shall be set to 0.
	Variation	–
	MemoryAddress	
	Type	uint32
	Direction	IN
	Comment	Starting address of server memory from which data are to be copied
	Variation	–
	MemorySize	
	Type	uint32
	Direction	IN
	Comment	Uncompressed memory size in bytes
	Variation	–
	BlockLength	
	Type	uint32
	Direction	INOUT
	Comment	Max. Number of bytes for one Dcm_ReadMemory
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	If the operation Dcm_ProcessRequestUpload returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Variation	–	





Possible Errors	E_OK E_NOT_OK DCM_E_PENDING
------------------------	-----------------------------------

Operation	ProcessTransferDataRead	
Comment	<p>The ProcessTransferDataRead callout is used to request memory data identified by the parameter memoryAddress and memorySize from the UDS request message.</p> <p>This service is needed for the implementation of UDS services:</p> <ul style="list-style-type: none"> • ReadMemoryByAddress • RequestUpload • ReadDataByIdentifier (in case of Dynamical DID defined by memory address) • TransferData 	
Variation	–	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	DCM_INITIAL DCM_PENDING DCM_CANCEL DCM_FORCE_RCRRP_OK
	Variation	–
	MemoryIdentifier	
	Type	uint8
	Direction	IN
	Comment	Identifier of the Memory Block (e.g. used if memory section distinguishing is needed) Note: If it's not used this parameter shall be set to 0.
	Variation	–
	MemoryAddress	
	Type	uint32
	Direction	IN
	Comment	Starting address of server memory from which data is to be retrieved.
	Variation	–
	MemorySize	
	Type	uint32
	Direction	IN
	Comment	Number of bytes in the MemoryData
	Variation	–
MemoryData		
Type	Dcm_RequestDataArrayType	
Direction	OUT	
Comment	Data read (Points to the diagnostic buffer in DCM)	
Variation	–	
ErrorCode		





	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	If the operation Dcm_ReadMemory returns value DCM_READ_FAILED, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	ProcessTransferDataWrite	
Comment	<p>The ProcessTransferDataWrite callout is used to write memory data identified by the parameter memoryAddress and memorySize. This service is needed for the implementation of UDS services :</p> <ul style="list-style-type: none"> • WriteMemoryByAddress • RequestDownload • TransferData 	
Variation	–	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	DCM_INITIAL DCM_PENDING DCM_CANCEL DCM_FORCE_RCRRP_OK
	Variation	–
	MemoryIdentifier	
	Type	uint8
	Direction	IN
	Comment	Identifier of the Memory Block (e.g. used by WriteDataByIdentifier service). Note: If it's not used this parameter shall be set to 0.
	Variation	–
	MemoryAddress	
	Type	uint32
	Direction	IN
	Comment	Starting address of server memory in which data is to be copied. Note: If it's not used (e.g. if the data is compressed) this parameter shall be set to 0.
	Variation	–
	MemorySize	
Type	uint32	
Direction	IN	
Comment	Number of bytes in MemoryData	
Variation	–	





	MemoryData	
	Type	Dcm_RequestDataArrayType
	Direction	IN
	Comment	Data to write (Points to the diagnostic buffer in DCM)
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	If the operation Dcm_WriteMemory returns value DCM_WRITE_FAILED, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

]()

8.8.3.10 RequestFileTransfer

[SWS_Dcm_91086] [

Name	RequestFileTransfer		
Comment	–		
IsService	true		
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRequestFileTransfer/DcmRequestFileTransfer UsePort)} == TRUE)		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	10	DCM_E_PENDING	Request is not yet finished. Further call(s) required to finish.
	12	DCM_E_FORCE_RCRRP	application request the transmission of a response Response Pending (NRC 0x78)

Operation	ProcessRequestAddFile	
Comment	Callout function. DCM shall call this function to start a RequestFileTransfer process with modeOfOperation equal to 0x01 (AddFile).	
Variation	–	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN





Comment	DCM_INITIAL: All In-parameters are valid. DCM_PENDING: All In-parameters are set to 0x00. DCM_CANCEL: All In-parameters are set to 0x00. DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x00.
Variation	–
filePathAndNameLength	
Type	uint16
Direction	IN
Comment	Defines the length in bytes for the parameter filePathAndName.
Variation	–
filePathAndName	
Type	const Dcm_FileAndDirNameType
Direction	IN
Comment	Defines the file system location of the server where the file which shall be added, deleted, replaced or read from depending on the parameter modeOfOperation parameter. In addition this parameter includes the file name of the file which shall be added, deleted, replaced or read as part of the file path.
Variation	–
dataFormatIdentifier	
Type	uint8
Direction	IN
Comment	This data-parameter is a one byte value with each nibble encoded separately. The high nibble specifies the "compressionMethod", and the low nibble specifies the "encryptingMethod". The value 0x00 specifies that neither compressionMethod nor encryptingMethod is used. Values other than 0x00 are vehicle manufacturer specific.
Variation	–
fileSizeUncompressed	
Type	uint64
Direction	IN
Comment	Defines the size of the uncompressed file to be download in bytes.
Variation	–
fileSizeCompressed	
Type	uint64
Direction	IN
Comment	Defines the size of the compressed file to be downloaded in bytes.
Variation	–
maxNumberOfBlockLength	
Type	uint64
Direction	OUT
Comment	Max number of bytes to be included in each TransferData request excluding the SID and the blockSequenceCounter.
Variation	–
ErrorCode	
Type	Dcm_NegativeResponseCodeType





	Direction	OUT
	Comment	If the operation Dcm_ProcessRequestAddFile returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	ProcessRequestDeleteFile	
Comment	Callout function. DCM shall call this function to start a RequestFileTransfer process with modeOfOperation equal to 0x02 (DeleteFile).	
Variation	–	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	DCM_INITIAL: All In-parameters are valid. DCM_PENDING: All In-parameters are set to 0x00. DCM_CANCEL: All In-parameters are set to 0x00. DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x00.
	Variation	–
	filePathAndNameLength	
	Type	uint16
	Direction	IN
	Comment	Defines the length in bytes for the parameter filePathAndName.
	Variation	–
	filePathAndName	
	Type	const Dcm_FileAndDirNameType
	Direction	IN
	Comment	Defines the file system location of the server where the file which shall be added, deleted, replaced or read from depending on the parameter modeOfOperation parameter. In addition this parameter includes the file name of the file which shall be added, deleted, replaced or read as part of the file path.
Variation	–	
ErrorCode		
Type	Dcm_NegativeResponseCodeType	
Direction	OUT	
Comment	If the operation Dcm_ProcessRequestDeleteFile returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.	
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	ProcessRequestReadDir		
Comment	Callout function. DCM shall call this function to start a RequestFileTransfer process with modeOfOperation equal to 0x05 (ReadDir).		
Variation	–		
Parameters	OpStatus		
	Type	Dcm_OpStatusType	
	Direction	IN	
	Comment	DCM_INITIAL: All In-parameters are valid. DCM_PENDING: All In-parameters are set to 0x00. DCM_CANCEL: All In-parameters are set to 0x00. DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x00.	
	Variation	–	
	filePathAndNameLength		
	Type	uint16	
	Direction	IN	
	Comment	Defines the length in bytes for the parameter filePathAndName.	
	Variation	–	
	filePathAndName		
	Type	const Dcm_FileAndDirNameType	
	Direction	IN	
	Comment	Defines the file system location of the server where the file which shall be added, deleted, replaced or read from depending on the parameter modeOfOperation parameter. In addition this parameter includes the file name of the file which shall be added, deleted, replaced or read as part of the file path.	
	Variation	–	
	dirInfoLength		
	Type	uint64	
	Direction	OUT	
	Comment	Defines the size of directory information to be uploaded in bytes.	
	Variation	–	
	maxNumberOfBlockLength		
	Type	uint64	
	Direction	OUT	
Comment	Max number of bytes to be included in each TransferData request excluding the SID and the blockSequenceCounter.		
Variation	–		
ErrorCode			
Type	Dcm_NegativeResponseCodeType		
Direction	OUT		
Comment	If the operation Dcm_ProcessRequestReadDir returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.		
Variation	–		
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP		

Operation	ProcessRequestReadFile	
Comment	<p>Callout function.</p> <p>DCM shall call this function to start a RequestFileTransfer process with modeOfOperation equal to 0x04 (ReadFile).</p>	
Variation	–	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	DCM_INITIAL: All In-parameters are valid. DCM_PENDING: All In-parameters are set to 0x00. DCM_CANCEL: All In-parameters are set to 0x00. DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x00.
	Variation	–
	filePathAndNameLength	
	Type	uint16
	Direction	IN
	Comment	Defines the length in bytes for the parameter filePathAndName.
	Variation	–
	filePathAndName	
	Type	const Dcm_FileAndDirNameType
	Direction	IN
	Comment	Defines the file system location of the server where the file which shall be added, deleted, replaced or read from depending on the parameter modeOfOperation parameter. In addition this parameter includes the file name of the file which shall be added, deleted, replaced or read as part of the file path.
	Variation	–
	dataFormatIdentifier	
	Type	uint8
	Direction	IN
	Comment	This data-parameter is a one byte value with each nibble encoded separately. The high nibble specifies the "compressionMethod", and the low nibble specifies the "encryptingMethod". The value 0x00 specifies that neither compressionMethod nor encryptingMethod is used. Values other than 0x00 are vehicle manufacturer specific.
	Variation	–
	fileSizeUncompressed	
	Type	uint64
	Direction	OUT
Comment	Defines the size of the uncompressed file to be uploaded in bytes.	
Variation	–	
fileSizeCompressed		
Type	uint64	
Direction	OUT	
Comment	Defines the size of the compressed file to be uploaded in bytes.	
Variation	–	
maxNumberOfBlockLength		
Type	uint64	





	Direction	OUT
	Comment	Max number of bytes to be included in each TransferData response excluding the SID and the blockSequenceCounter.
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
	Comment	If the operation Dcm_ProcessRequestReadFile returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
	Variation	–
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	ProcessRequestReplaceFile	
Comment	Callout function. DCM shall call this function to start a RequestFileTransfer process with modeOfOperation equal to 0x03 (ReplaceFile).	
Variation	–	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	DCM_INITIAL: All In-parameters are valid. DCM_PENDING: All In-parameters are set to 0x00. DCM_CANCEL: All In-parameters are set to 0x00. DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x00.
	Variation	–
	filePathAndNameLength	
	Type	uint16
	Direction	IN
	Comment	Defines the length in bytes for the parameter filePathAndName.
	Variation	–
	filePathAndName	
	Type	const Dcm_FileAndDirNameType
	Direction	IN
	Comment	Defines the file system location of the server where the file which shall be added, deleted, replaced or read from depending on the parameter modeOfOperation parameter. In addition this parameter includes the file name of the file which shall be added, deleted, replaced or read as part of the file path.
	Variation	–
	dataFormatIdentifier	
Type	uint8	
Direction	IN	





	Comment	This data-parameter is a one byte value with each nibble encoded separately. The high nibble specifies the "compressionMethod", and the low nibble specifies the "encryptingMethod". The value 0x00 specifies that neither compressionMethod nor encryptingMethod is used. Values other than 0x00 are vehicle manufacturer specific.
	Variation	–
	fileSizeUncompressed	
	Type	uint64
	Direction	IN
	Comment	Defines the size of the uncompressed file to be download in bytes.
	Variation	–
	fileSizeCompressed	
	Type	uint64
	Direction	IN
	Comment	Defines the size of the compressed file to be downloaded in bytes.
	Variation	–
	maxNumberOfBlockLength	
	Type	uint64
	Direction	OUT
	Comment	Max number of bytes to be included in each TransferData request excluding the SID and the blockSequenceCounter.
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
Comment	If the operation Dcm_ProcessRequestReplaceFile returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.	
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	ReadFileOrDir	
Comment	Callout function. DCM shall call this function when data shall be sent as a response to UDS service TransferData if there's an ongoing RequestFileTransfer process started with 0x04 (ReadFile) or 0x05 (Read Dir).	
Variation	–	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	DCM_INITIAL: All In-parameters are valid. DCM_PENDING: All In-parameters are set to 0x00. DCM_CANCEL: All In-parameters are set to 0x00. DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x00.





	Variation	–
	DataLength	
	Type	uint64
	Direction	INOUT
	Comment	As in, the parameter defines the maximum block length to be used, i.e. the value of maxNumberOfBlockLength sent to the client in the response of RequestFileTransfer. As out, the parameter defines the actual length in bytes for the parameter Data. The value shall not exceed, but might be less, the value provided as in parameter.
	Variation	–
	Data	
	Type	Dcm_ResponseDataArrayType
	Direction	IN
	Comment	Pointer to the data to be written.
	Variation	–
	ErrorCode	
	Type	Dcm_NegativeResponseCodeType
	Direction	OUT
Comment	If the operation Dcm_ReadFileOrDir returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.	
Variation	–	
Possible Errors	E_OK E_NOT_OK DCM_E_PENDING DCM_E_FORCE_RCRRP	

Operation	WriteFile	
Comment	Callout function. DCM shall call this function when data is received using UDS service TransferData if there's an ongoing RequestFileTransfer process started with 0x01 (AddFile) or 0x03 (ReplaceFile).	
Variation	–	
Parameters	OpStatus	
	Type	Dcm_OpStatusType
	Direction	IN
	Comment	DCM_INITIAL: All In-parameters are valid. DCM_PENDING: All In-parameters are set to 0x00. DCM_CANCEL: All In-parameters are set to 0x00. DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x00.
	Variation	–
	DataLength	
	Type	uint64
	Direction	IN
Comment	Defines the length in bytes for the parameter Data. The value will not exceed, but might be less, compared to the value of maxNumberOfBlockLength return in Dcm_ProcessRequestFileTransfer.	





	Variation	–	
	Data		
	Type	Dcm_RequestDataArrayType	
	Direction	IN	
	Comment	Pointer to the data to be written.	
	Variation	–	
	ErrorCode		
	Type	Dcm_NegativeResponseCodeType	
	Direction	OUT	
	Comment	If the operation Dcm_WriteFile returns value E_NOT_OK, the DCM module shall send a negative response with NRC code equal to the parameter Error Code parameter value.	
Variation	–		
Possible Errors	–		

]()

8.8.3.11 DCMservices

[SWS_Dcm_00698] [

Name	DCMservices		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	GetActiveProtocol		
Comment	–		
Variation	–		
Parameters	ActiveProtocolType		
	Type	Dcm_ProtocolType	
	Direction	OUT	
	Comment	–	
	Variation	–	
	ConnectionId		
	Type	uint16	
	Direction	OUT	
	Comment	–	
	Variation	–	
TesterSourceAddress			





	Type	uint16
	Direction	OUT
	Comment	–
	Variation	–
Possible Errors	E_OK	

Operation	GetSecurityLevel	
Comment	–	
Variation	–	
Parameters	SecLevel	
	Type	Dcm_SecLevelType
	Direction	OUT
	Comment	–
Possible Errors	E_OK	
	E_NOT_OK	

Operation	GetSesCtrlType	
Comment	–	
Variation	–	
Parameters	SesCtrlType	
	Type	Dcm_SesCtrlType
	Direction	OUT
	Comment	–
Possible Errors	E_OK	
	E_NOT_OK	

Operation	ResetToDefaultSession	
Comment	–	
Variation	–	
Possible Errors	E_OK	
	E_NOT_OK	

Operation	SetActiveDiagnostic	
Comment	Allows to activate and deactivate the call of ComM_DCM_ActiveDiagnostic() function.	
Variation	–	
Parameters	active	
	Type	boolean
	Direction	IN
Comment	If false Dcm shall not call ComM_DCM_ActiveDiagnostic(). If true Dcm will call ComM_DCM_ActiveDiagnostic().	





	Variation	–
Possible Errors	E_OK	

]()

8.8.3.12 DCM_Roe

The RoeEventId shall be a Portdefined argument value.

[SWS_Dcm_00699] [

Name	DCM_Roe		
Comment	–		
IsService	true		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent)}		
Possible Errors	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed

Operation	TriggerOnEvent		
Comment	–		
Variation	–		
Possible Errors	E_OK E_NOT_OK		

]()

8.8.3.13 Authentication

[SWS_Dcm_91072] [

Name	Authentication		
Comment	–		
IsService	true		
Variation	–		
Possible Errors	0	E_OK	Operation successful

Operation	SetDeauthenticatedRole		
Comment	Sets a new role used in deauthenticated state for that connection. The set role is valid until the connection switches into authenticated state or the ECU is reset.		
Variation	–		



△

Parameters	deauthenticatedRole	
	Type	Dcm_AuthenticationRoleType
	Direction	IN
	Comment	New deauthenticated role that is assigned to that connection.
	Variation	–
Possible Errors	E_OK	

]()

8.8.4 NvDataInterface

8.8.4.1 DataServices_{DID}

[SWS_Dcm_91061] [

Name	DataServices_{DID}	
Comment	–	
IsService	false	
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_ATOMIC_NV_DATA_INTERFACE) DID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)}	
Data Elements	data	
	Type	{DID}_Struct_DataType
	Variation	DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})

] ([RS_Diag_04218](#))

8.8.5 Ports

This section formally specifies the corresponding AUTOSAR Service using the concepts of the Software-Component-Template. The following definition can be generated completely out of the configuration of the Dcm, which defines the exact ports that are present and their names.

Naming of the port : The prefix of the port name is fixed and defined hereafter (e.g. DataServices_). The name behind the prefix corresponds to the name of the associated container in the ECU configuration and can be freely defined during the configuration step. e.g. : for a [DcmDspData](#) container called Speed the port name would be DataServices_Speed

```

1 ServiceSwComponentType Dcm {
2
3   //the presence and name of this port is configuration-independent
4   ProvidePort DCMservices DCMservices;
5
6   //see configuration parameter DcmDspSecurityRow

```

```

7   RequirePort SecurityAccess_{SecurityLevel} SecurityAccess_{
      SecurityLevel};
8   ...
9
10
11  //see configuration parameter DcmDspData
12  RequirePort DataServices_{Data} DataServices_{Data};
13  ProvidePort DataServices_{Data} DataServices_{Data}; // Only if
      the data can be written and DcmDspDataUsePort is set to
      USE_DATA_SENDER_RECEIVER or to USE_DATA_SENDER_RECEIVER_AS_SERVICE
14  ...
15
16
17  //see configuration parameter DcmDspVehInfoData
18  RequirePort InfotypeServices_{VehInfoData}
19  InfotypeServices_{VehInfoData}
20  ...
21
22
23  //see configuration parameter DcmDspRoutine
24  RequirePort RoutineServices_{RoutineName}
25  RoutineServices_{RoutineName};
26  ...
27
28  //see configuration parameter DcmDspRequestControl
29  RequirePort RequestControlServices_{Tid}
30  RequestControlServices_{Tid};
31  ...
32
33  //see configuration parameter DcmDslCallbackDCMRequestService
34  RequirePort CallbackDCMRequestServices
35  CallbackDCMRequestServices_{SWC};
36  ...
37
38  //see configuration parameter
39  DcmDsdServiceRequestManufacturerNotication
40  RequirePort ServiceRequestNotification
41  ServiceRequestManufacturerNotication_{Name};
42  ...
43  //see configuration parameter DcmDsdServiceRequestSupplierNotication
44  RequirePort ServiceRequestNotification
45  ServiceRequestSupplierNotication_{SWC};
46  ...
47
48  //Note: When service 0x19 subfunctions 0x14 is used (call to //
49  Dem_GetNextFilteredDTCAndFDC), the following is defined:
50  //Non-DEM-internal calculated fault detection counters are typically
51  //requested from SW-Cs through the RTE. To indicate an equivalent call-
52  tree //for these runables, a work-around is used: The Dcm main
53  function //specifies a trigger to the DEM interface GeneralEvtInfo
54  (operation //GetFaultDetectionCounter), which triggers the
55  according ehavior (refer to //RunnableEntity
56  GetFaultDetectionCounter, chapter "Service Interface //
57  DiagnosticInfo & General" in DEM SWS)."
```

```

52 (The client-server interface can be used from the DEM.)
53
54 RunnableEntity MainFunction
55     symbol \ARApiRef{Dcm_MainFunction}"
56     _____canbeInvokedConcurrently_=_FALSE
57     _____SSCP_=_port_CBFaultDetectCtrDummy, _
58     GetFaultDetectionCounter
59 Connector_from_CBFaultDetectCtrDummy_to_Dem/GeneralEvtInfo
60 }
  
```

8.8.5.1 Dcm_CallbackDCMRequestServices_{Name}

[SWS_Dcm_01033] [

Name	CallbackDCMRequestServices_{Name}		
Kind	RequiredPort	Interface	CallbackDCMRequestServices
Description	-		
Variation	Name = {ecuc(Dcm/DcmConfigSet/DcmDsl/DcmDslCallbackDCMRequestService.SHORT-NAME)}		

]()

8.8.5.2 DataServices_DIDRange_{Range}

[SWS_Dcm_01034] [

Name	DataServices_DIDRange_{Range}		
Kind	RequiredPort	Interface	DataServices_DIDRange_{Range}
Description	-		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRange.DcmDspDidRangeUsePort)} == TRUE Range = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDidRange.SHORT-NAME)})		

]()

8.8.5.3 DataServices_{DID}

[SWS_Dcm_91058] [

Name	DataServices_{DID}		
Kind	ProvidedPort	Interface	–
Description	–		
Variation	<pre>(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_NV_DATA_INTERFACE)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidWrite)} == NULL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidRead)} != NULL) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl)} !=NULL)) DID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)}</pre>		

|(RS_Diag_04218)

[SWS_Dcm_91060] [

Name	DataService_{DID}		
Kind	Provided RequiredPort	Interface	–
Description	–		
Variation	<pre>(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_NV_DATA_INTERFACE)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidWrite)} != NULL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidRead)} != NULL)) DID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)}</pre>		

|(RS_Diag_04218)

[SWS_Dcm_91059] [

Name	DataServices_{DID}		
Kind	RequiredPort	Interface	–
Description	–		
Variation	<pre>(({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_NV_DATA_INTERFACE)) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidWrite)} == NULL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidRead)} != NULL) ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef->DcmDspDidInfo/DcmDspDidControl)} !=NULL)) DID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)}</pre>		

|(RS_Diag_04218)

8.8.5.4 DataServices_{Data}

[SWS_Dcm_01035] [

Name	DataServices_{Data}		
Kind	RequiredPort	Interface	-
Description	-		
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort)} == (USE_DATA_SYNCH_CLIENT_SERVER USE_DATA_SENDER_RECEIVER USE_DATA_SENDER_RECEIVER_AS_SERVICE)) Data = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)} {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData.SHORT-NAME)}		

]()

[SWS_Dcm_01310] [

Name	DataServices_{Data}		
Kind	RequiredPort	Interface	-
Description	-		
Variation	((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_DATA_ELEMENT_SPECIFIC_INTERFACES) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == (USE_DATA_SENDER_RECEIVER USE_DATA_SENDER_RECEIVER_AS_SERVICE))) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidWrite)} == NULL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidRead)} != NULL))) Data = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}		

]()

[SWS_Dcm_01031] [

Name	DataServices_{Data}		
Kind	ProvidedPort	Interface	-
Description	-		
Variation	((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_DATA_ELEMENT_SPECIFIC_INTERFACES) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == (USE_DATA_SENDER_RECEIVER USE_DATA_SENDER_RECEIVER_AS_SERVICE))) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidWrite)} != NULL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidRead)} == NULL)) Data = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}		

]()

[SWS_Dcm_01311] [

Name	DataServices_{Data}		
Kind	Provided RequiredPort	Interface	DataServices_{Data}
Description	-		





Variation	<pre>((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == USE_DATA_ELEMENT_SPECIFIC_INTERFACES)) &&({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort)} == (USE_DATA_SENDER_RECEIVER USE_DATA_SENDER_RECEIVER_AS_SERVICE))) &&({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidWrite)} != NULL) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidRead)} != NULL) Data = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspData.SHORT-NAME)}</pre>
------------------	--

]()

8.8.5.5 IOControlRequest_{DID}

[SWS_Dcm_01312] [

Name	IOControlRequest_{DID}		
Kind	Provided RequiredPort	Interface	IOControlRequest_{DID}
Description	-		
Variation	<pre>((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE USE_DATA_SENDER_RECEIVER_AS_SERVICE))) && ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidControl)} != NULL) DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})</pre>		

]()

8.8.5.6 IOControlResponse_{DID}

[SWS_Dcm_01313] [

Name	IOControlResponse_{DID}		
Kind	RequiredPort	Interface	IOControlResponse_{DID}
Description	-		
Variation	<pre>((({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidUsePort)} == (USE_ATOMIC_SENDER_RECEIVER_INTERFACE USE_DATA_SENDER_RECEIVER_AS_SERVICE))) &&({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidInfoRef-> DcmDspDidInfo/DcmDspDidControl)} != NULL) DID = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspDid.SHORT-NAME)})</pre>		

] ([RS_Diag_04218](#))

8.8.5.7 DCM_Roe_{RoeName}

[SWS_Dcm_01032] [

Name	DCM_Roe_{RoeName}		
Kind	ProvidedPort	Interface	DCM_Roe
Description	–		
Port Defined Argument Value(s)	Type	uint8	
	Value	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent/DcmDspRoeEventId.value)}	
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent)} RoeName = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent.SHORT-NAME)}		

]()

8.8.5.8 DCMServices

[SWS_Dcm_01030] [

Name	DCMServices		
Kind	ProvidedPort	Interface	DCMServices
Description	–		
Variation	–		

]()

8.8.5.9 InfotypeServices_{VehInfoData}

[SWS_Dcm_01037] [

Name	InfotypeServices_{VehInfoData}		
Kind	RequiredPort	Interface	InfotypeServices_{VehInfoData}
Description	–		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData/DcmDspVehInfoDataUsePort)}==TRUE VehInfoData = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData.SHORT-NAME)}		

]()

8.8.5.10 RequestControlServices_{Tid}

[SWS_Dcm_01038] [

Name	RequestControlServices_{Tid}		
Kind	RequiredPort	Interface	RequestControlServices_{Tid}
Description	–		
Variation	Tid = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl.SHORT-NAME)}		

]()

8.8.5.11 RequestFileTransfer

[SWS_Dcm_91143] [

Name	RequestFileTransfer		
Kind	RequiredPort	Interface	RequestFileTransfer
Description	–		
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRequestFileTransfer/DcmRequestFileTransferUsePort)}) == TRUE)		

]()

8.8.5.12 ServiceRequestManufacturerNotification_{Name}

[SWS_Dcm_01039] [

Name	ServiceRequestManufacturerNotification_{Name}		
Kind	RequiredPort	Interface	ServiceRequestNotification
Description	–		
Variation	({ecuc(Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceRequestManufacturerNotification)}) != NULL) Name = {ecuc(Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceRequestManufacturerNotification.SHORT-NAME)}		

]()

8.8.5.13 ServiceRequestSupplierNotification_{Name}

[SWS_Dcm_01042] [

Name	ServiceRequestSupplierNotification_{Name}		
Kind	RequiredPort	Interface	ServiceRequestNotification
Description	–		





Variation	{ecuc(Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceRequestSupplierNotification)} != NULL Name = {ecuc(Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceRequestSupplierNotification.SHORT-NAME)}
------------------	--

]()

8.8.5.14 RoutineServices_{RoutineName}

[SWS_Dcm_01040] [

Name	RoutineServices_{RoutineName}		
Kind	RequiredPort	Interface	RoutineServices_{RoutineName}
Description	-		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.DcmDspRoutineUsePort)} == TRUE RoutineName = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine.SHORT-NAME)}		

]()

8.8.5.15 SecurityAccess_{SecurityLevel}

[SWS_Dcm_01041] [

Name	SecurityAccess_{SecurityLevel}		
Kind	RequiredPort	Interface	SecurityAccess_{SecurityLevel}
Description	-		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.DcmDspSecurityUsePort)} == USE_ASYNC_CLIENT_SERVER SecurityLevel = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.SHORT-NAME)}		

]()

8.8.5.16 Dcm_DiagnosticSessionControlModeSwitchInterface

[SWS_Dcm_91033] [

Name	DiagnosticSessionControlModeSwitchInterface		
Kind	ProvidedPort	Interface	Dcm_DiagnosticSessionControlModeSwitchInterface
Description	A ModeSwitchInterface PPortPrototype used to notify SW-Cs about the current Diagnostic Session		





Variation	–
------------------	---

]()

8.8.5.17 Dcm_EcuResetModeSwitchInterface

[SWS_Dcm_91034] [

Name	EcuResetModeSwitchInterface		
Kind	ProvidedPort	Interface	Dcm_EcuResetModeSwitchInterface
Description	A ModeSwitchInterface PPortPrototype used to notify SW-Cs about an upcoming ECU Reset and its type		
Variation	–		

]()

8.8.5.18 Dcm_ModeRapidPowerShutDownModeSwitchInterface

[SWS_Dcm_91035] [

Name	ModeRapidPowerShutDownModeSwitchInterface		
Kind	ProvidedPort	Interface	Dcm_ModeRapidPowerShutDownModeSwitchInterface
Description	A ModeSwitchInterface PPortPrototype used to notify SW-Cs about the rapid power shut down mode		
Variation	–		

]()

8.8.5.19 Dcm_CommunicationControlModeSwitchInterface_{ComMChannel-Name}

[SWS_Dcm_91036] [

Name	CommunicationControlModeSwitchInterface_{ComMChannelName}		
Kind	ProvidedPort	Interface	CommunicationControlModeSwitchInterface_{ComMChannelName}
Description	A ModeSwitchInterface PPortPrototype used to notify SW-Cs about the communication control of the indicated ComM channel		





Variation	ComMChannelName = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlAllChannel/DcmDspAllComMChannelRef->ComMChannel.SHORT-NAME)} or {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlSpecificChannel/DcmDspSpecificComMChannelRef->ComMChannel.SHORT-NAME)}
------------------	--

]()

8.8.5.20 Dcm_ControlDTCSettingModeSwitchInterface

[SWS_Dcm_91037] [

Name	ControlDTCSettingModeSwitchInterface		
Kind	ProvidedPort	Interface	Dcm_ControlDTCSettingModeSwitchInterface
Description	A ModeSwitchInterface PPortPrototype used to notify SW-Cs about the DTC Setting mode		
Variation	-		

]()

8.8.5.21 Dcm_ResponseOnEventModeSwitchInterface_{RoeEventID}

[SWS_Dcm_91038] [

Name	ResponseOnEventModeSwitchInterface_{RoeEventID}		
Kind	ProvidedPort	Interface	ResponseOnEvent_{RoeEventID}
Description	A ModeSwitchInterface PPortPrototype used to notify SW-Cs about the mode of the indicated Response On Event		
Variation	RoeEventID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent.SHORT-NAME)}		

]()

8.8.5.22 Dcm_SecurityAccessModeSwitchInterface

[SWS_Dcm_91039] [

Name	SecurityAccessModeSwitchInterface		
Kind	ProvidedPort	Interface	Dcm_SecurityAccessModeSwitchInterface
Description	A ModeSwitchInterface PPortPrototype used to notify SW-Cs about the current Security Level		
Variation	-		

]()

8.8.5.23 Dcm_UploadDownloadServices

[SWS_Dcm_91084] [

Name	UploadDownloadServices		
Kind	RequiredPort	Interface	UploadDownloadServices
Description	–		
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspMemoryTransfer.DcmDspMemoryTransferUsePort)} == TRUE)		

]()

8.8.5.24 Dcm_Authentication_{Connection}

[SWS_Dcm_91073] [

Name	Authentication_{Connection}		
Kind	RequiredPort	Interface	Authentication
Description	–		
Port Defined Argument Value(s)	Type	uint16	
	Value	{ecuc(Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection.DcmDslProtocolRxConnectionId)}	
Variation	Connection = { Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection.Short-Name}}		

]()

8.8.6 ModeDeclarationGroups

8.8.6.1 DcmDiagnosticSessionControl

[SWS_Dcm_91019] [

Name	DcmDiagnosticSessionControl	
Kind	ModeDeclarationGroup	
Category	EXPLICIT_ORDER	
Initial mode	DCM_DEFAULT_SESSION	
On transition value	255	
Modes	DCM_DEFAULT_SESSION	0
	DCM_PROGRAMMING_SESSION	1
	DCM_EXTENDED_DIAGNOSTIC_SESSION	2





	DCM_SAFETY_SYSTEM_DIAGNOSTIC_SESSION	3
Description	ModeDeclarationGroup representing the different diagnostic sessions Further modes to be added: {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow.SHORT-NAME)}	

]()

Note: According [SWS_Dcm_CONSTR_6001] there are standardized mode declaration which are part of the standardized AUTOSAR interface. Note: Refer [ecuc_sws_2108] defining the symbolic name prefix

8.8.6.2 DcmEcuReset

[SWS_Dcm_91021] [

Name	DcmEcuReset	
Kind	ModeDeclarationGroup	
Category	EXPLICIT_ORDER	
Initial mode	DCM_NONE	
On transition value	255	
Modes	DCM_NONE	0
	DCM_HARD	1
	DCM_KEYONOFF	2
	DCM_SOFT	3
	DCM_JUMPTOBOOTLOADER	4
	DCM_JUMPTOSYSSUPPLIERBOOTLOADER	5
	DCM_EXECUTE	6
Description	ModeDeclarationGroup representing the different ECU reset types	

]()

8.8.6.3 DcmModeRapidPowerShutDown

[SWS_Dcm_91023] [

Name	DcmModeRapidPowerShutDown	
Kind	ModeDeclarationGroup	
Category	EXPLICIT_ORDER	
Initial mode	DCM_ENABLE_RAPIDPOWERSHUTDOWN	
On transition value	255	





Modes	DCM_ENABLE_RAPIDPOWERSHUTDOWN	0
	DCM_DISABLE_RAPIDPOWERSHUTDOWN	1
Description	ModeDeclarationGroup representing the enable/disable state of rapid power shutdown	

]()

8.8.6.4 DcmCommunicationControl

[SWS_Dcm_91025] [

Name	DcmCommunicationControl	
Kind	ModeDeclarationGroup	
Category	EXPLICIT_ORDER	
Initial mode	DCM_ENABLE_RX_TX_NORM_NM	
On transition value	255	
Modes	DCM_ENABLE_RX_TX_NORM	0
	DCM_ENABLE_RX_DISABLE_TX_NORM	1
	DCM_DISABLE_RX_ENABLE_TX_NORM	2
	DCM_DISABLE_RX_TX_NORMAL	3
	DCM_ENABLE_RX_TX_NM	4
	DCM_ENABLE_RX_DISABLE_TX_NM	5
	DCM_DISABLE_RX_ENABLE_TX_NM	6
	DCM_DISABLE_RX_TX_NM	7
	DCM_ENABLE_RX_TX_NORM_NM	8
	DCM_ENABLE_RX_DISABLE_TX_NORM_NM	9
	DCM_DISABLE_RX_ENABLE_TX_NORM_NM	10
DCM_DISABLE_RX_TX_NORM_NM	11	
Description	ModeDeclarationGroup representing the different communication control states	

]()

8.8.6.5 DcmControlDTCSetting

[SWS_Dcm_91027] [

Name	DcmControlDTCSetting	
Kind	ModeDeclarationGroup	
Category	EXPLICIT_ORDER	
Initial mode	DCM_ENABLEDTCSETTING	



△

On transition value	255	
Modes	DCM_ENABLEDTCSETTING	0
	DCM_DISABLEDTCSETTING	1
Description	ModeDeclarationGroup representing the enable/disable state for DTC storage	

]()

8.8.6.6 DcmResponseOnEvent

[SWS_Dcm_91029] [

Name	DcmResponseOnEvent	
Kind	ModeDeclarationGroup	
Category	EXPLICIT_ORDER	
Initial mode	DCM_EVENT_CLEARED	
On transition value	255	
Modes	DCM_EVENT_STARTED	0
	DCM_EVENT_STOPPED	1
	DCM_EVENT_CLEARED	2
Description	ModeDeclarationGroup representing the state of a Response On Event	

]()

8.8.6.7 DcmSecurityAccess

[SWS_Dcm_91031] [

Name	DcmSecurityAccess	
Kind	ModeDeclarationGroup	
Category	EXPLICIT_ORDER	
Initial mode	DCM_SEC_LEV_LOCKED	
On transition value	255	
Modes	DCM_SEC_LEV_LOCKED	0
Description	ModeDeclarationGroup representing the different diagnostic security levels Further modes to be added: {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDsp Security/DcmDspSecurityRow.SHORT-NAME)}	

]()

8.8.6.8 DcmAuthenticationState

[SWS_Dcm_91067] [

Name	DcmAuthenticationState_{Connection}	
Kind	ModeDeclarationGroup	
Category	EXPLICIT_ORDER	
Initial mode	DCM_DEAUTHENTICATED	
On transition value	255	
Modes	DCM_DEAUTHENTICATED	0
	DCM_AUTHENTICATED	1
Description	Representing the authentication state of a diagnostic connection.	

]()

8.8.7 Mode-Switch-Interfaces

8.8.7.1 Dcm_DiagnosticSessionControlModeSwitchInterface

[SWS_Dcm_91020] [

Name	Dcm_DiagnosticSessionControlModeSwitchInterface	
Comment	A SW-C that wants to get informed about the current Diagnostic Session requires the Mode SwitchInterface Dcm_DiagnosticSessionControlModeSwitchInterface	
IsService	true	
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow.SHORT-NAME)}	
ModeGroup	diagnosticSession	DcmDiagnosticSessionControl

]()

8.8.7.2 Dcm_EcuResetModeSwitchInterface

[SWS_Dcm_91022] [

Name	Dcm_EcuResetModeSwitchInterface	
Comment	A SW-C that wants to get informed about an upcoming ECU Reset requires the ModeSwitch Interface Dcm_EcuResetModeSwitchInterface	
IsService	true	
Variation	-	
ModeGroup	ecuReset	DcmEcuReset

]()

8.8.7.3 Dcm_ModeRapidPowerShutDownModeSwitchInterface

[SWS_Dcm_91024] [

Name	Dcm_ModeRapidPowerShutDownModeSwitchInterface	
Comment	A SW-C that wants to get informed about the rapid power shut down mode requires the ModeSwitchInterface Dcm_ModeRapidPowerShutDownModeSwitchInterface	
IsService	true	
Variation	-	
ModeGroup	modeRapidPowerShutDown	DcmModeRapidPowerShutDown

]()

8.8.7.4 Dcm_CommunicationControlModeSwitchInterface

[SWS_Dcm_91026] [

Name	CommunicationControlModeSwitchInterface_{ComMChannelName}	
Comment	A SW-C that wants to get informed about the communication control of a ComM channel requires the ModeSwitchInterface Dcm_CommunicationControlModeSwitchInterface	
IsService	true	
Variation	ComMChannelName = ({ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlAllChannel/DcmDspAllComMChannelRef->ComMChannel.SHORT-NAME)} {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlSpecificChannel/DcmDspSpecificComMChannelRef->	
ModeGroup	communicationControl	DcmCommunicationControl

]()

8.8.7.5 Dcm_ControlDTCSettingModeSwitchInterface

[SWS_Dcm_91028] [

Name	Dcm_ControlDTCSettingModeSwitchInterface	
Comment	A SW-C that wants to get informed about the DTC Setting mode requires the ModeSwitchInterface Dcm_ControlDTCSettingModeSwitchInterface	
IsService	true	
Variation	-	
ModeGroup	controlDTCSetting	DcmControlDTCSetting

]()

8.8.7.6 Dcm_ResponseOnEventModeSwitchInterface

[SWS_Dcm_91030] [

Name	ResponseOnEvent_{RoeEventID}	
Comment	A SW-C that wants to get informed about a Response On Event mode requires the ModeSwitch Interface Dcm_ResponseOnEventModeSwitchInterface	
IsService	true	
Variation	RoeEventID = {ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoe Event.SHORT-NAME)}	
ModeGroup	responseOnEvent	DcmResponseOnEvent

}]()

8.8.7.7 Dcm_SecurityAccessModeSwitchInterface

[SWS_Dcm_91032] [

Name	Dcm_SecurityAccessModeSwitchInterface	
Comment	A SW-C that wants to get informed about the current Security Level requires the ModeSwitch Interface Dcm_SecurityAccessModeSwitchInterface	
IsService	true	
Variation	{ecuc(Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow.SHORT-NAME)}	
ModeGroup	securityAccess	DcmSecurityAccess

}]()

8.8.7.8 Dcm_AuthenticationStateModeSwitchInterface

[SWS_Dcm_91074] [

Name	Dcm_AuthenticationStateModeSwitchInterface_{Connection}	
Comment	-	
IsService	true	
Variation	Connection = Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDsl Connection/DcmDslMainConnection.Short-Name	
ModeGroup	authenticationState	DcmAuthenticationState

}]()

8.9 External diagnostic service processing

The following chapter applies only to external processed diagnostic services.

8.9.1 <Module>_<DiagnosticService>

[SWS_Dcm_00763] [

Service Name	<Module>_<DiagnosticService>	
Syntax	Std_ReturnType <Module>_<DiagnosticService> (Dcm_ExtendedOpStatusType OpStatus, Dcm_MsgContextType* pMsgContext, Dcm_NegativeResponseCodeType* ErrorCode)	
Service ID [hex]	0x32	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	OpStatus	DCM_INITIAL DCM_PENDING DCM_CANCEL DCM_FORCE_RCRRP_OK DCM_POS_RESPONSE_SENT DCM_POS_RESPONSE_FAILED DCM_NEG_RESPONSE_SENT DCM_NEG_RESPONSE_FAILED
Parameters (inout)	pMsgContext	Message-related information for one diagnostic protocol identifier. The pointers in pMsgContext shall point behind the SID.
Parameters (out)	ErrorCode	If the operation <Module>_<DiagnosticService> returns value E_NOT_OK, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful E_NOT_OK: Request was not successful DCM_E_PENDING: Request is not yet finished DCM_E_FORCE_RCRRP: Application requests the transmission of a response Response Pending (NRC 0x78)
Description	Callout function. The Dcm shall call this callout function as soon as valid message is received on relevant DcmRxPduId on SID level. The usecase of multiple diagnostic protocols will be possible by using different arguments and the function shall be programmed in a way that it is reentrant. Caller is responsible for the lifetime of the argument pMsgContext. The name of the callout is defined within parameter DcmDsdSidTabFnc	
Available via	Dcm_Externals.h	

]()

8.9.2 <Module>_<DiagnosticService>_<SubService>

[SWS_Dcm_00764] [

Service Name	<Module>_<DiagnosticService>_<SubService>	
Syntax	Std_ReturnType <Module>_<DiagnosticService>_<SubService> (Dcm_ExtendedOpStatusType OpStatus, Dcm_MsgContextType* pMsgContext, Dcm_NegativeResponseCodeType* ErrorCode)	
Service ID [hex]	0x33	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	





Parameters (in)	OpStatus	DCM_INITIAL DCM_PENDING DCM_CANCEL DCM_FORCE_RCRRP_OK DCM_POS_RESPONSE_SENT DCM_POS_RESPONSE_FAILED DCM_NEG_RESPONSE_SENT DCM_NEG_RESPONSE_FAILED
Parameters (inout)	pMsgContext	Message-related information for one diagnostic protocol identifier. The pointers in pMsgContext shall point behind the SID.
Parameters (out)	ErrorCode	If the operation <Module>_<DiagnosticService>_<SubService> returns value E_NOT_OK, the Dcm module shall send a negative response with NRC code equal to the parameter ErrorCode parameter value.
Return value	Std_ReturnType	E_OK: Request was successful E_NOT_OK: Request was not successful DCM_E_PENDING: Request is not yet finished DCM_E_FORCE_RCRRP: Application requests the transmission of a response Response Pending (NRC 0x78)
Description	Callout function. If a DcmDsdSubServiceFnc is configured for the received subservice, the Dcm shall call this callout function as soon as this subservice is requested. The usecase of multiple diagnostic protocols will be possible by using different arguments and the function shall be programmed in a way that it is reentrant. Caller is responsible for the lifetime of the argument pMsgContext. The name of the callout is defined within parameter DcmDsdSubServiceFnc.	
Available via	Dcm_Externals.h	

⌋()

8.10 Internal interfaces (not normative)

The following interfaces are used in the [Dcm](#) SWS in order to improve the understanding of the [Dcm](#) module behavior. An implementation is not required to use these interfaces.

8.10.1 DslInternal_SetSecurityLevel

```
1 void
2 DslInternal_SetSecurityLevel(Dcm_SecLevelType SecurityLevel)
```

This function sets a new security level value in the [Dcm](#) module. NOTE: for the definition of the parameter, refer to [Dcm_GetSecurityLevel](#).

8.10.2 DslInternal_SetSesCtrlType

```
1 void
2 DslInternal_SetSesCtrlType(Dcm_SesCtrlType SesCtrlType)
```

This function sets a new session control type value in the [Dcm](#) module. NOTE: for the definition of the parameter, refer to the [Dcm_GetSesCtrlType](#).

8.10.3 DspInternal_DcmConfirmation

```
1 void
2 DspInternal_DcmConfirmation(Dcm_IdContextType idContext,
```

```
3 uint16 ConnectionId
4 Dcm_ConfirmationStatusType status)
```

This function confirms the successful transmission or a transmission error of a diagnostic service. This is the right time to perform any application state transitions.

This [API](#) is also called if the response to a diagnostic service is suppressed.

8.10.4 DslInternal_ResponseOnOneEvent

```
1 Dcm_StatusType
2 DslInternal_ResponseOnOneEvent(const Dcm_MsgType MsgPtr,
3 Dcm_MsgLenType MsgLen,
4 uint16 ConnectionId)
```

This [API](#) executes the processing of one event, requested internally in the DCM.

8.10.5 DslInternal_ResponseOnOneDataByPeriodicId

```
1 Dcm_StatusType
2 DslInternal_ResponseOnOneDataByPeriodicId(uint8 PeriodicId)
```

This [API](#) provides the processing of one periodic [ID](#) event, requested internally in the DCM. The frequency of calling this function depends on the rate given in the original [ReadDataByPeriodicID](#) request (parameter [transmissionMode](#)).

8.10.6 DsdInternal_StartPagedProcessing

```
1 void
2 DsdInternal_StartPagedProcessing(const Dcm_MsgContextType* pMsgContext)
```

With this API, the [DSP](#) submodule gives the complete response length to the [Dcm](#) module and starts paged-buffer handling. This [API](#) starts no transmission!

8.10.7 DspInternal_CancelPagedBufferProcessing

```
1 void
2 DspInternal_CancelPagedBufferProcessing()
```

[Dcm](#) informs [DSP](#), that processing of paged-buffer was cancelled due to errors. Upon this call, [DSP](#) is not allowed to process further on paged-buffer handling.

8.10.8 DsdInternal_ProcessPage

```
1 void
2 DsdInternal_ProcessPage(Dcm_MsgLenType FilledPageLen)
```

DSP requests transmission of filled page.

9 Sequence diagrams

9.1 Overview

For clarification, the following sequence diagrams don't represent the full communication mechanism between the *Dcm* module and the PduR module. This is to keep the sequence diagrams simple. Before the *Dcm_TpRxIndication* call, the PduR module will ask the *Dcm* module for a buffer by calling *Dcm_StartOfReception* and *Dcm_CopyRxData*. This exchange is not shown on the next sequence diagrams. After a *PduR_DcmTransmit()* request from the *Dcm* module to the PduR module, data exchanges with *Dcm_CopyTxData* service, are not shown in the sequence diagrams. The function *Xxx_StartProtocol()* shall be called with the very first diagnostic request.

9.2 DSL (Diagnostic Session Layer)

9.2.1 Start Protocol

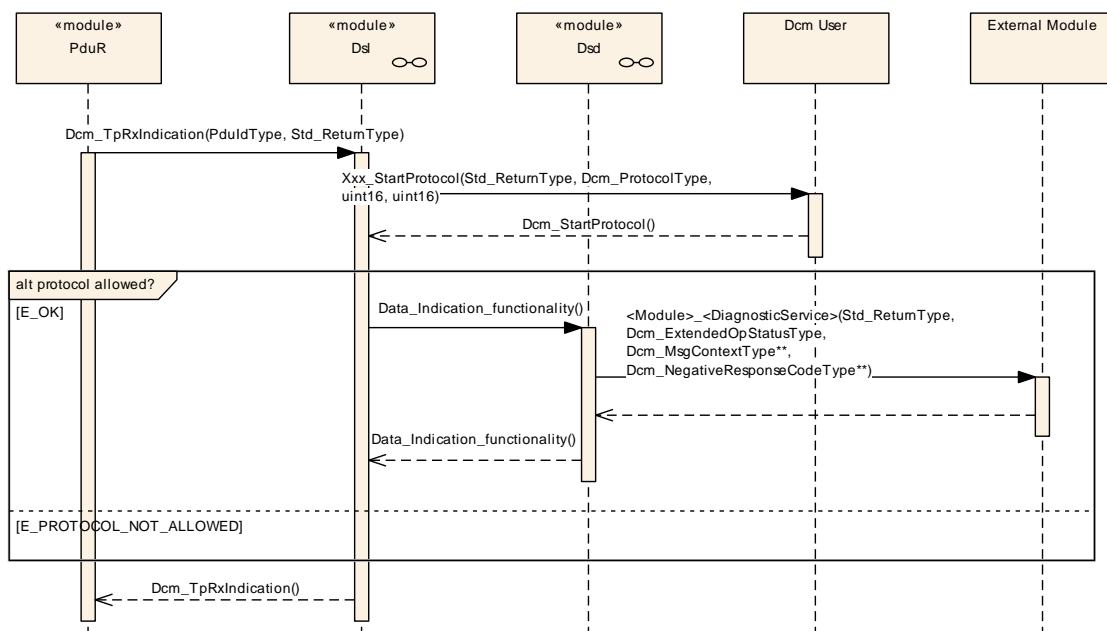


Figure 9.1

9.2.2 Process Busy behavior

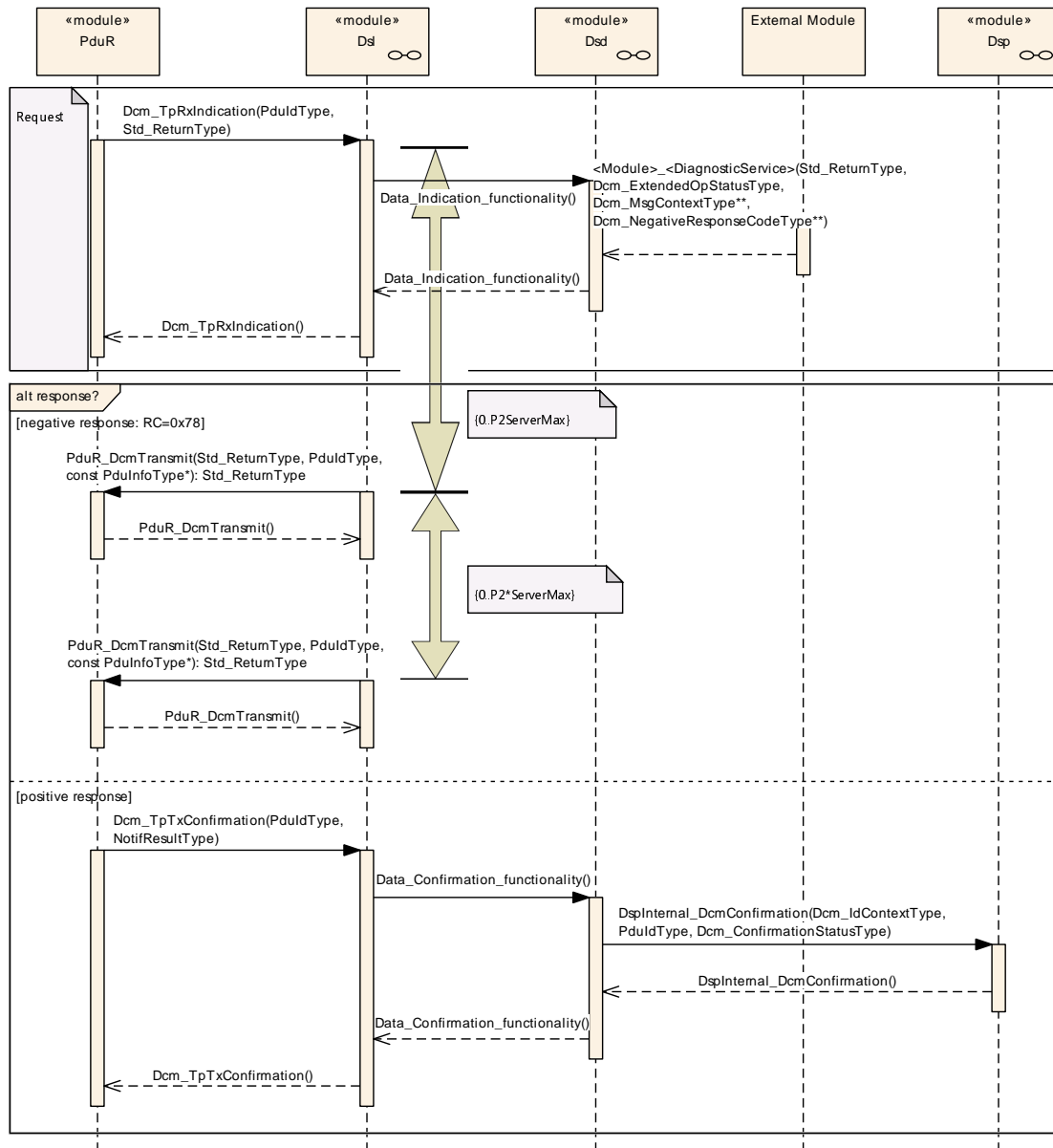


Figure 9.2

Internally, the `DSL` submodule calculates the time to response the tester. In the case that the external module processing the request doesn't close the request by returning `E_OK` or `E_NOT_OK` to `<Module>_<DiagnosticService>()` or `<Module>_<DiagnosticService>_<SubService>()` APIs call (in case of normal response handling) or `DsdInternal_ProcessPage()` (in case of paged-buffer handling) during the `P2ServerMax` and/or `P2*ServerMax`, the `DSL` submodule sends a negative response (requestCorrectlyReceived-ResponsePending) independently.

9.2.3 Update Diagnostic Session Control when timeout occurs

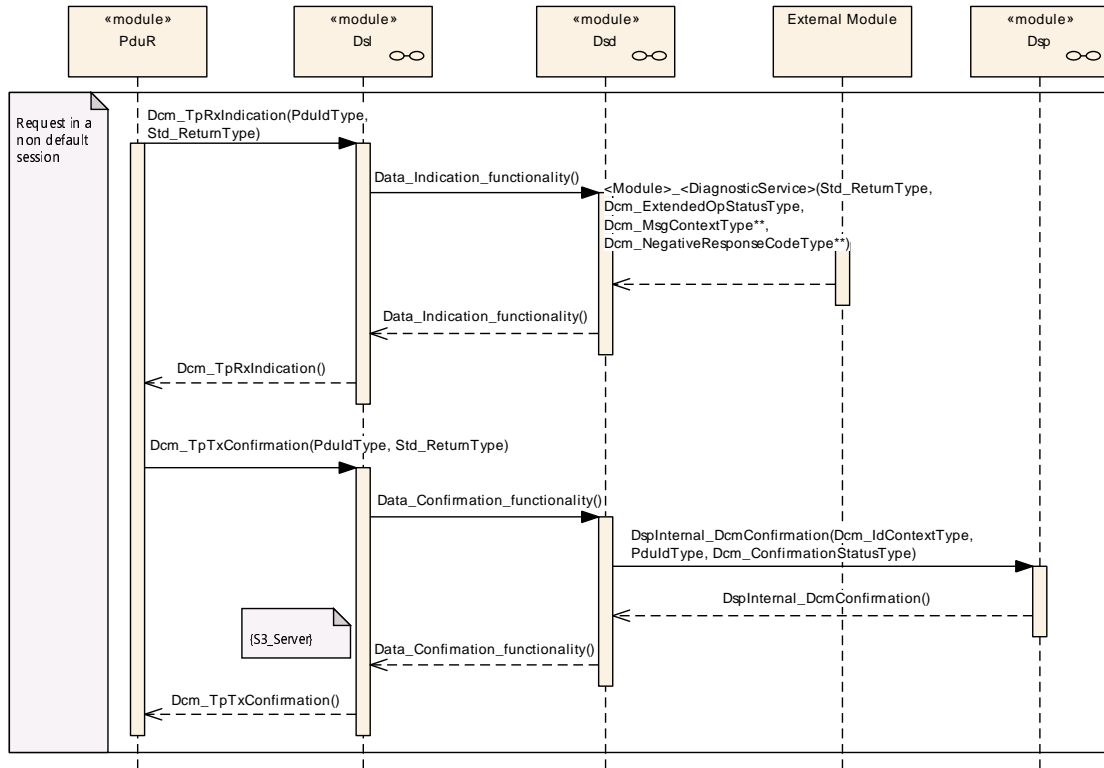


Figure 9.3

The [DSL](#) submodule resets session control value to default, if in a non-default session S3server timeout occurs. S3server timeout timer will be started with every data confirmation from the PduR module.

9.2.4 Process single response of ReadDataByPeriodicId

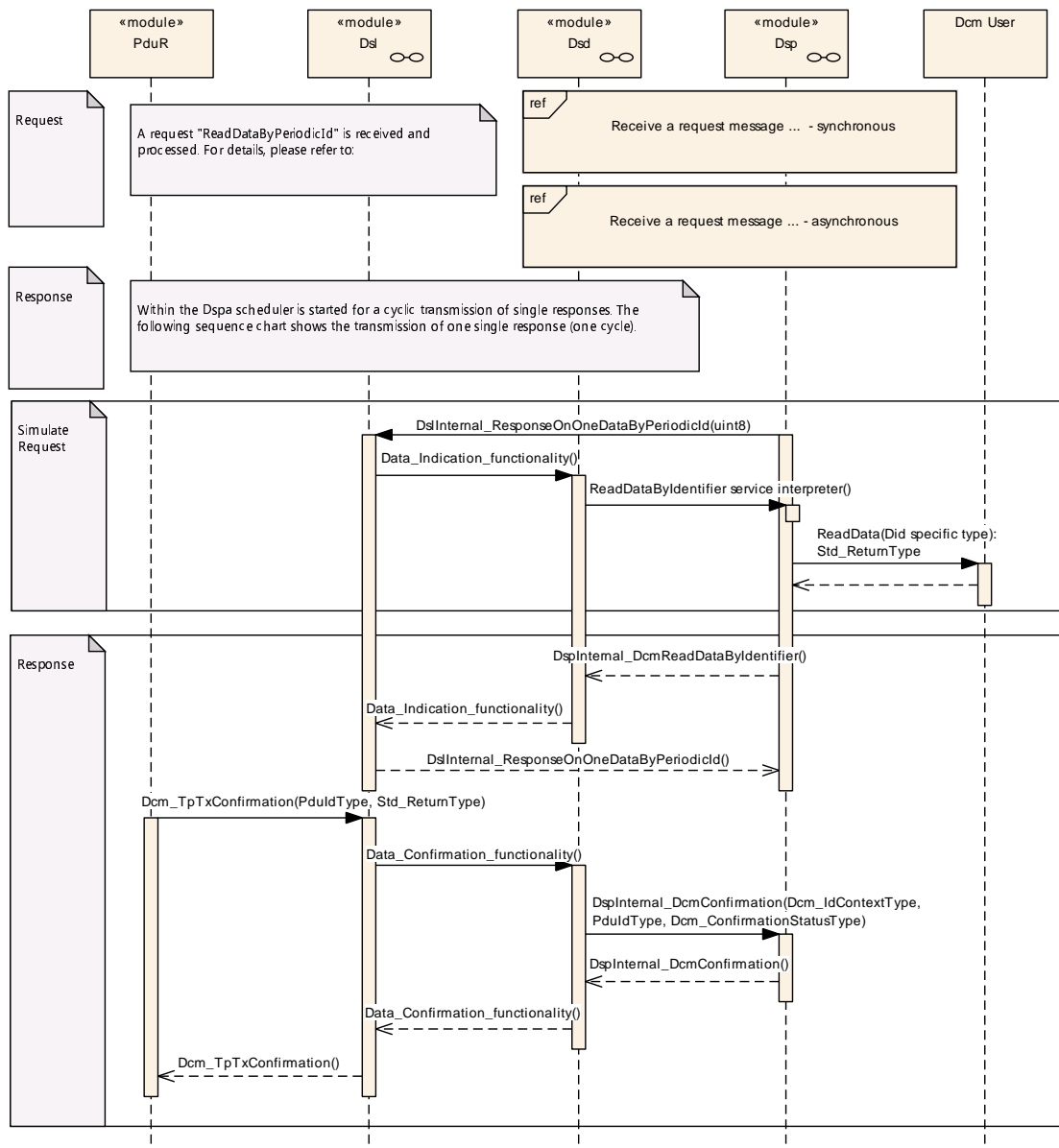


Figure 9.4

The **DSP** submodule requests sampling and transmission of Periodic Identifier data, when an event to Periodic Identifier occurs (i. e. a given time period is over). The **DSP** submodule initiates the sending of one periodic identifier calling the function `ResponseOnOneDataByPeriodicId()` provided by the **DSL** submodule.

Within this function the **DSL** submodule simulates a "ReadDataByIdentifier" request for the given PeriodicId. The High byte of the DataIdentifier shall be set to 0xF2 as specified in [17]) and the low byte is set to value of the PeriodicId.

The ReadData interfaces of the corresponding Datas of the **DID** are called to get the **DID** value. The **Dcm** module is not able to receive for the same periodic identifier

another event request from the DSP submodule, unless the confirmation of the current transmission is received.

9.2.5 Process single event-triggered response of ResponseOnEvent

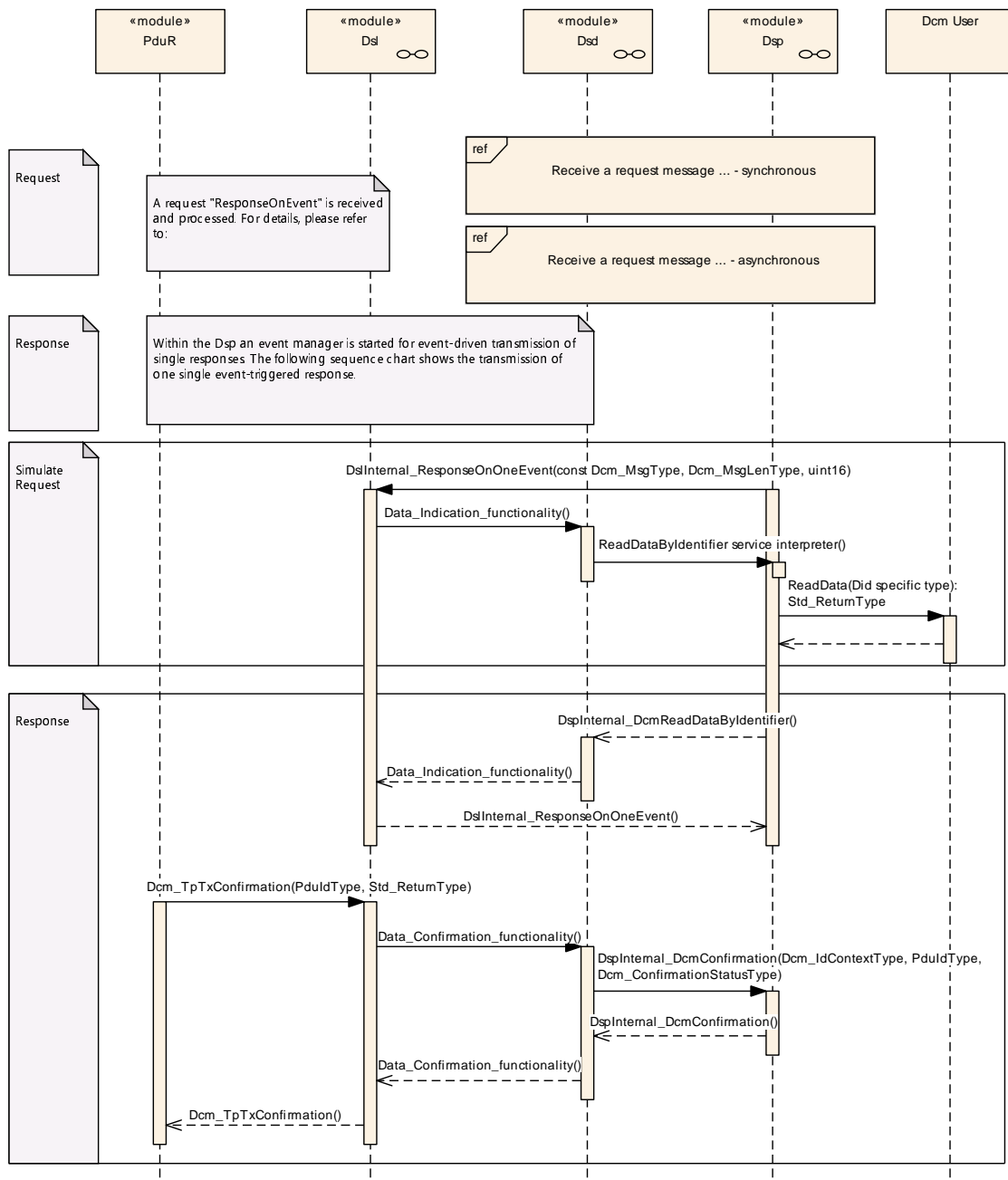


Figure 9.5

This sequence diagram shows an example for ResponseOnEvent. ResponseOnEvent is setup and started for onDTCStatusChange. Event changes are reported to the Dcm which will trigger a serviceToRespondTo.

9.2.6 Process concurrent requests

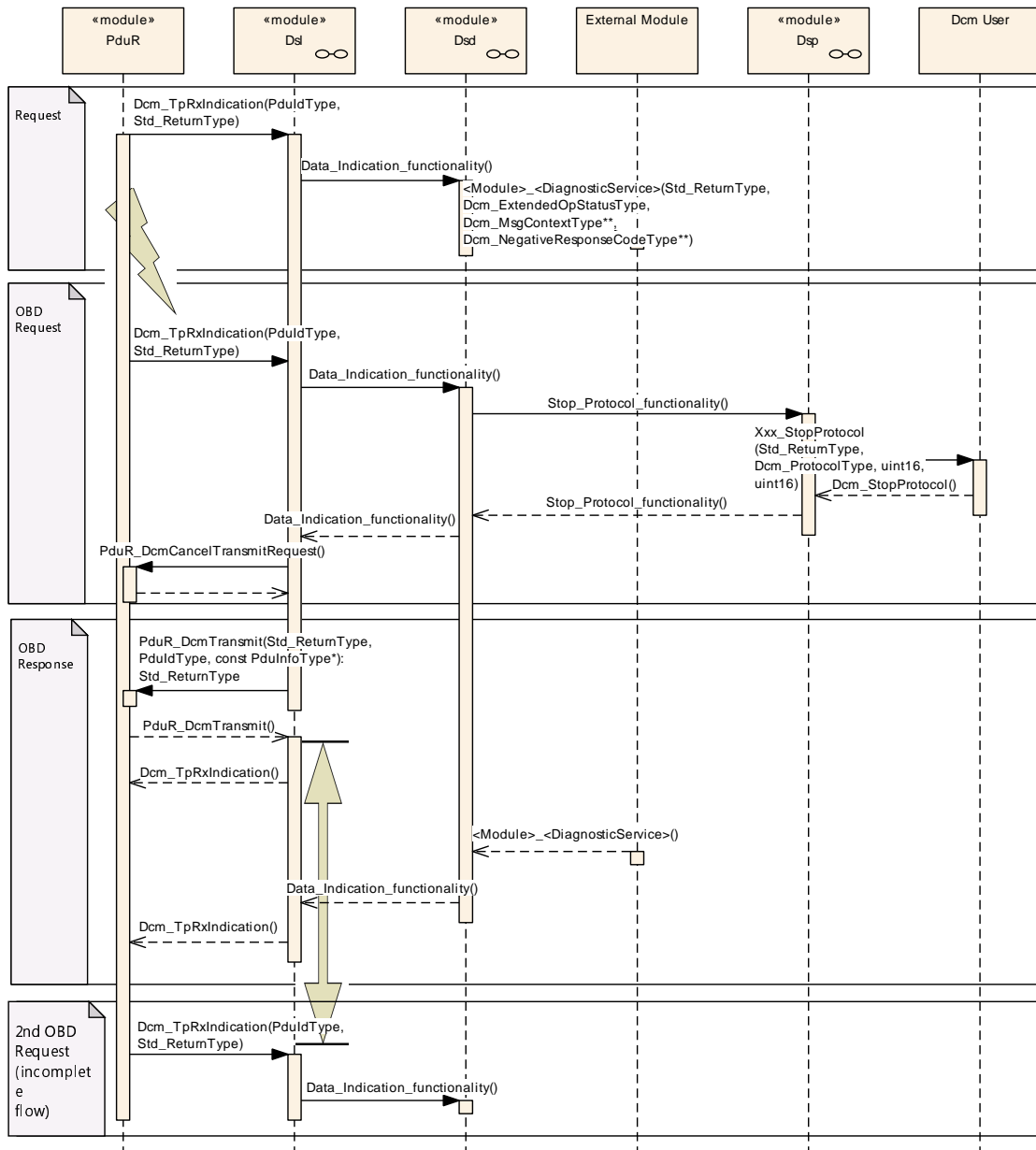


Figure 9.6

On reception of **OBD** request in parallel to processing of a normal diagnostic request (e.g enhanced diagnostic protocol, customer diagnostic protocol), running diagnostic request will be preempted. This is due to the configured higher priority of **OBD** protocol (see configuration parameter `DcmDslProtocolPriority`).

The following is processed on reception of 1st **OBD** request:

- The Application is informed of the protocol stop (done with `Xxx_StopProtocol()`) and resets to a stable state (e.g. switch of digital Ios,..).

- Lower Layer is requested to cancel ongoing transmission on the same N-PDU (done with `PduR_DcmCancelTransmitRequest()`).
- If the `Dcm` is not able to switch fast enough from non `OBD` to `OBD` protocol, the `DSL` submodule responses with a negative response "BusyRepeatRequest" (NRC 0x21) to `OBD` tester. It is in the responsibility of the system designer to ensure that the legislative timings are satisfied.

As long as the external module processing the request is not finished (finish is indicated by returning `E_OK` or `E_NOT_OK` to `<Module>_<DiagnosticService>()/<Module>_<DiagnosticService>_<SubService>()` [API](#) call) or no timeout occurs, the `DSL` submodule responses with negative response "BusyRepeatRequest".

With receiving `E_OK` or `E_NOT_OK` from the external module to `<Module>_<DiagnosticService>()/<Module>_<DiagnosticService>_<SubService>()` [API](#) call, the `DSL` submodule will not transmit a response to old request. There will also not given any negative response to inform first tester about preemption of diagnostic request.

If the external module processing the request never returns `E_OK` or `E_NOT_OK` to `<Module>_<DiagnosticService>()/<Module>_<DiagnosticService>_<SubService>()` [API](#) call, the `DSL` submodule runs into timeout and switches directly to further processing of preempting protocol.

9.2.7 Interface to ComManager

9.2.7.1 Handling in Default Session

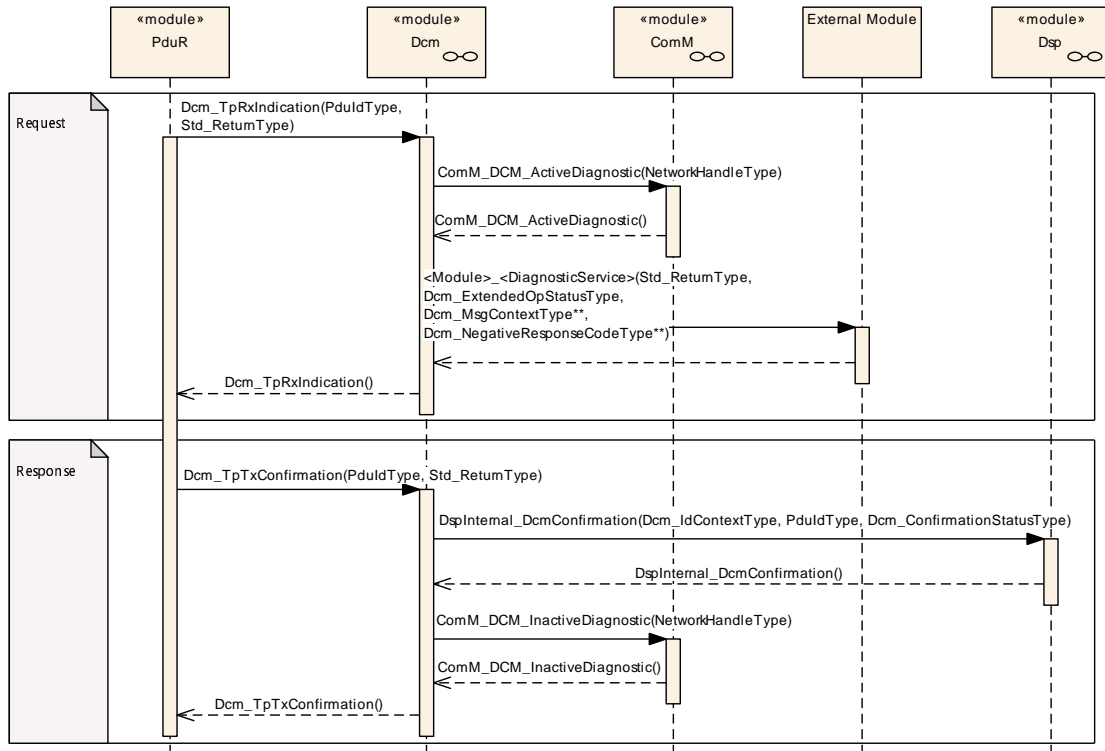


Figure 9.7

9.2.7.2 Handling in Non-Default Session

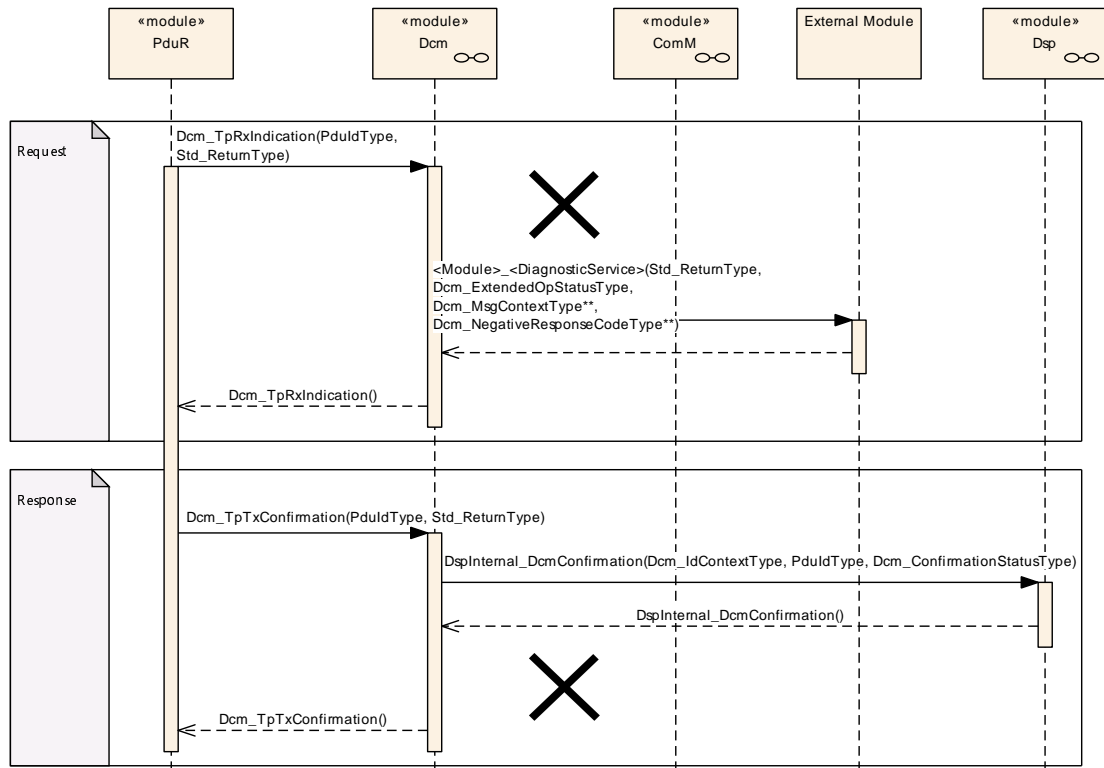


Figure 9.8

9.2.7.3 Session transitions

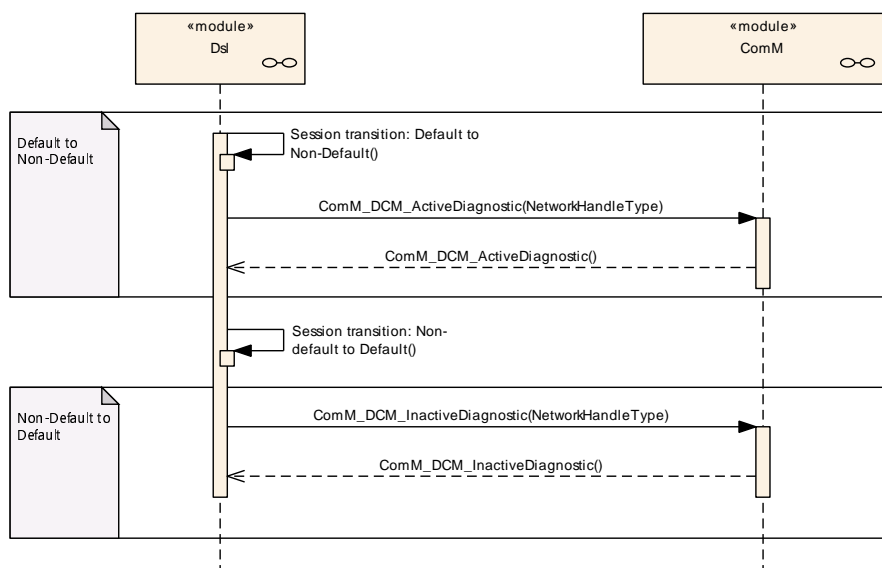


Figure 9.9

9.2.7.4 Communication States

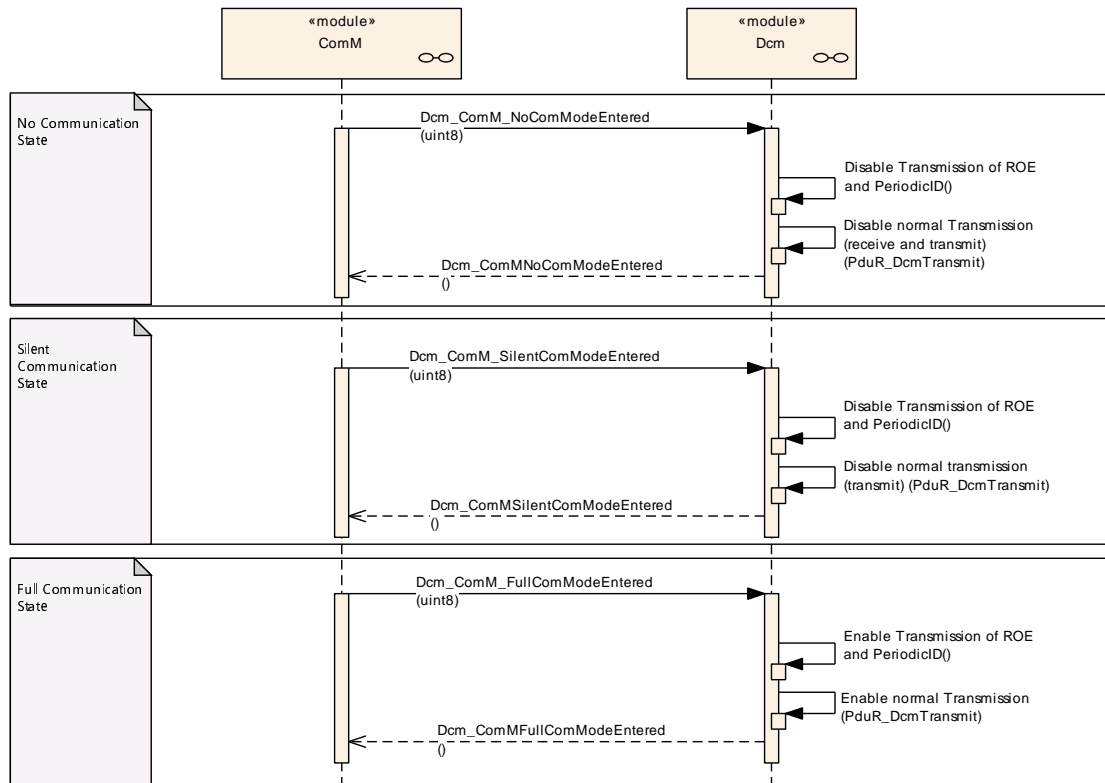


Figure 9.10

DSD (Diagnostic Service Dispatcher) Receive a request message and transmit a positive response message - synchronous transmission

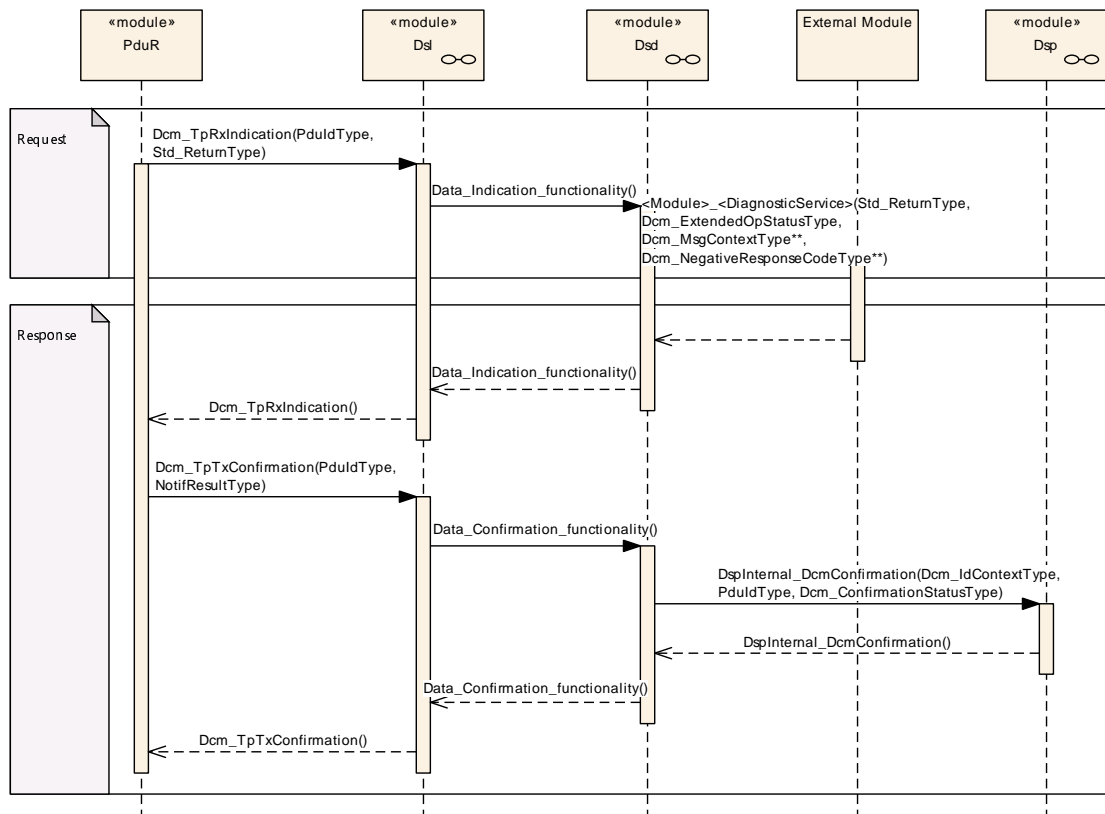


Figure 9.11

Receive a request message and transmit a positive response message - asynchronous transmission

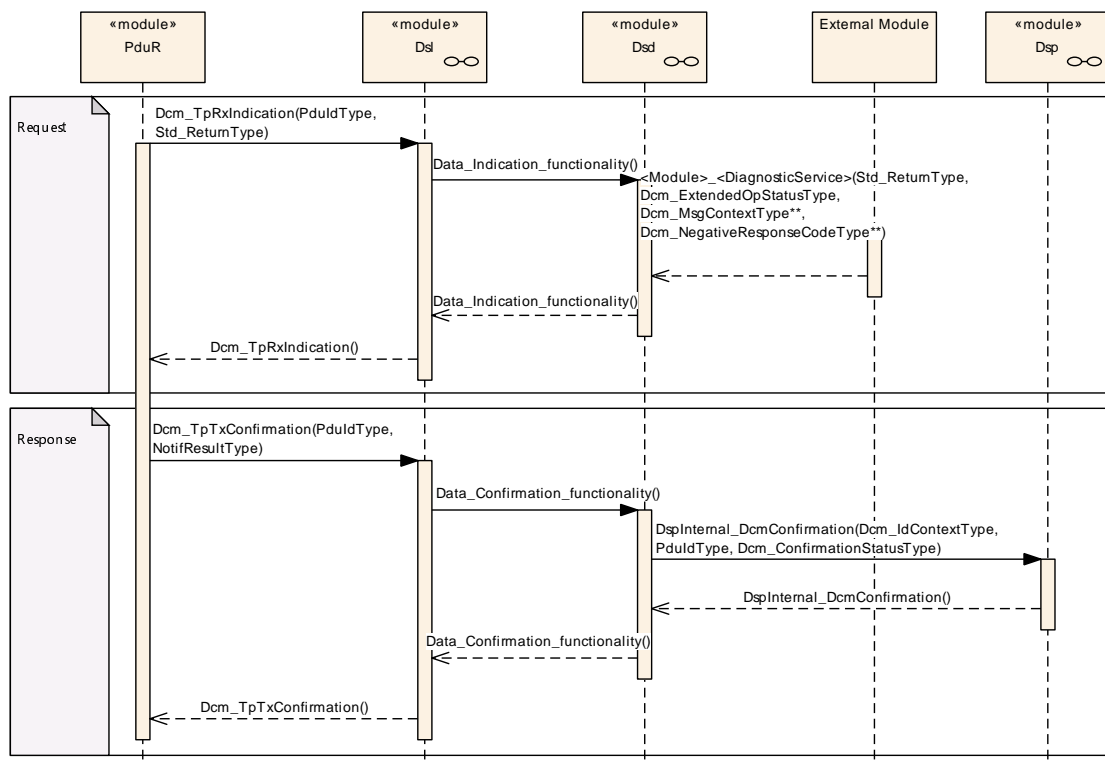


Figure 9.12

Receive a request message and suppress a positive response

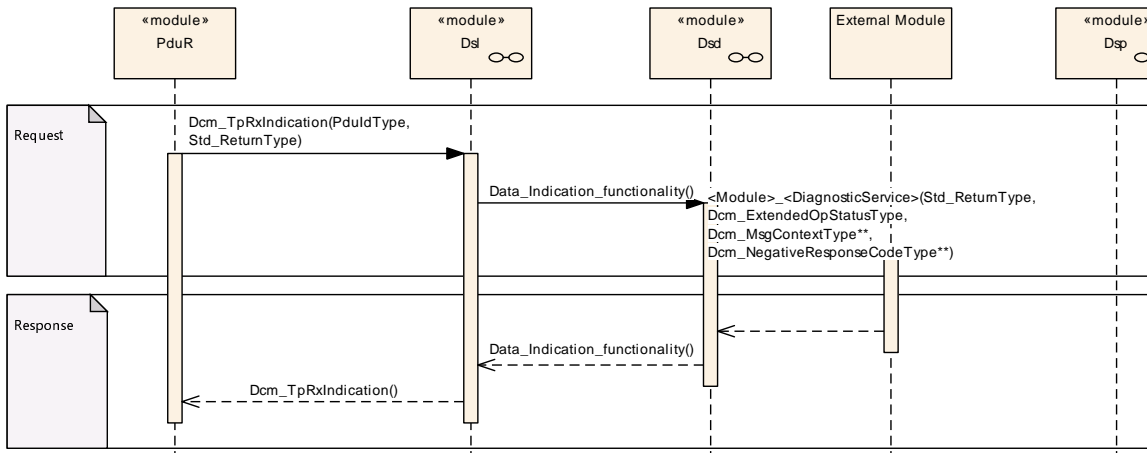


Figure 9.13

9.2.8 Receive request message and transmit negative response message

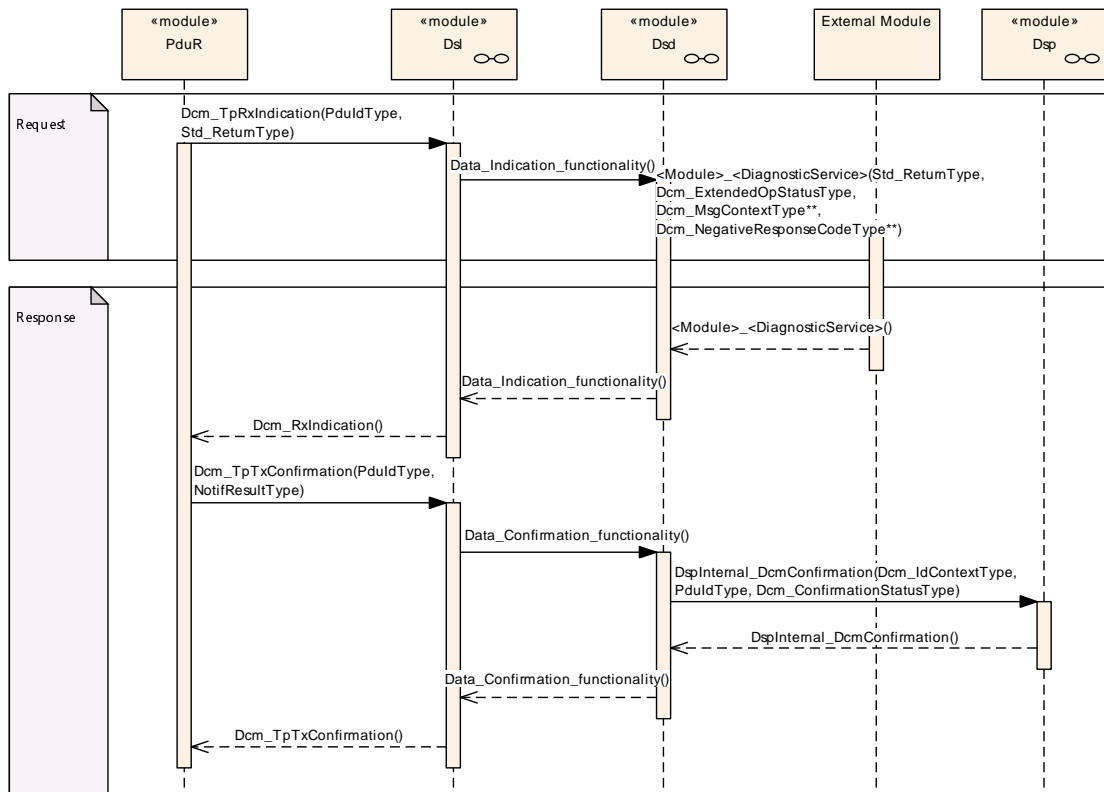


Figure 9.14

9.2.9 Process Service Request with paged-buffer

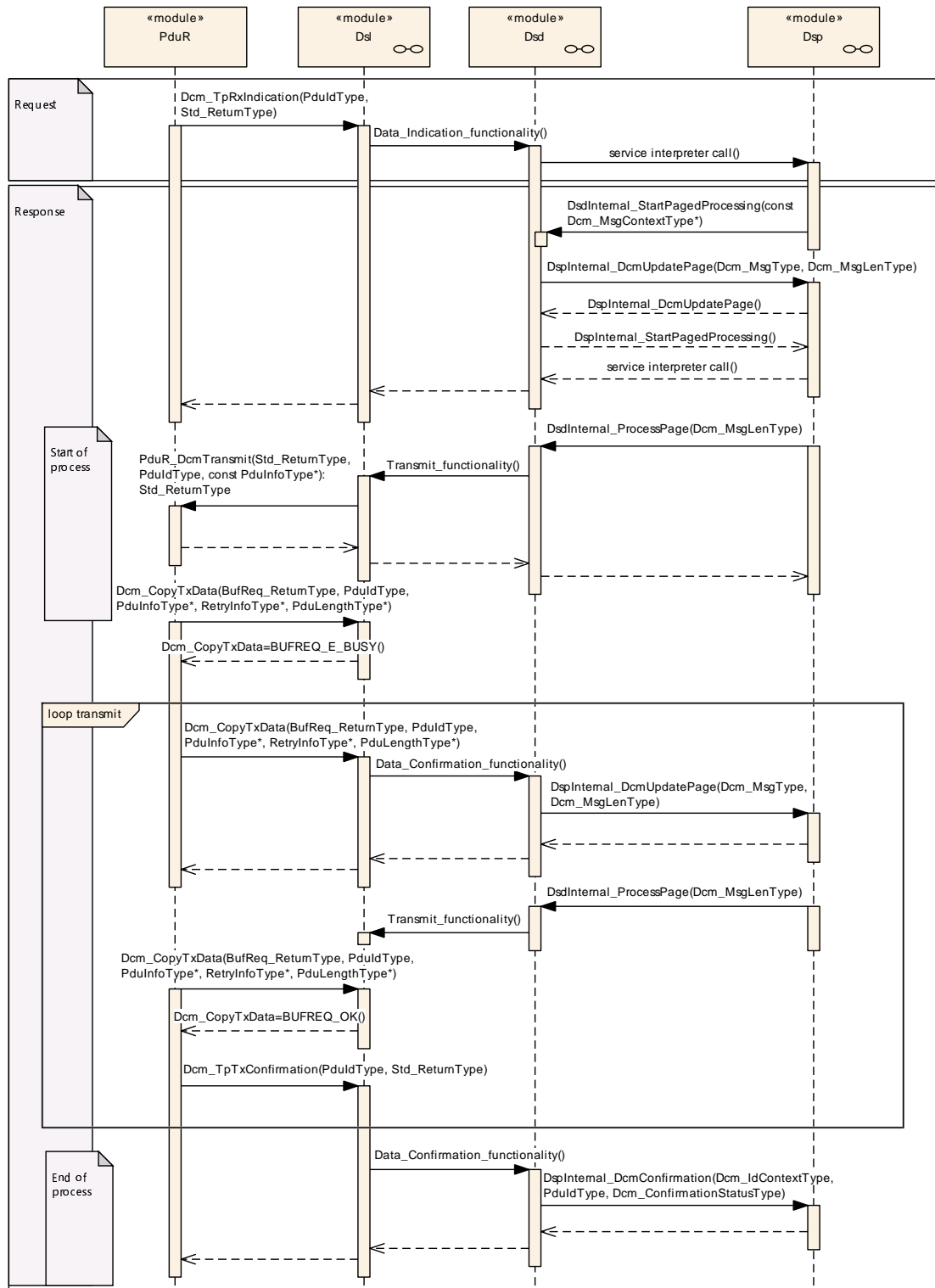


Figure 9.15

The following flow is processed in case no error occurs on the Application side:

Start of process:

- 4) `DsdInternal_StartPagedProcessing()`: With this API, the `DSP` submodule gives the complete response length to the `Dcm` module and starts paged-buffer handling. This API starts no transmission!
- 5) `UpdatePage()`: The `Dcm` module requests data to be transmitted.
- 6) `DsdInternal_ProcessPage()`: With this API, the `DSP` submodule requests transmission of the current page.
- 8) `PduR_DcmTransmit()`: The `Dcm` module requests transmission to the lower layers.
- 9) `Dcm_CopyTxData`: The buffer is filled and the `Dcm` module shall return "BUFREQ_OK"(10).

Start of the loop:

- 11) `Dcm_CopyTxData`: The `PduR` module requests the buffer but the buffer is not filled by the `DSP` submodule.
- 12 + 13) `UpdatePage`: The `Dcm` module requests the `DSP` submodule to fill the next page.
- 14) By returning "BUFREQ_E_BUSY", the `Dcm` module indicates that the buffer has to be filled by the `DSP` submodule.
- 15) `DsdInternal_ProcessPage()`: With this API, the `DSP` submodule requests transmission of the current page.
- 17) Then, on the next call of `Dcm_CopyTxData` the buffer is filled and the `Dcm` module shall return "BUFREQ_OK" (18).

LOOP: The flow 10 to 18 is repeated as long data can be sent.

End of the loop:

- n-2 -> n) `Dcm_TpTxConfirmation` When all data is send, the `PduR` module indicates the sending with a confirmation, which is given to the `DSP` submodule.
- The APIs 4, 5 and 6 are needed only for paged-buffer transmission.

Page buffer timeout handling:

The `Dcm` module reacts in the following described way, when the `DSP` submodule starts paged-buffer handling, but is not able to process further on filling the response data. E.g. there are problems to access data from an EEPROM device. When providing the Pagebuffer to the `DSP` submodule (13: `UpdatePage()`), and getting a negative Tx confirmation from underlying Transport Layers, the following error handling is carried out in the `Dcm` module:

- The `Dcm` module stops further processing of paged-buffer (item 15),
- The `Dcm` module requests the `DSP` submodule (14: `DsdInternal_CancelPagedBufferProcessing()`) to stop further processing of `PagedBuffer`.

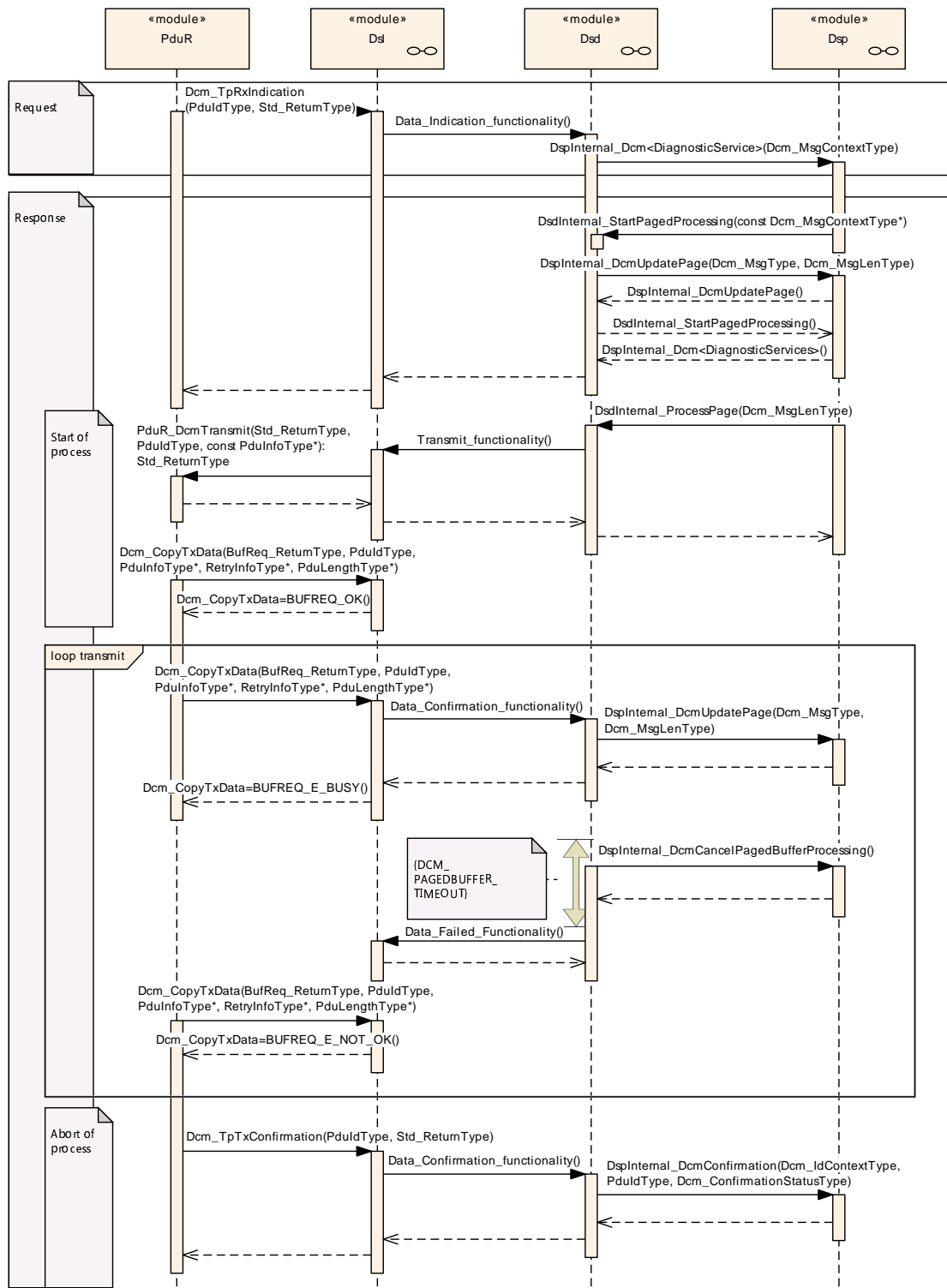


Figure 9.16

9.2.10 Process copy data in reception

Please refer to Figure 9 "CanTp I-PDU reception" in [10, SWS PduR].

9.2.11 Process copy data in transmission

Please refer to Figure 14 "CanTp I-PDU transmission" in [10, SWS PduR].

9.3 DSP (Diagnostic Service Processing)

9.3.1 Interface DSP - DEM (service 0x19, 0x14, 0x85)

Please refer to Section 9 in [12, SWS Dem].

9.3.2 Interface special services

9.3.2.1 Process Diagnostic Session Control

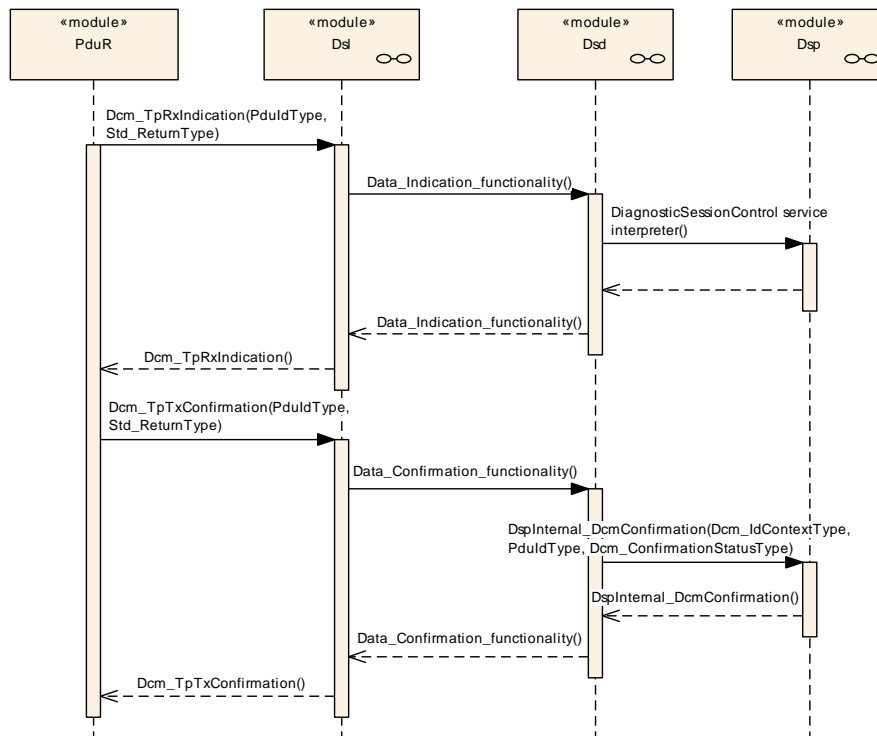


Figure 9.17

Above sequence diagram shows processing of Diagnostic Session Control request from a tester. Note that the new diagnostic session and timing parameters only apply after the transmission confirmation of the server positive response

9.3.2.2 Process Tester Present

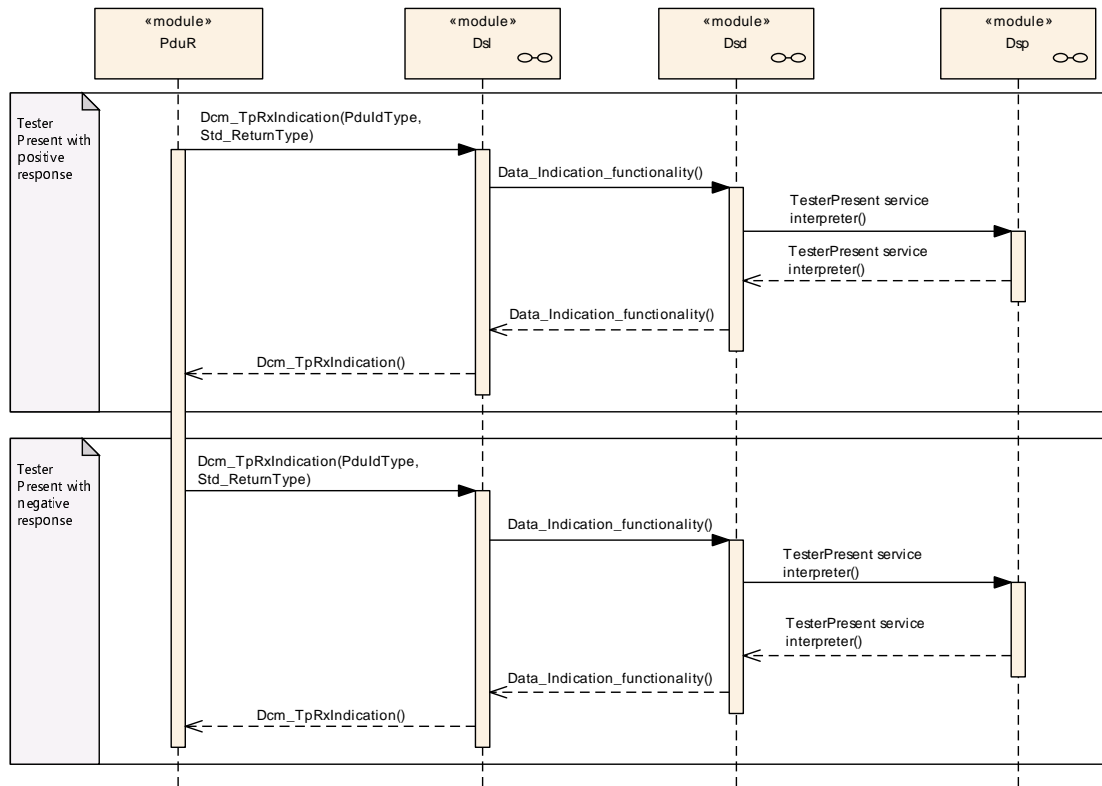


Figure 9.18

Above sequence diagram shows processing of TesterPresent commands, which are not of type functional addressed with subfunction 0x80. These TesterPresent commands are interpreted in the **DSL** submodule (more details can be found in Section 7.3.4.3 Concurrent "TesterPresent keep alive logic").

All the other TesterPresent commands are processed in the following way: On a command TesterPresent the **DSD** submodule calls the **DSP** submodule with the function TesterPresent(). The sequence chart also shows the case when an error occurs and a negative response is sent.

9.3.2.3 Process Security Access

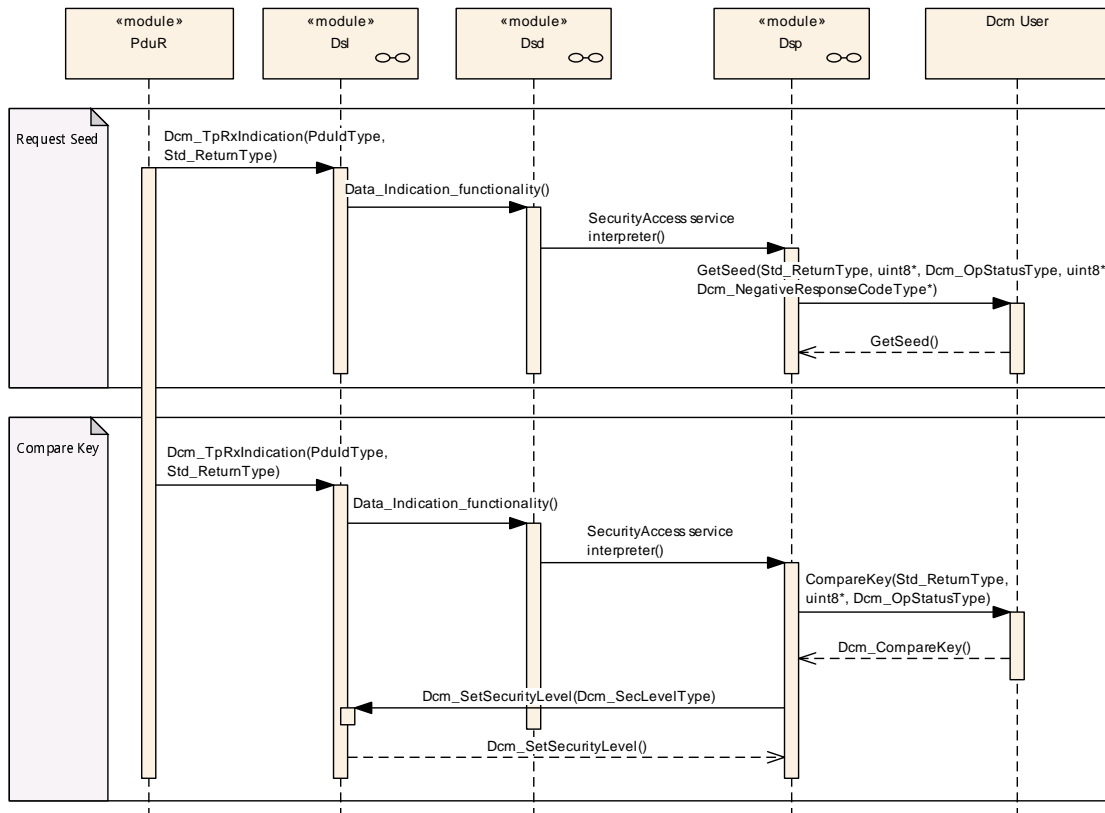


Figure 9.19

To get the security access, the **DSD** submodule has to call the **DSP** submodule to get the seed value from the application. If no error is detected, the seed value is sent in the positive response.

In a second step, the **DSP** submodule gets the key calculated by the tester and requests the application to compare this key with the internal calculated key. If no error occurs, the new access type is set in the **DSL** submodule and a positive response is sent.

9.3.2.4 Process ResponseOnEvent OnDtcChange

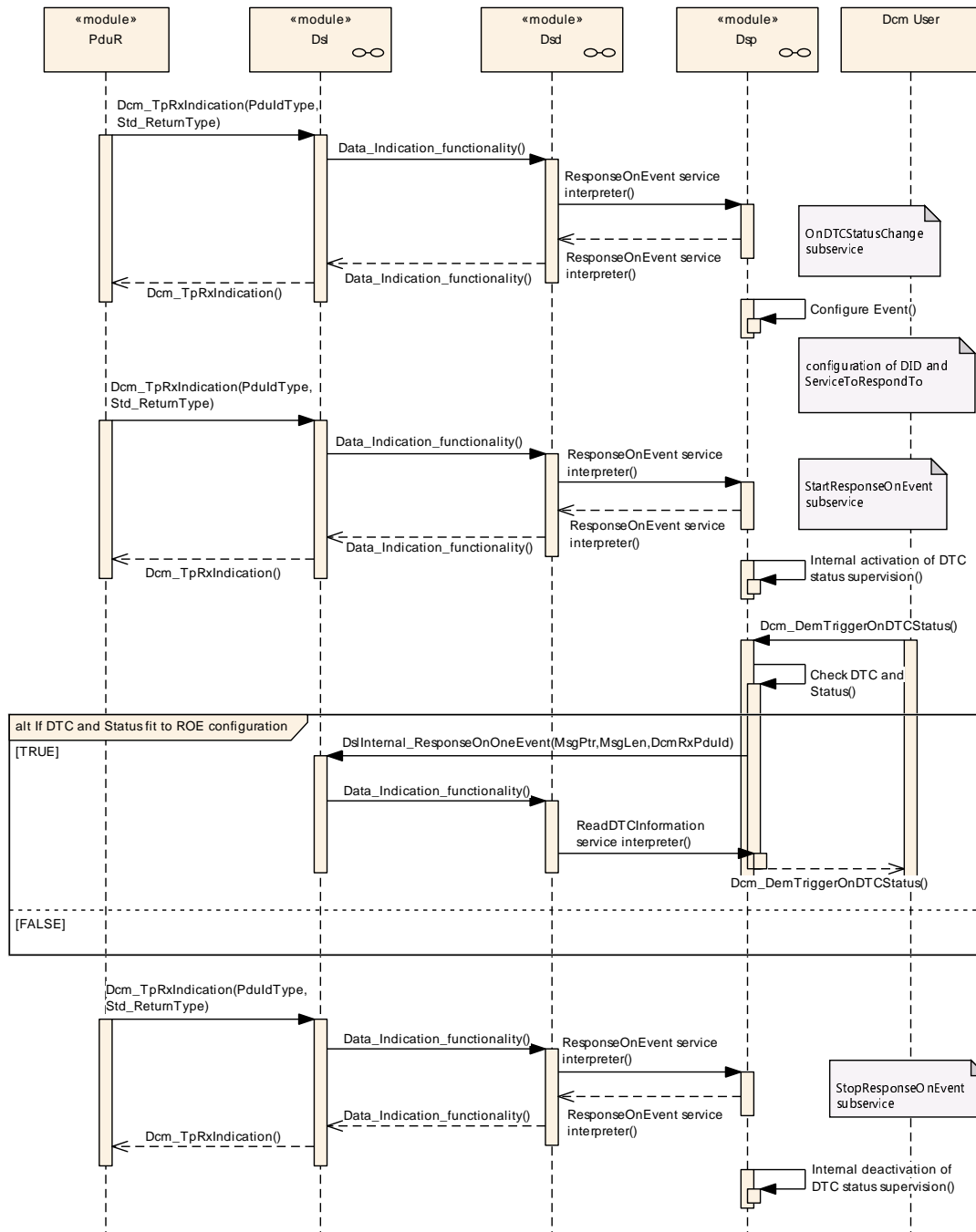


Figure 9.20

Above sequence diagram shows processing of ResponseOnEvent service for sub-service OnDtcChange.

After configuration and activation of the event by the service ResponseOnEvent, the Dcm checks the status of the configured DTC on every call to interface `Dcm_DemTriggerOnDTCStatus` in order to identify if the event shall be trigger. This interface is

called by DEM for any **DTC** status change and independent of the activation/unactivation of ResponseOnEvent.

9.3.2.5 Process ResponseOnEvent OnChangeOfDataIdentifier

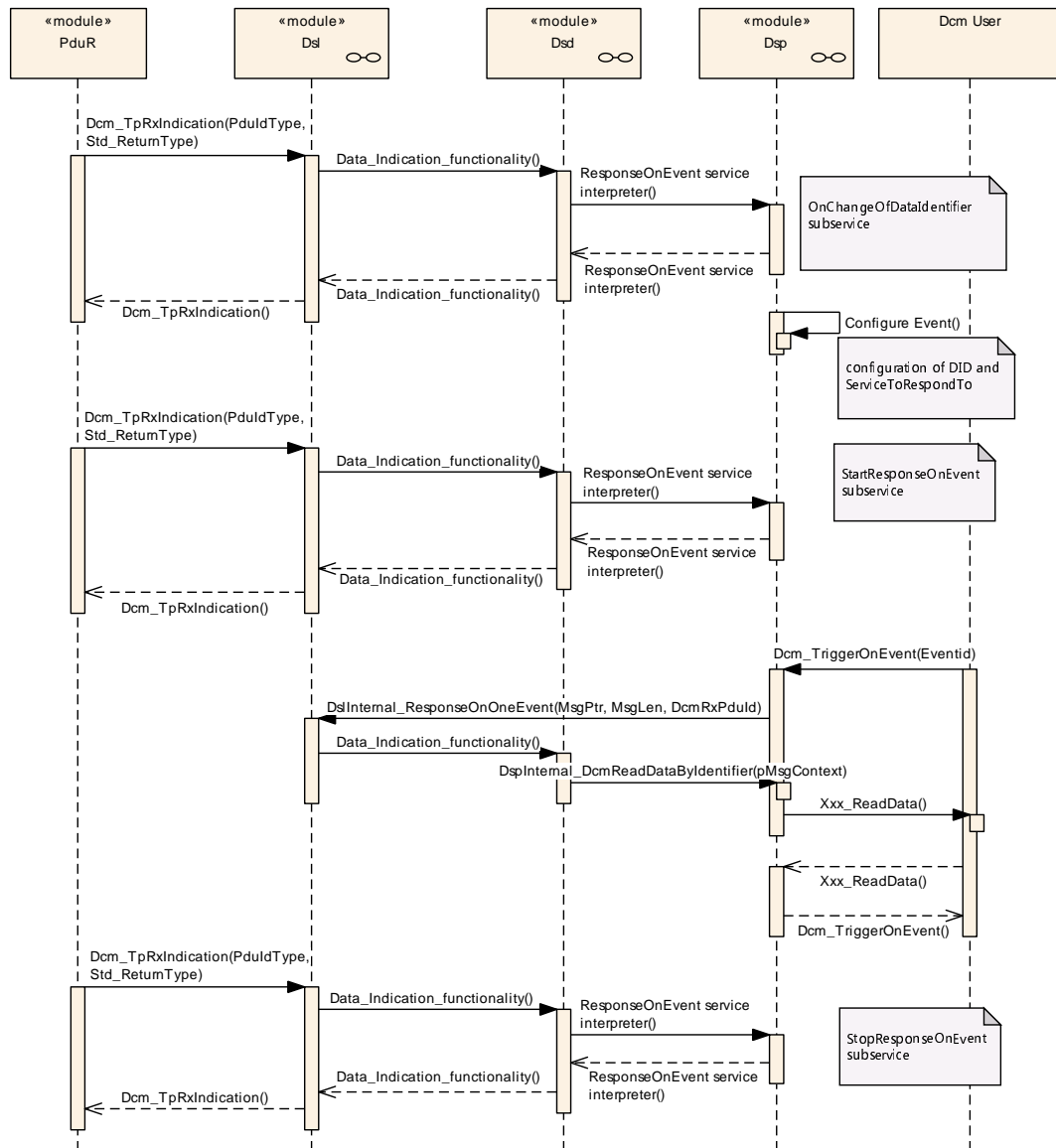


Figure 9.21

Above sequence diagram shows processing of ResponseOnEvent service for subservice OnChangeOfDataIdentifier in the case the event is externally managed (The event can be internally managed, but is not describe in this diagram).

After configuration and external activation of the event by the service ResponseOnEvent, the **Dcm** wait to be trigger by the external module managing this DID.

9.3.2.6 Process Jump to Bootloader

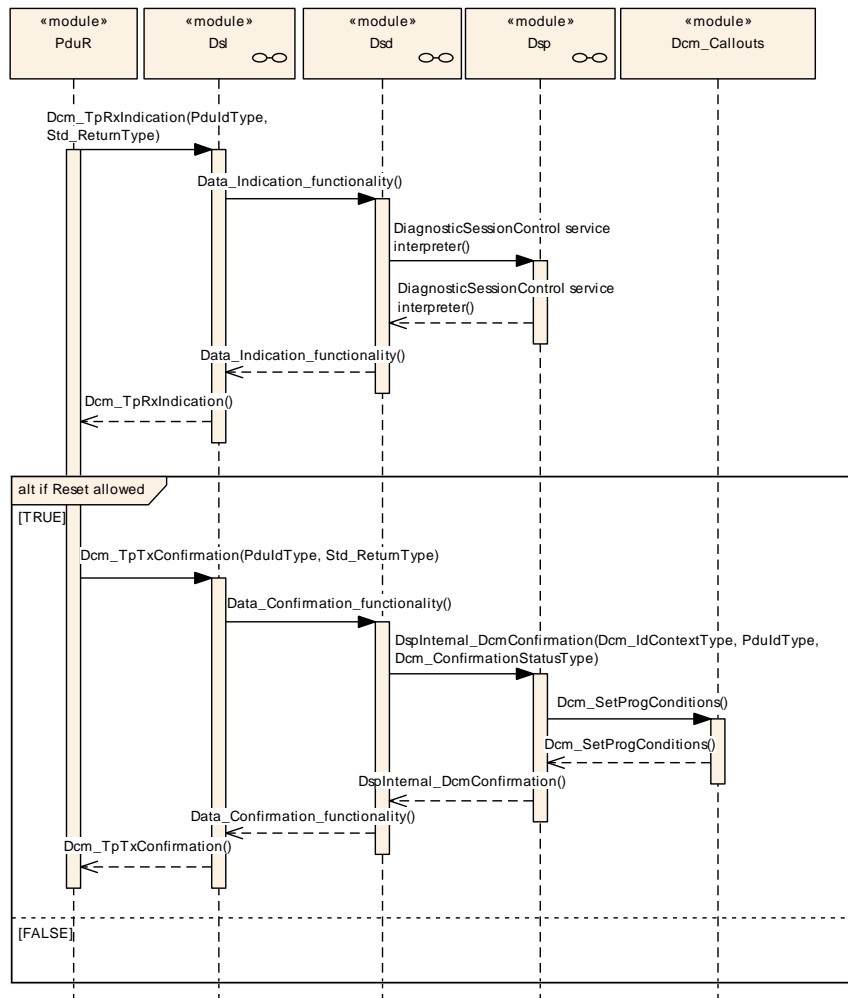


Figure 9.22

Above sequence diagram shows processing of a jump to bootloader on reception of DiagnosticSessionControl. On reception of DiagnosticSessionControl, the **Dcm** checks if the requested session is configured to trigger a jump to bootloader. In positive case, the **Dcm** start the jump to bootloader process:

- Transmission of **NRC 0x78** (ResponsePending)
- On confirmation of transmission of **NRC 0x78**, the **Dcm** calls the callout **DcmSetProgConditions** to store all information needed for the bootloader

10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers. In order to support the specification Chapter 10.1 describes fundamentals.

It also specifies a template (table) you shall use for the parameter specification. We intend to leave Chapter 10.1 in the specification to guarantee comprehension.

Chapter 10.2 specifies the structure (containers) and the parameters of the module <MODULE_ABBREVIATION>.

Chapter 10.4 specifies published information of the module <MODULE_ABBREVIATION>.

10.1 How to read this chapter

For details refer to the chapter 10.1 "Introduction to configuration specification" in SWS_BSWGeneral [7].

10.2 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters describe Chapter 7 and Chapter 8.

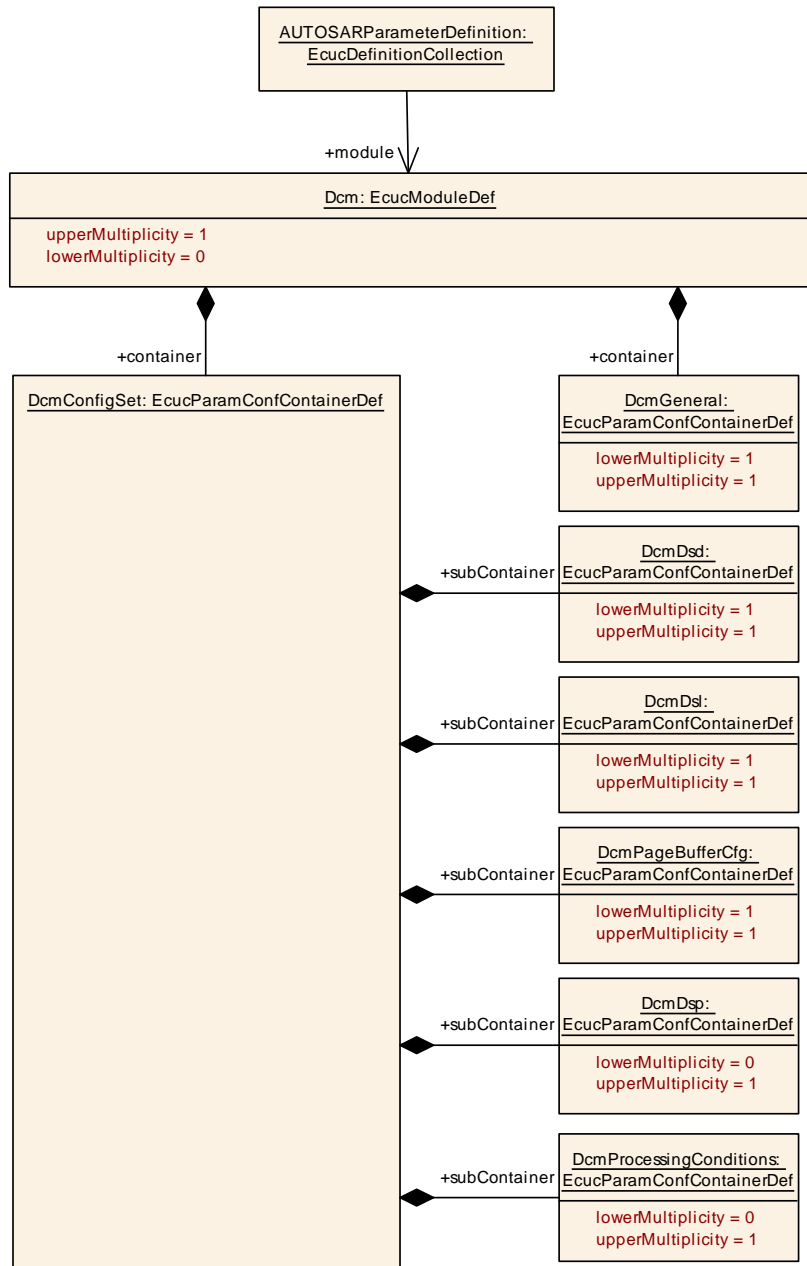


Figure 10.1: Configuration overview

10.2.1 Dcm

Module SWS Item	ECUC_Dcm_01082	
Module Name	Dcm	
Module Description	Configuration of the Dcm (Diagnostic Communications Manager) module.	
Post-Build Variant Support	true	
Supported Config Variants	VARIANT-LINK-TIME, VARIANT-POST-BUILD, VARIANT-PRE-COMPILE	
Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmConfigSet	1	This container contains the configuration parameters and sub containers of the DCM module supporting multiple configuration sets.
DcmGeneral	1	Contains general configuration parameters valid for the entire Dcm module.

10.2.2 DcmConfigSet

SWS Item	[ECUC_Dcm_00819]
Container Name	DcmConfigSet
Parent Container	Dcm
Description	This container contains the configuration parameters and sub containers of the DCM module supporting multiple configuration sets.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDsd	1	These parameters configure the Diagnostic Service Dispatcher submodule.
DcmDsl	1	These parameters configure the Diagnostic Session Layer submodule.
DcmDsp	0..1	These parameters apply to Diagnostic Service Processing. There will always be one set of these parameters per Dcm. Please note: Although the multiplicity is set to 0..1. It can be expected that this container exists in any valid DCM configuration.
DcmPageBufferCfg	1	This container contains the configuration (parameters) for Page Buffer handling
DcmProcessing Conditions	0..1	This container contains the configuration for mode arbitration functionality of the Dcm

10.2.2.1 DcmPageBufferCfg

SWS Item	[ECUC_Dcm_00775]
Container Name	DcmPageBufferCfg
Parent Container	DcmConfigSet
Description	This container contains the configuration (parameters) for Page Buffer handling
Configuration Parameters	

Name	DcmPagedBufferEnabled [ECUC_Dcm_00776]		
Parent Container	DcmPageBufferCfg		
Description	Allow to enable or disable the Paged buffer mechanism. true = Paged buffer handling enabled false = Paged Buffer handling disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

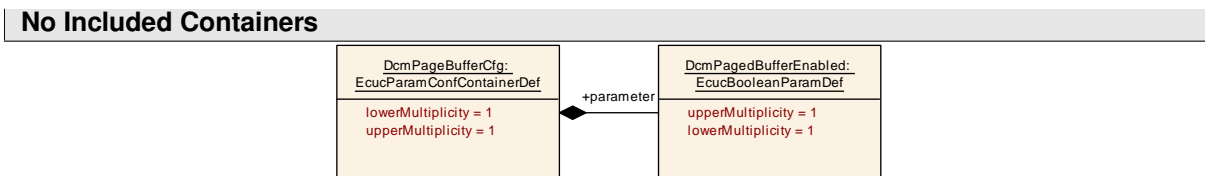


Figure 10.2: DcmPageBufferCfg configuration overview

10.2.2.2 DcmProcessingConditions

SWS Item	[ECUC_Dcm_00932]
Container Name	DcmProcessingConditions
Parent Container	DcmConfigSet
Description	This container contains the configuration for mode arbitration functionality of the Dcm
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmModeCondition	1..*	<p>This container contains the configuration of a mode condition or an environmental conditions which can be used as argument in DcmModeRules.</p> <p>One DcmModeCondition shall contain either one DcmSwcModeRef or one DcmBswModeRef or one DcmSwcSRDataElementRef.</p> <p>Please note that the Dcm acts as well as mode manager. Therefore the references DcmSwcModeRef or one DcmBswModeRef. might point to provided ModeDeclarationGroupPrototypes of the Dcm itself as well as to provided ModeDeclarationGroupPrototypes of other Bsw Modules or software components.</p> <p>In case of a configured DcmSwcModeRef or DcmBswModeRef only the DcmConditionType DCM_EQUALS or DCM_EQUALS_NOT are applicable.</p> <p>In case of DcmSwcSRDataElementRef all literals of DcmConditionType are possible.</p>
DcmModeRule	1..*	<p>This container contains the configuration of a mode rule which represents a logical expression with DcmModeConditions or other DcmModeRules as arguments.</p> <p>All arguments are processed with the operator defined by DcmLogicalOperator, for instance: Argument_A AND Argument_B AND Argument_C</p>

10.2.3 DcmDsd

10.2.3.1 DcmDsd

SWS Item	[ECUC_Dcm_00688]
Container Name	DcmDsd
Parent Container	DcmConfigSet
Description	These parameters configure the Diagnostic Service Dispatcher submodule.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDsdServiceRequestManufacturerNotification	0..*	<p>The name of this container is used to define the name of the R-Port through which the DCM accesses the interface ServiceRequestNotification. The R-Port is named ServiceRequestManufacturerNotification_{Name} where {Name} is the name of the container DcmDsdServiceRequestManufacturerNotification.</p> <p>The lowerMultiplicity is 0: If container DcmDsdServiceRequestManufacturerNotification does not exist the Indication API is not available.</p>
DcmDsdServiceRequestSupplierNotification	0..*	<p>The name of this container is used to define the name of the R-Port through which the DCM accesses the interface ServiceRequestNotification. The R-Port is named ServiceRequestSupplierNotification_<SWC> where <SWC> is the name of the container DcmDsdServiceRequestSupplierNotification.</p> <p>The lowerMultiplicity is 0: If the container DcmDsdRequestSupplierNotification does not exist the Indication API is not available.</p>
DcmDsdServiceTable	1..256	<p>This container contains the configuration (DSD parameters) for a Service Identifier Table.</p> <p>Note: It is allowed to add OBD services to a DcmDsdServiceTable related to a UDS Protocol. But it is not allowed to add UDS services to a DcmDsdServiceTable related to an OBD Protocol.</p>

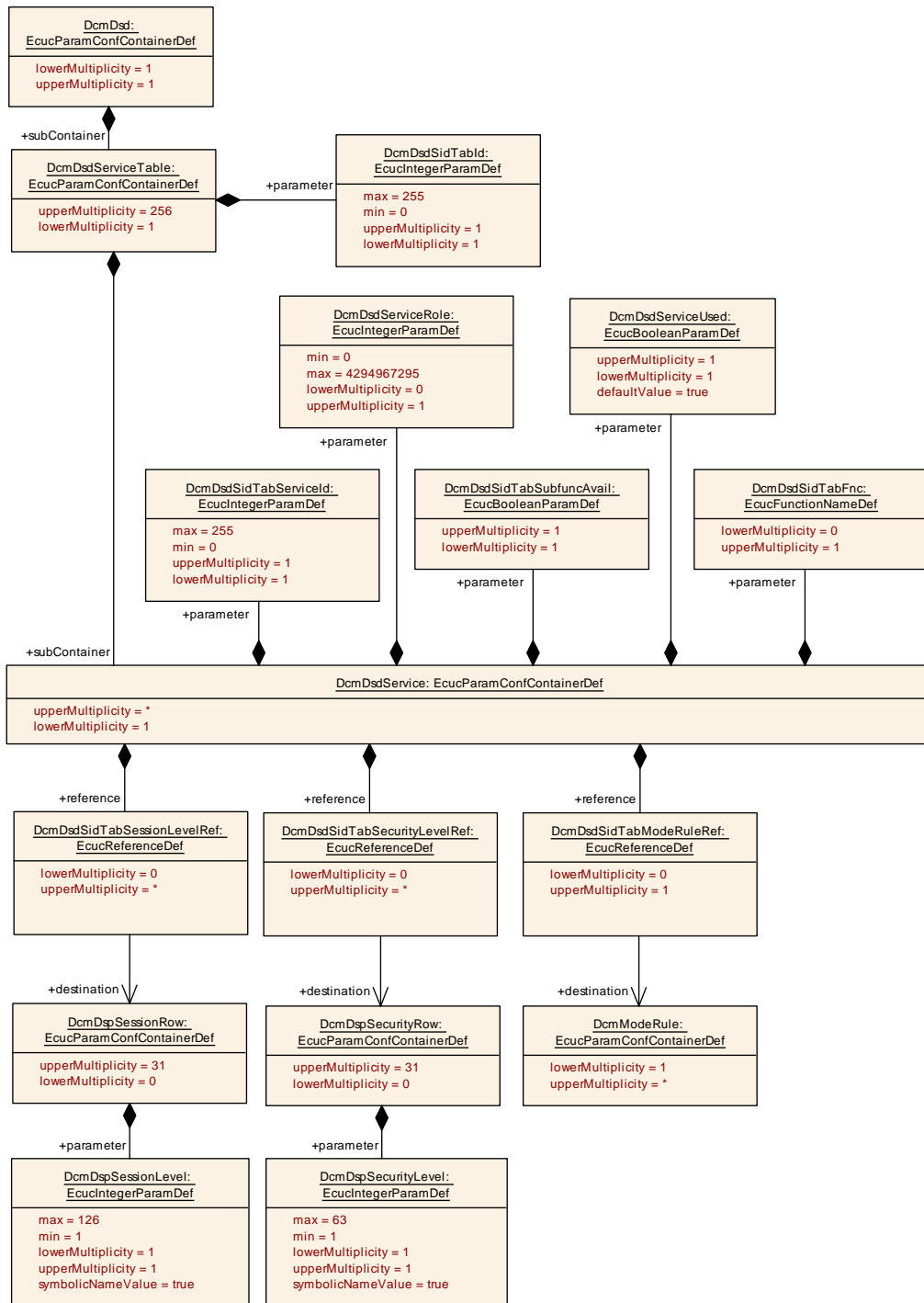


Figure 10.5: DcmDsd configuration overview

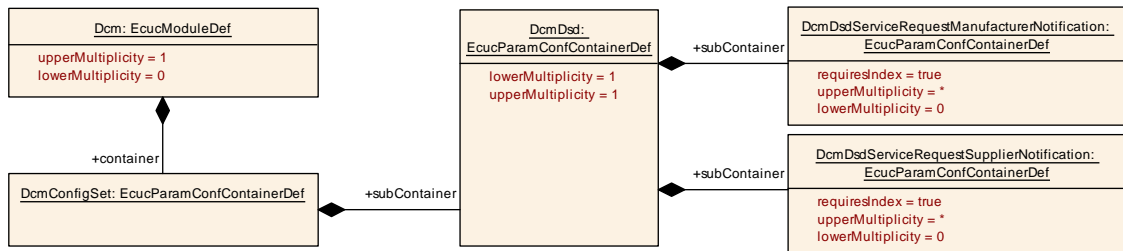


Figure 10.6: DcmDsdIndication configuration overview

10.2.3.2 DcmDsdService

SWS Item	[ECUC_Dcm_00689]
Container Name	DcmDsdService
Parent Container	DcmDsdServiceTable
Description	This container contains the configuration (DSD parameters) for a Service.
Configuration Parameters	

Name	DcmDsdServiceRole [ECUC_Dcm_01139]		
Parent Container	DcmDsdService		
Description	Bitfield where each bit represents one dedicated role. A diagnostic service is granted access if the bit value is 1. If a bit value is 0, the service is not allowed to be executed for that role.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	DcmDsdServiceUsed [ECUC_Dcm_01044]		
Parent Container	DcmDsdService		
Description	<p>Allows to activate or deactivate the usage of a Service. This parameter can be used for multi-purpose ECUs.</p> <p>true - service is available</p> <p>false - service is not available</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	true		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	DcmDsdSidTabFnc [ECUC_Dcm_00777]		
Parent Container	DcmDsdService		
Description	<p>Callback function of the ECU Supplier specific component for the particular service. The function's prototype is as described for <Module>_<DiagnosticService>. If this parameter is not configured, the service is handled Dcm-internally.</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDsdSidTabServiceId [ECUC_Dcm_00735]		
Parent Container	DcmDsdService		
Description	<p>Identifier of the service.</p> <p>The possible service identifiers are defined in ISO 14229-1 and ISO 15031-5.</p>		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDsdSidTabSubfuncAvail [ECUC_Dcm_00737]		
Parent Container	DcmDsdService		
Description	<p>Information about whether the service has subfunctions or not. This parameter is used for the handling of the "suppressPosRspMsgIndicationBit" as defined in ISO 14229-1, which can be used as a reference for the configuration.</p> <p>true - service has subfunctions, suppressPosRspMsgIndicationBit is available</p> <p>false - service has no subfunctions, suppressPosRspMsgIndicationBit is not available</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDsdSidTabModeRuleRef [ECUC_Dcm_00918]		
Parent Container	DcmDsdService		
Description	Reference to a DcmDspModeRule which controls the execution of the service. If there is no reference configured, no mode rule check shall be performed.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		

Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDsdSidTabSecurityLevelRef [ECUC_Dcm_00733]		
Parent Container	DcmDsdService		
Description	<p>Reference to a Security Level in which the service is allowed to be executed. Multiple references are allowed for a service.</p> <p>Please refer to ISO 14229-1, ISO 15031-5 and chapter "Verification of the Service Security Access levels."</p> <p>If there is no reference configured, no service security verification shall be performed.</p>		
Multiplicity	0..*		
Type	Reference to DcmDspSecurityRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDsdSidTabSessionLevelRef [ECUC_Dcm_00734]		
Parent Container	DcmDsdService		
Description	<p>Reference to a Session Level in which the service is allowed to be executed. Multiple references are allowed for a service.</p> <p>Please refer to ISO 14229-1, ISO 15031-5 and chapter "Verification of the Diagnostic Session".</p> <p>If there is no reference configured, no diagnostic session verification shall be performed.</p>		
Multiplicity	0..*		
Type	Reference to DcmDspSessionRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDsdSubService	0..*	This container contains the configuration (DSD parameters) for a subservice of a service. Only those services may have subservices, which have the DcmDsdSidTabSubfuncAvail configured as TRUE.

10.2.3.3 DcmDsdServiceRequestManufacturerNotification

SWS Item	[ECUC_Dcm_00681]
Container Name	DcmDsdServiceRequestManufacturerNotification
Parent Container	DcmDsd
Description	<p>The name of this container is used to define the name of the R-Port through which the DCM accesses the interface ServiceRequestNotification. The R-Port is named ServiceRequestManufacturerNotification_{Name} where {Name} is the name of the container DcmDsdServiceRequestManufacturerNotification.</p> <p>The lowerMultiplicity is 0: If container DcmDsdServiceRequestManufacturerNotification does not exist the Indication API is not available.</p> <p>Attributes: requiresIndex=true</p>
Configuration Parameters	

No Included Containers

10.2.3.4 DcmDsdServiceRequestSupplierNotification

SWS Item	[ECUC_Dcm_00816]
Container Name	DcmDsdServiceRequestSupplierNotification
Parent Container	DcmDsd
Description	<p>The name of this container is used to define the name of the R-Port through which the DCM accesses the interface ServiceRequestNotification. The R-Port is named ServiceRequestSupplierNotification_<SWC> where <SWC> is the name of the container DcmDsdServiceRequestSupplierNotification.</p> <p>The lowerMultiplicity is 0: If the container DcmDsdRequestSupplierNotification does not exist the Indication API is not available.</p> <p>Attributes: requiresIndex=true</p>
Configuration Parameters	

No Included Containers

10.2.3.5 DcmDsdServiceTable

SWS Item	[ECUC_Dcm_00732]
Container Name	DcmDsdServiceTable
Parent Container	DcmDsd
Description	<p>This container contains the configuration (DSD parameters) for a Service Identifier Table.</p> <p>Note: It is allowed to add OBD services to a DcmDsdServiceTable related to a UDS Protocol. But it is not allowed to add UDS services to a DcmDsdServiceTable related to an OBD Protocol.</p>
Configuration Parameters	

Name	DcmDsdSidTabId [ECUC_Dcm_00736]	
Parent Container	DcmDsdServiceTable	
Description	Due to using possibly more service tables, the unique DcmDsdSidTabId can be used to identify them.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default Value		
Post-Build Variant Value	false	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDsdService	1..*	This container contains the configuration (DSD parameters) for a Service.

Note : The [Dcm](#) internal interaction with the [DSP](#) is implementation specific and therefore not explicitly configured.

10.2.3.6 DcmDsdSubService

SWS Item	[ECUC_Dcm_00802]
Container Name	DcmDsdSubService
Parent Container	DcmDsdService
Description	This container contains the configuration (DSD parameters) for a subservice of a service. Only those services may have subservices, which have the <code>DcmDsdSidTabSubfuncAvail</code> configured as TRUE.
Configuration Parameters	

Name	DcmDsdSubServiceFnc [ECUC_Dcm_00942]		
Parent Container	DcmDsdSubService		
Description	<p>Callback function of the ECU Supplier specific component for the particular service. The function's prototype is as described for <code><Module>_<DiagnosticService>_<SubService></code>.</p> <p>If this parameter is not configured, the subservice is handled Dcm-internally.</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDsdSubServiceId [ECUC_Dcm_00803]		
Parent Container	DcmDsdSubService		
Description	Identifier of the subservice. The possible subservice identifiers are defined in ISO 14229-1 and ISO 15031-5.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 127		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDsdSubServiceRole [ECUC_Dcm_01140]		
Parent Container	DcmDsdSubService		
Description	Bitfield where each bit represents one dedicated role. A sub-function of a diagnostic service is granted access if the bit value is 1. If a bit value is 0, the service is not allowed to be executed for that role.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	DcmDsdSubServiceUsed [ECUC_Dcm_01047]		
Parent Container	DcmDsdSubService		
Description	<p>Allows to activate or deactivate the usage of a Subservice. This parameter can be used for multi-purpose ECUs.</p> <p>true - subservice is available false - subservice is not available.</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	true		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	DcmDsdSubServiceModeRuleRef [ECUC_Dcm_00924]		
Parent Container	DcmDsdSubService		
Description	<p>Reference to a DcmDspModeRule which controls the execution of the subservice.</p> <p>If there is no reference configured, no mode rule check shall be performed.</p>		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDsdSubServiceSecurityLevelRef [ECUC_Dcm_00812]		
Parent Container	DcmDsdSubService		
Description	<p>Reference to a Security Level in which the subservice is allowed to be executed. Multiple references are allowed for a subservice.</p> <p>Please refer to ISO 14229-1, ISO 15031-5 and chapter "Verification of the Service Security Access levels."</p> <p>If there is no reference configured, no subservice security verification shall be performed.</p>		
Multiplicity	0..*		
Type	Reference to DcmDspSecurityRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDsdSubServiceSessionLevelRef [ECUC_Dcm_00804]		
Parent Container	DcmDsdSubService		
Description	<p>Reference to a Session Level in which the subservice is allowed to be executed. Multiple references are allowed for a subservice.</p> <p>Please refer to ISO 14229-1, ISO 15031-5 and chapter "Verification of the Diagnostic Session".</p> <p>If there is no reference configured, no diagnostic session verification shall be performed.</p>		
Multiplicity	0..*		
Type	Reference to DcmDspSessionRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

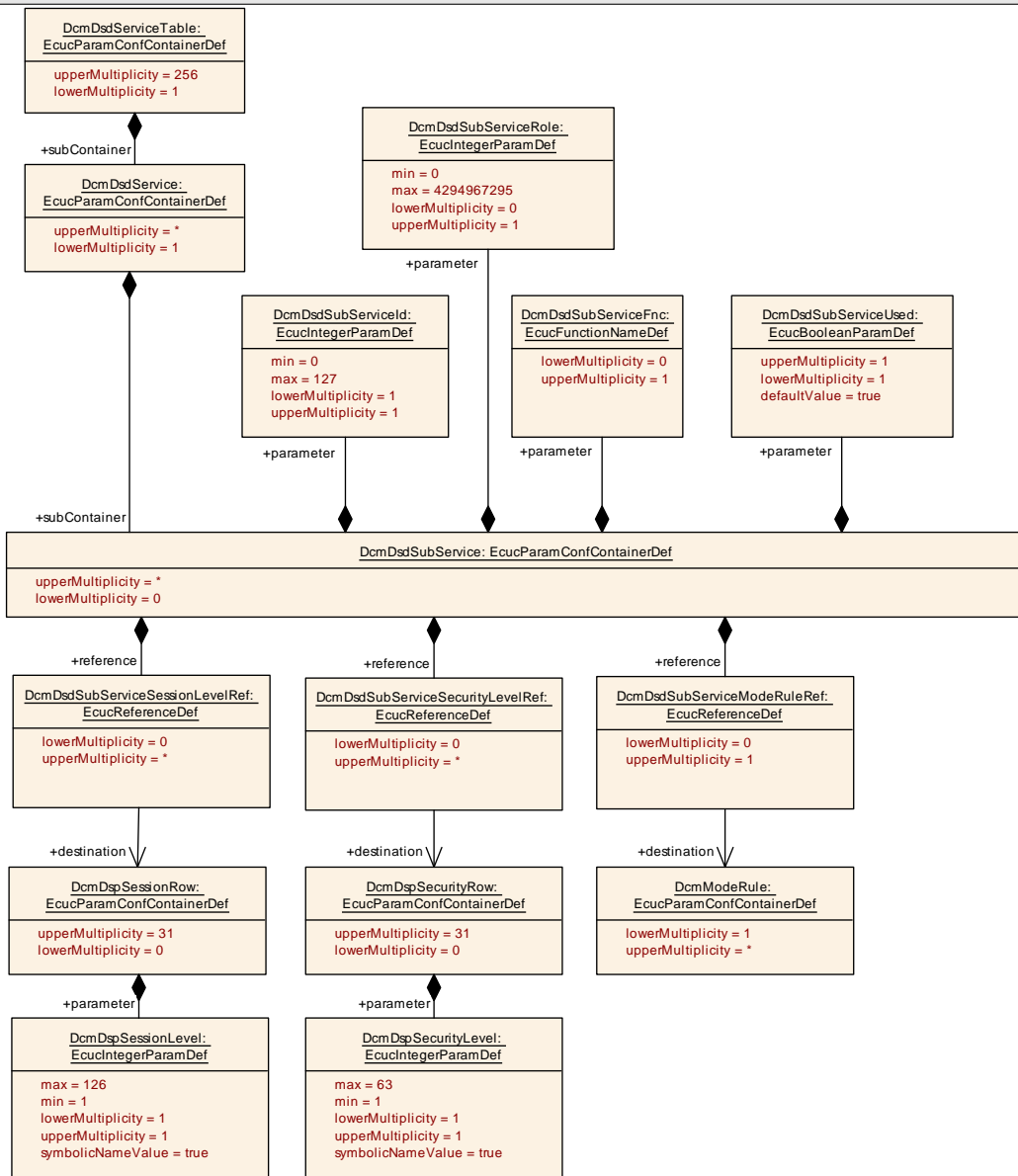


Figure 10.7: DcmDsdSubService configuration overview

10.2.4 DcmDsl

10.2.4.1 DcmDsl

SWS Item	[ECUC_Dcm_00690]
Container Name	DcmDsl
Parent Container	DcmConfigSet
Description	These parameters configure the Diagnostic Session Layer submodule.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDslBuffer	1..256	This container contains the configuration of a diagnostic buffer.
DcmDslCallbackDCMRequestService	0..*	Each DcmDslCallbackDCMRequestService container defines an R-Port with the CallbackDCMRequestServices interface which the Dcm uses to ask permission for protocol changes from the application software. The R-Port has the name CallbackDCMRequestServices_<SWC> where <SWC> is the name of this container.
DcmDslDiagResp	1	This container contains the configuration of the automatic requestCorrectlyReceivedResponsePending response management in the Dcm.
DcmDslProtocol	1	This container contains the configurations of the diagnostic protocols used in Dcm.

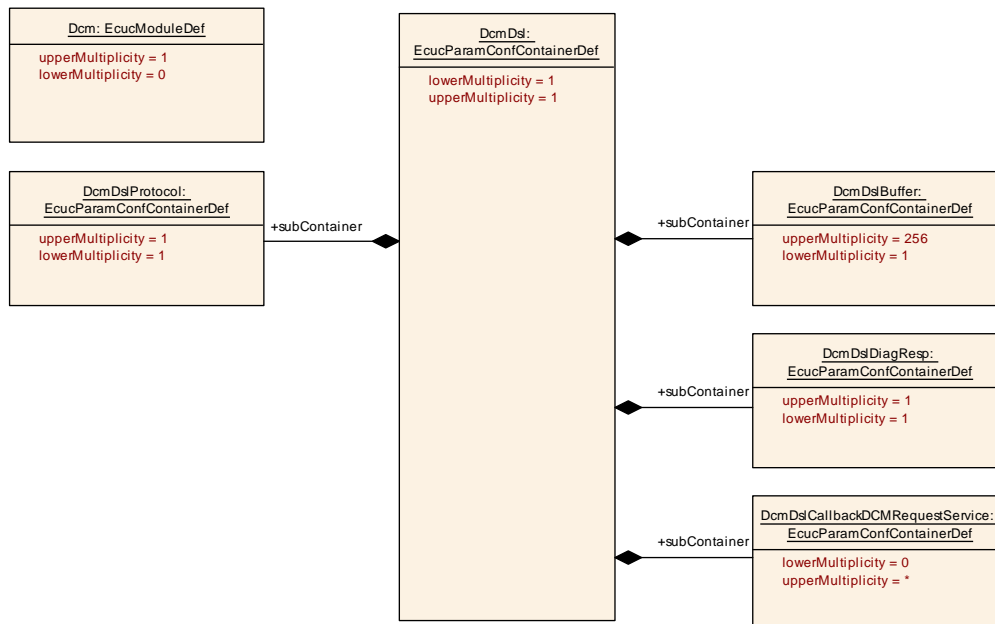


Figure 10.8: DcmDsl configuration overview

10.2.4.2 DcmDslBuffer

SWS Item	[ECUC_Dcm_00739]
Container Name	DcmDslBuffer
Parent Container	DcmDsl
Description	This container contains the configuration of a diagnostic buffer.
Configuration Parameters	

Name	DcmDslBufferSize [ECUC_Dcm_00738]		
Parent Container	DcmDslBuffer		
Description	<p>Size of the diagnostic buffer in bytes.</p> <p>For a linear buffer the size shall be as large as the longest diagnostic message (request or response).</p> <p>For a paged buffer the size has impacts on the application performance.</p>		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	8 .. 4294967294		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

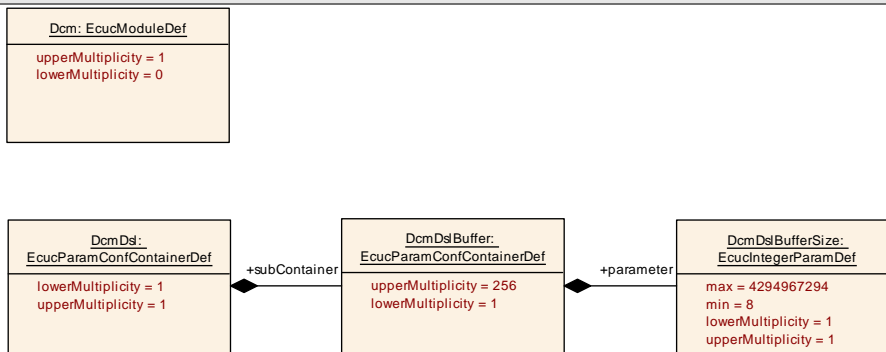


Figure 10.9: DcmDslBuffer configuration overview

10.2.4.3 DcmDslCallbackDCMRequestService

SWS Item	[ECUC_Dcm_00679]
Container Name	DcmDslCallbackDCMRequestService
Parent Container	DcmDsl
Description	<p>Each DcmDslCallbackDCMRequestService container defines an R-Port with the CallbackDCMRequestServices interface which the Dcm uses to ask permission for protocol changes from the application software. The R-Port has the name CallbackDCMRequestServices_<SWC> where <SWC> is the name of this container.</p>
Configuration Parameters	

No Included Containers

10.2.4.4 DcmDslDiagResp

SWS Item	[ECUC_Dcm_00691]
Container Name	DcmDslDiagResp
Parent Container	DcmDsl
Description	This container contains the configuration of the automatic requestCorrectlyReceivedResponsePending response management in the Dcm.
Configuration Parameters	

Name	DcmDslDiagRespMaxNumRespPend [ECUC_Dcm_00693]		
Parent Container	DcmDslDiagResp		
Description	Maximum number of negative responses with response code 0x78 (requestCorrectlyReceivedResponsePending) allowed for a request. If Dcm reaches this limit, an automatic 0x10 (generalReject) final response will be transmitted and the service processing will be cancelled. Value 0x00 means that no NRC 0x78 response will be transmitted.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslDiagRespOnSecondDeclinedRequest [ECUC_Dcm_00914]		
Parent Container	DcmDslDiagResp		
Description	<p>Defines the reaction upon a second request (ClientB) that can not be processed (e.g. due to priority assessment).</p> <p>TRUE: when the second request (Client B) can not be processed, it shall be answered with NRC21 BusyRepeatRequest.</p> <p>FALSE: when the second request (Client B) can not be processed, it shall not be responded.</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.4.5 DcmDslProtocol

SWS Item	[ECUC_Dcm_00694]
Container Name	DcmDslProtocol
Parent Container	DcmDsl
Description	This container contains the configurations of the diagnostic protocols used in Dcm.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDslProtocolRow	1..*	This container contains the configuration of one particular diagnostic protocol used in Dcm.

Container Name	DcmDslProtocolRow
Parent Container	DcmDslProtocol
Description	This container contains the configuration of one particular diagnostic protocol used in Dcm.
Configuration Parameters	

Name	DcmDslProtocolMaximumResponseSize [ECUC_Dcm_01020]		
Parent Container	DcmDslProtocolRow		
Description	This parameter is mandatory and defines the maximum length of the response message in case DcmPagedBufferEnabled == TRUE		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default Value	4095		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslProtocolPriority [ECUC_Dcm_00699]		
Parent Container	DcmDslProtocolRow		
Description	<p>Protocol priority used during protocol preemption. A higher priority protocol may preempt a lower priority protocol. Lower numeric values represent higher protocol priority:</p> <p>0 - Highest protocol priority</p> <p>255 - Lowest protocol priority</p>		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslProtocolRowUsed [ECUC_Dcm_01043]		
Parent Container	DcmDslProtocolRow		
Description	<p>Allows to activate or deactivate the usage of a Protocol. This parameter can be used for multi-purpose ECUs.</p> <p>true - protocol is available</p> <p>false - protocol is not available.</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	true		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	DcmDslProtocolTransType [ECUC_Dcm_00700]		
Parent Container	DcmDslProtocolRow		
Description	This parameter is used only if the protocol is of type DCM_ROE_ON_XXX. It selects the transmission type of the protocol.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	TYPE1	Messages on the DcmTxPduld already used for normal diagnostic responses. The outgoing messages must be synchronized with 'normal outgoing messages', which have a higher priority.	
	TYPE2	Messages on a separate DcmTxPduld.	
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: ECU		

Name	DcmDslProtocolType [ECUC_Dcm_01110]		
Parent Container	DcmDslProtocolRow		
Description	The diagnostic protocol type for the DCM DSL protocol that is being configured. Implementation Type: Dcm_ProtocolType		
Multiplicity	1		
Type	EcucEnumerationParamDef (Symbolic Name generated for this parameter)		
Range	DCM_OBD_ON_CAN	OBD on CAN (ISO15765-4; ISO15031-5)	
	DCM_OBD_ON_FLEXRAY		
	DCM_OBD_ON_IP		
	DCM_PERIODICTRANS_ON_CAN		
	DCM_PERIODICTRANS_ON_FLEXRAY		
	DCM_PERIODICTRANS_ON_IP		
	DCM_ROE_ON_CAN		
	DCM_ROE_ON_FLEXRAY		
	DCM_ROE_ON_IP		
	DCM_SUPPLIER_1	Reserved for SW supplier specific	
	DCM_SUPPLIER_10	Reserved for SW supplier specific	
	DCM_SUPPLIER_11	Reserved for SW supplier specific	
	DCM_SUPPLIER_12	Reserved for SW supplier specific	
	DCM_SUPPLIER_13	Reserved for SW supplier specific	
	DCM_SUPPLIER_14	Reserved for SW supplier specific	
	DCM_SUPPLIER_15	Reserved for SW supplier specific	
	DCM_SUPPLIER_2	Reserved for SW supplier specific	
	DCM_SUPPLIER_3	Reserved for SW supplier specific	
	DCM_SUPPLIER_4	Reserved for SW supplier specific	
	DCM_SUPPLIER_5	Reserved for SW supplier specific	
	DCM_SUPPLIER_6	Reserved for SW supplier specific	
	DCM_SUPPLIER_7	Reserved for SW supplier specific	
	DCM_SUPPLIER_8	Reserved for SW supplier specific	
	DCM_SUPPLIER_9	Reserved for SW supplier specific	
	DCM_UDS_ON_CAN	UDS on CAN (ISO15765-3; ISO14229-1)	
	DCM_UDS_ON_FLEXRAY	UDS on FlexRay (Manufacturer specific; ISO14229-1)	
	DCM_UDS_ON_IP		
DCM_UDS_ON_LIN			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Scope / Dependency	scope: ECU
---------------------------	------------

Name	DcmDspProtocolEcuAddr [ECUC_Dcm_01081]		
Parent Container	DcmDslProtocolRow		
Description	Ecu source address used for diagnostic communication. This parameter is required for Generic Connections.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmSendRespPendOnRestart [ECUC_Dcm_01114]		
Parent Container	DcmDslProtocolRow		
Description	If set to TRUE, the Dcm will send a NRC 0x78 before a transition to bootloader or performing an ECU reset. If set to False, no 0x78 is send in this case.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	true		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmTimStrP2ServerAdjust [ECUC_Dcm_00729]		
Parent Container	DcmDslProtocolRow		
Description	<p>This parameter is used to guarantee that the diagnostic response is available on the bus before reaching P2 by adjusting the current DcmDspSessionP2ServerMax.</p> <p>This parameter mainly represents the software architecture dependent communication delay between the time the transmission is initiated by DCM and the time when the message is actually transmitted to the bus.</p> <p>The parameter value is defined in seconds and must be a multiple of DcmTaskTime.</p>		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. 1]		

Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmTimStrP2StarServerAdjust [ECUC_Dcm_00728]		
Parent Container	DcmDslProtocolRow		
Description	<p>This parameter is used to guarantee that the diagnostic response is available on the bus before reaching P2Star by adjusting the current DcmDspSessionP2StarServerMax.</p> <p>This parameter mainly represents the software architecture dependent communication delay between the time the transmission is initiated by DCM and the time when the message is actually transmitted to the bus.</p> <p>The parameter value is defined in seconds and must be a multiple of DcmTaskTime.</p>		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. 5]		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDemClientRef [ECUC_Dcm_01083]		
Parent Container	DcmDslProtocolRow		
Description	Reference to DemClient in Dem configuration. Used by the Dem to distinguish different client calls.		
Multiplicity	1		
Type	Reference to DemClient		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslProtocolRxBufferRef [ECUC_Dcm_00701]		
Parent Container	DcmDslProtocolRow		
Description	Reference to a configured diagnostic buffer that is used for diagnostic request reception for the protocol.		
Multiplicity	1		
Type	Reference to DcmDslBuffer		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslProtocolSIDTable [ECUC_Dcm_00702]		
Parent Container	DcmDslProtocolRow		
Description	Reference to a service table that is used for diagnostic request processing for this protocol.		
Multiplicity	1		
Type	Reference to DcmDsdServiceTable		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslProtocolTxBufferRef [ECUC_Dcm_00704]		
Parent Container	DcmDslProtocolRow		
Description	Reference to a configured diagnostic buffer that is used for diagnostic response transmission for the protocol.		
Multiplicity	1		
Type	Reference to DcmDslBuffer		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	

Scope / Dependency	scope: ECU
---------------------------	------------

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDslConnection	1..*	<p>This container contains the configuration of a communication channel for one particular protocol.</p> <p>Note that it is allowed to communicate with multiple testers, therefore multiple connections may be configured for a protocol.</p>

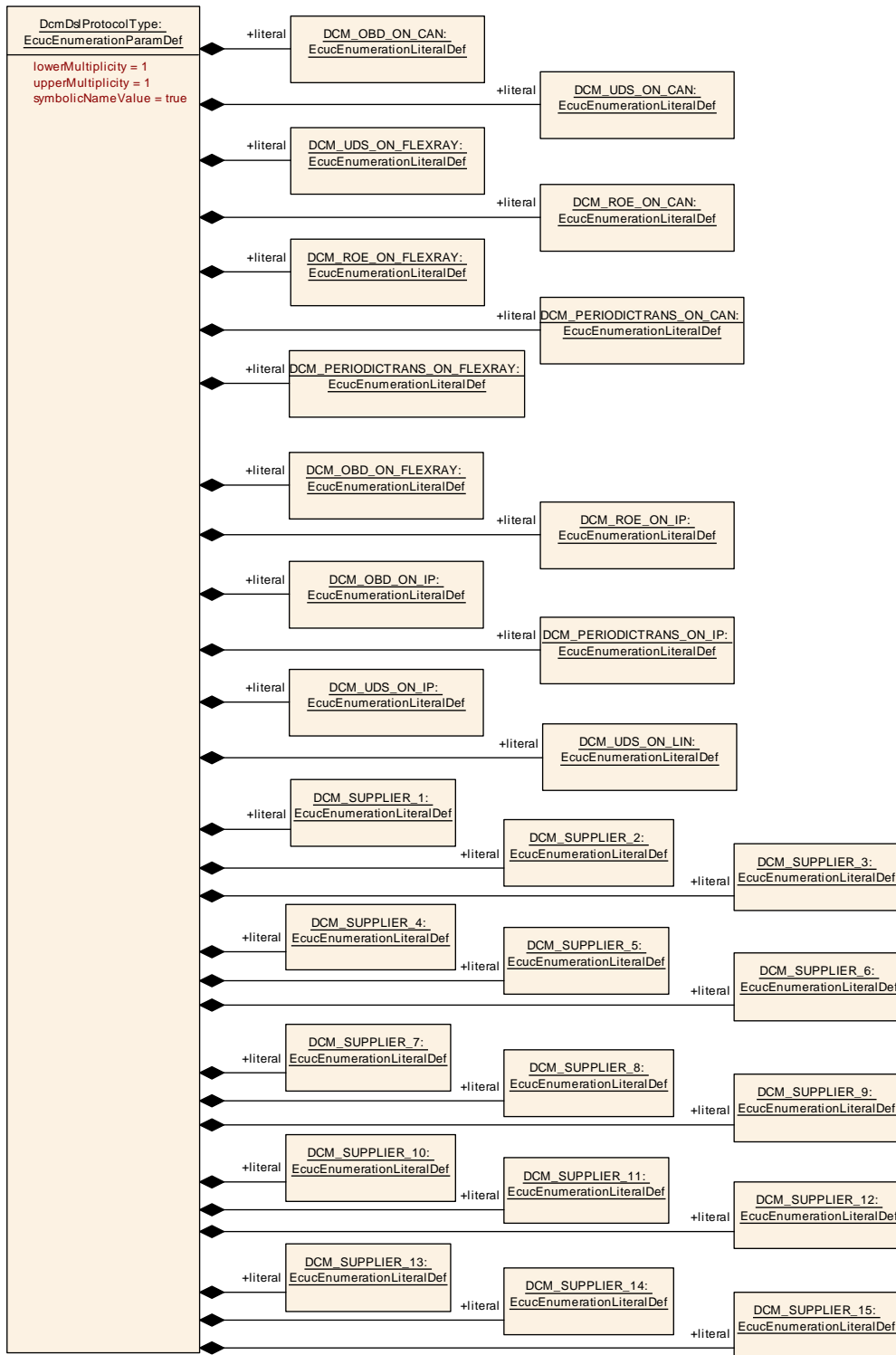


Figure 10.11: DcmDsIProtocolType configuration overview

10.2.4.7 DcmDsIConnection

SWS Item	[ECUC_Dcm_00705]
-----------------	------------------

Container Name	DcmDslConnection
Parent Container	DcmDslProtocolRow
Description	<p>This container contains the configuration of a communication channel for one particular protocol.</p> <p>Note that it is allowed to communicate with multiple testers, therefore multiple connections may be configured for a protocol.</p>
Configuration Parameters	

Container Choices		
Container Name	Multiplicity	Scope / Dependency
DcmDslMainConnection	0..1	This container contains the configuration for a main connection of a diagnostic protocol. Additionally it may contain references to ROE and Periodic connections if the protocol type or protocol transmission type needs them.
DcmDslPeriodicTransmission	0..1	This container contains the configuration of a periodic transmission connection.
DcmDslResponseOnEvent	0..1	<p>This container contains the configuration of a ResponseOnEvent connection.</p> <p>The PDU referenced by this transmission channel can produce meta data items of type TARGET_ADDRESS_16 and SOURCE_ADDRESS_16.</p>

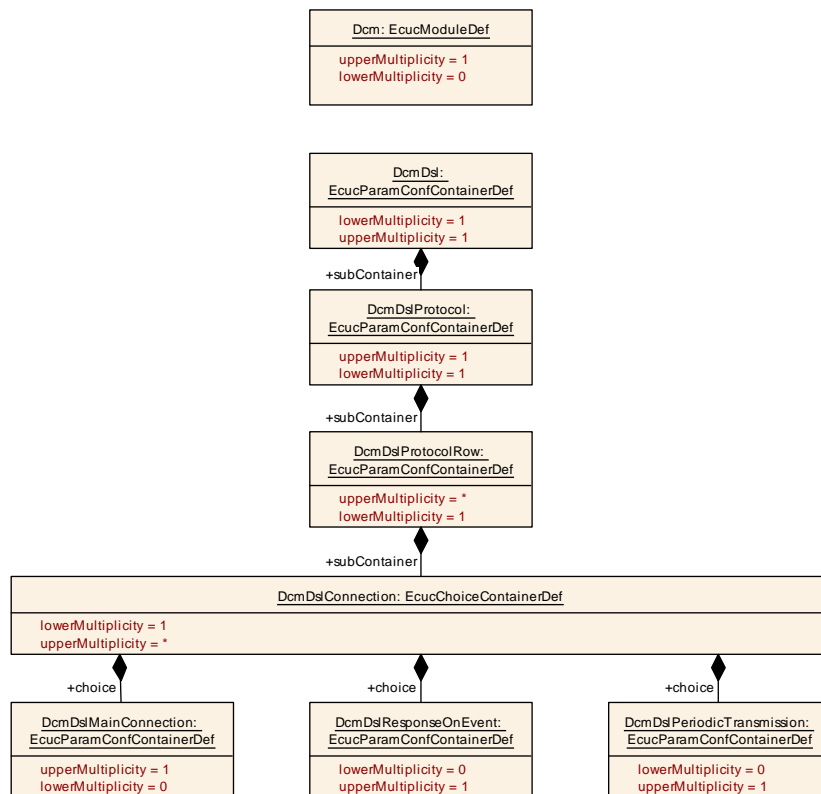


Figure 10.12: DcmDslConnectionTable configuration overview

10.2.4.8 DcmDslMainConnection

SWS Item	[ECUC_Dcm_00706]
Container Name	DcmDslMainConnection
Parent Container	DcmDslConnection
Description	This container contains the configuration for a main connection of a diagnostic protocol. Additionally it may contain references to ROE and Periodic connections if the protocol type or protocol transmission type needs them.
Configuration Parameters	

Name	DcmDslProtocolRxConnectionId [ECUC_Dcm_00826]		
Parent Container	DcmDslMainConnection		
Description	Unique identifier of the tester which uses this connection for diagnostic communication.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslProtocolRxTesterSourceAddr [ECUC_Dcm_01115]		
Parent Container	DcmDslMainConnection		
Description	Tester source address uniquely describes a client and will be used e.g within the jump to Bootloader interfaces. This parameter is not required for generic connections (DcmPdus with MetaDataLength >= 1).		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslPeriodicTransmissionConRef [ECUC_Dcm_00707]		
Parent Container	DcmDslMainConnection		
Description	Reference to a periodic transmission connection which is used for the processing of periodic transmission events.		
Multiplicity	0..1		
Type	Reference to DcmDslPeriodicTransmission		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslProtocolComMChannelRef [ECUC_Dcm_00952]		
Parent Container	DcmDslMainConnection		
Description	Reference to the ComMChannel on which the DcmDslProtocolRxPdu is received and the DcmDslProtocolTxPdu is transmitted.		
Multiplicity	1		
Type	Symbolic name reference to ComMChannel		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslROEConnectionRef [ECUC_Dcm_00708]		
Parent Container	DcmDslMainConnection		
Description	Reference to a ResponseOnEvent connection which is used for the processing of ResponseOnEvent events.		
Multiplicity	0..1		
Type	Reference to DcmDslResponseOnEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDslProtocolRx	1..*	This container contains the configuration parameters of a reception channel in a diagnostic connection. The PDU referenced by this reception channel can consume meta data items of type SOURCE_ADDRESS_16 and TARGET_ADDRESS_16.
DcmDslProtocolTx	0..1	This container contains the configuration parameters of a transmission channel in a diagnostic connection. The PDU referenced by this transmission channel can produce meta data items of type TARGET_ADDRESS_16 and SOURCE_ADDRESS_16.

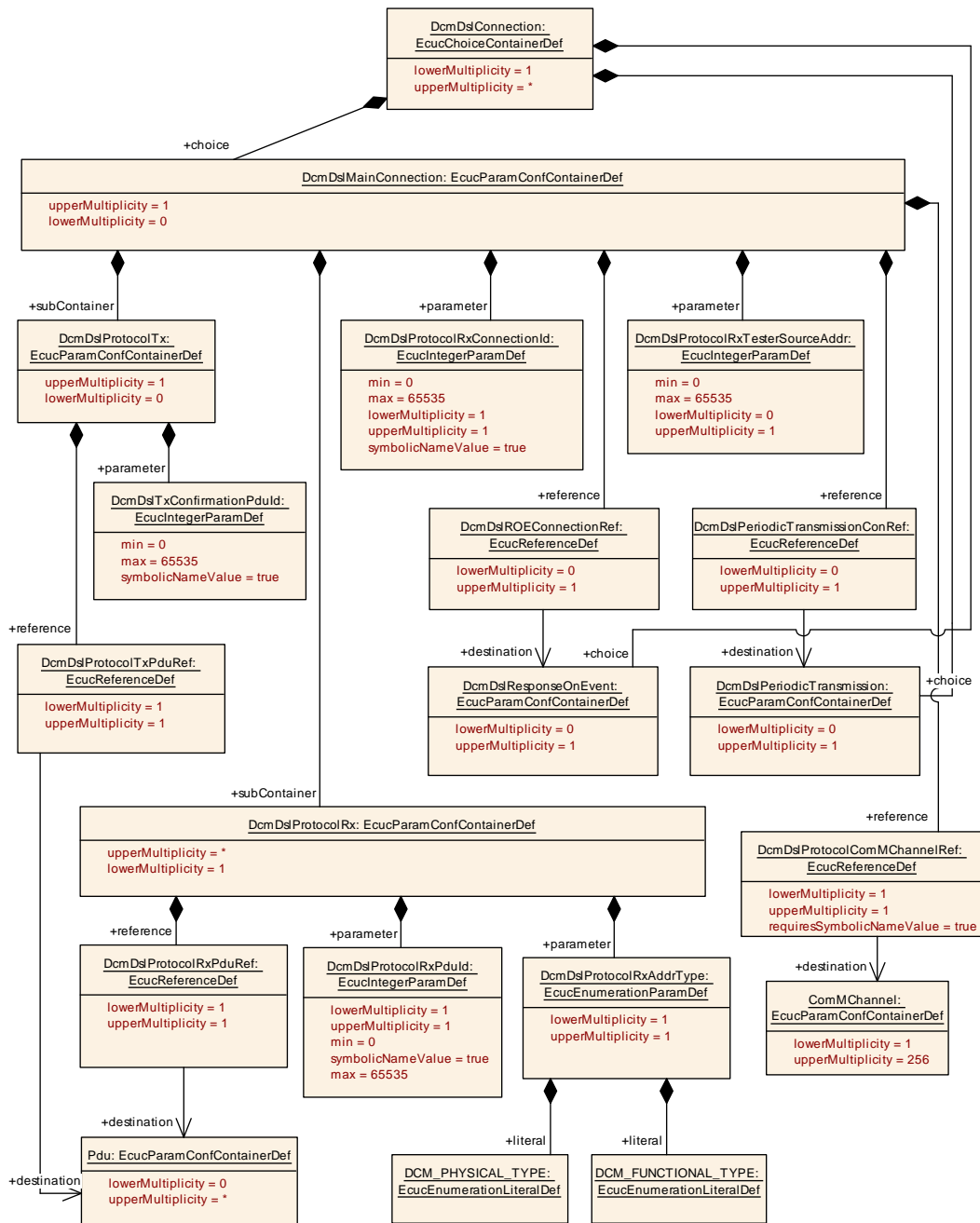


Figure 10.13: DcmDsMainConnection configuration overview

10.2.4.9 DcmDsIProtocolRx

SWS Item	[ECUC_Dcm_00709]
Container Name	DcmDsIProtocolRx
Parent Container	DcmDsMainConnection

Description	<p>This container contains the configuration parameters of a reception channel in a diagnostic connection.</p> <p>The PDU referenced by this reception channel can consume meta data items of type SOURCE_ADDRESS_16 and TARGET_ADDRESS_16.</p>
Configuration Parameters	

Name	DcmDslProtocolRxAddrType [ECUC_Dcm_00710]		
Parent Container	DcmDslProtocolRx		
Description	<p>Selects the addressing type of the reception channel. Physical addressing is used for 1:1 communication, functional addressing is used for 1:N communication.</p> <p>For details refer to ISO 14229-1.</p>		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DCM_FUNCTIONAL_TYP E	FUNCTIONAL = 1 to n communication	
Post-Build Variant Value	DCM_PHYSICAL_TYPE	PHYSICAL = 1 to 1 communications using physical addressing	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslProtocolRxPduld [ECUC_Dcm_00687]		
Parent Container	DcmDslProtocolRx		
Description	Identifier of the PDU that is used for this reception channel.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslProtocolRxPduRef [ECUC_Dcm_00770]		
Parent Container	DcmDslProtocolRx		
Description	Reference to a Pdu in EcuC that is used for this reception channel.		
Multiplicity	1		
Type	Reference to Pdu		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.4.10 DcmDslProtocolTx

SWS Item	[ECUC_Dcm_00711]
Container Name	DcmDslProtocolTx
Parent Container	DcmDslMainConnection
Description	<p>This container contains the configuration parameters of a transmission channel in a diagnostic connection.</p> <p>The PDU referenced by this transmission channel can produce meta data items of type TARGET_ADDRESS_16 and SOURCE_ADDRESS_16.</p>
Configuration Parameters	

Name	DcmDslTxConfirmationPduId [ECUC_Dcm_00864]		
Parent Container	DcmDslProtocolTx		
Description	Identifier of the PDU that is used by the lower level module for transmission confirmation of responses on this channel.		
Multiplicity	1		
Type	EcuIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslProtocolTxPduRef [ECUC_Dcm_00772]		
Parent Container	DcmDslProtocolTx		
Description	Reference to a Pdu in EcuC that is used for this transmission channel.		
Multiplicity	1		
Type	Reference to Pdu		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.4.11 DcmDslPeriodicTransmission

SWS Item	[ECUC_Dcm_00741]
Container Name	DcmDslPeriodicTransmission
Parent Container	DcmDslConnection
Description	This container contains the configuration of a periodic transmission connection.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDslPeriodicConnection	0..255	This container contains the configuration of a transmission channel for a periodic transmission connection. The PDU referenced by this transmission channel can produce meta data items of type TARGET_ADDRESS_16 and SOURCE_ADDRESS_16.

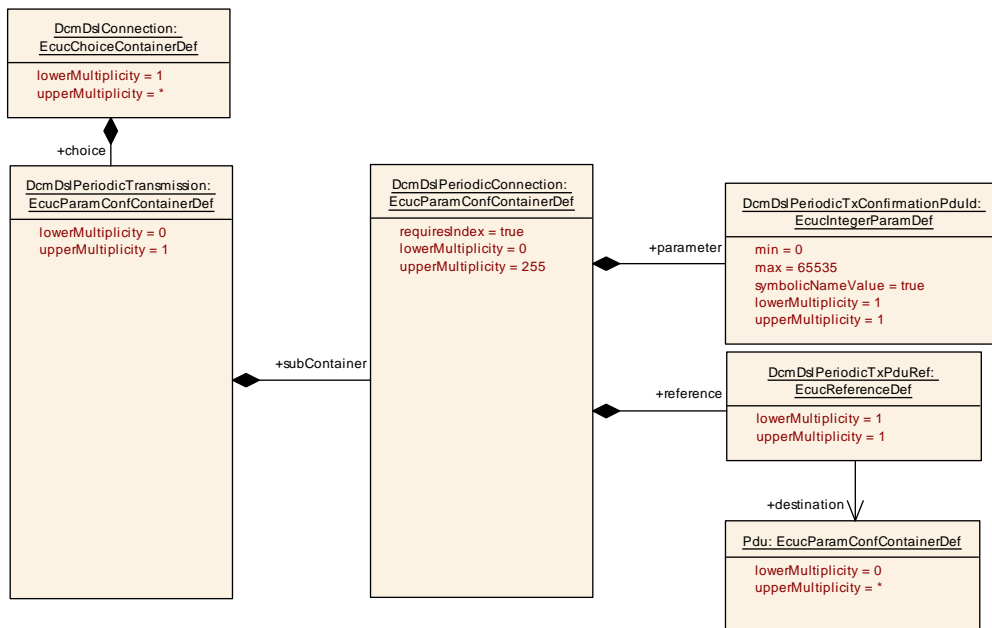


Figure 10.14: DcmDslPeriodicTransmission configuration overview

10.2.4.12 DcmDslPeriodicConnection

SWS Item	[ECUC_Dcm_00897]
Container Name	DcmDslPeriodicConnection
Parent Container	DcmDslPeriodicTransmission
Description	<p>This container contains the configuration of a transmission channel for a periodic transmission connection.</p> <p>The PDU referenced by this transmission channel can produce meta data items of type TARGET_ADDRESS_16 and SOURCE_ADDRESS_16.</p> <p>Attributes: requiresIndex=true</p>
Configuration Parameters	

Name	DcmDslPeriodicTxConfirmationPduId [ECUC_Dcm_00862]	
Parent Container	DcmDslPeriodicConnection	
Description	Identifier of the PDU that is used by the lower level module for transmission confirmation of responses on this channel.	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	
Default Value		
Post-Build Variant Value	false	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslPeriodicTxPduRef [ECUC_Dcm_00742]		
Parent Container	DcmDslPeriodicConnection		
Description	Reference to a Pdu in EcuC that is used for this periodic transmission channel.		
Multiplicity	1		
Type	Reference to Pdu		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.4.13 DcmDslResponseOnEvent

SWS Item	[ECUC_Dcm_00744]
Container Name	DcmDslResponseOnEvent
Parent Container	DcmDslConnection
Description	This container contains the configuration of a ResponseOnEvent connection. The PDU referenced by this transmission channel can produce meta data items of type TARGET_ADDRESS_16 and SOURCE_ADDRESS_16.
Configuration Parameters	

Name	DcmDslRoeTxConfirmationPduId [ECUC_Dcm_00863]	
Parent Container	DcmDslResponseOnEvent	
Description	Identifier of the PDU that is used by the lower level module for transmission confirmation of responses on this connection.	
Multiplicity	0..1	
Type	EcuIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	
Default Value		
Post-Build Variant Multiplicity	false	

Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDslRoeTxPduRef [ECUC_Dcm_00743]		
Parent Container	DcmDslResponseOnEvent		
Description	Reference to a Pdu in EcuC that is used for this ResponseOnEvent transmission connection.		
Multiplicity	0..1		
Type	Reference to Pdu		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

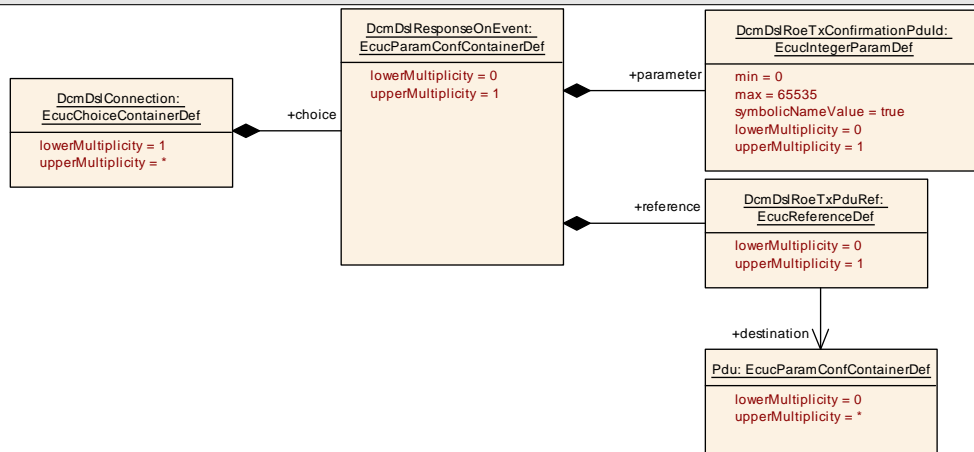


Figure 10.15: DcmDslResponseOnEvent configuration overview

10.2.5 DcmDsp

SWS Item	[ECUC_Dcm_00712]
Container Name	DcmDsp
Parent Container	DcmConfigSet
Description	These parameters apply to Diagnostic Service Processing. There will always be one set of these parameters per Dcm. Please note: Although the multiplicity is set to 0..1. It can be expected that this container exists in any valid DCM configuration.
Configuration Parameters	

Name	DcmDspDataDefaultEndianness [ECUC_Dcm_00987]		
Parent Container	DcmDsp		
Description	Defines the default endianness belonging to a DID, RID or PID if the corresponding data does not define an endianness.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BIG_ENDIAN	Most significant byte shall be stored at the lowest address.	
	LITTLE_ENDIAN	Most significant byte shall be stored at the highest address	
	OPAQUE	Opaque data endianness	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspDDDIDcheckPerSourceDID [ECUC_Dcm_00966]		
Parent Container	DcmDsp		
Description	<p>Defines the check for session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22).</p> <p>true: Dcm module shall check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF</p> <p>false: Dcm module shall not check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF</p>		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspEnableObdMirror [ECUC_Dcm_01061]		
Parent Container	DcmDsp		
Description	DcmDspEnableObdMirror defines whether a DID inside the OBD range (F400-F4FF) and the OBD InfoType range (F800-F8FF) shall get the DID value as defined for OBD on reception of the UDS Service ReadDataByIdentifier (0x22), or not. It also defines whether a RID inside the OBD range (E000-E0FF) shall handle the RID as defined for OBD on reception of the UDS Service RoutineControl (0x31), or not.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspMaxDidToRead [ECUC_Dcm_00638]		
Parent Container	DcmDsp		
Description	Indicates the maximum allowed DIDs in a single "ReadDataByIdentifier" request.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	

Scope / Dependency	scope: ECU
---------------------------	------------

Name	DcmDspMaxPeriodicDidToRead [ECUC_Dcm_00956]		
Parent Container	DcmDsp		
Description	Indicates the maximum allowed periodicDIDs which can be read in a single "ReadDataByPeriodicIdentifier" request.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspPowerDownTime [ECUC_Dcm_00818]		
Parent Container	DcmDsp		
Description	<p>This parameter indicates to the client the minimum time of the stand-by sequence the server will remain in the power-down sequence.</p> <p>The resolution of this parameter is one second per count.</p> <p>The following values are valid: 00 - FE hex: 0 - 254 s powerDownTime; FF hex: indicates a failure or time not available.</p> <p>This value needs to be defined by the integrator according to the ECU capabilities. This parameter has to be available if the service EcuReset, sub-service enableRapidPowerShutDown is configured.</p>		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspAuthentication	0..1	This container contains the configuration for the UDS service Authentication (0x29).
DcmDspClearDTC	0..1	This container contains the configuration for the Clear DTC service.
DcmDspComControl	0..1	Provides the configuration of the CommunicationControl mechanism.
DcmDspCommon Authorization	0..*	This container contains the configuration (parameters) for the common Authorization being equal for several services / sub-services.
DcmDspControlDTC Setting	0..1	Provide the configuration of the ControlDTCSetting mechanism.
DcmDspData	0..*	This container contains the configuration (parameters) of a Data belonging to a DID
DcmDspDataInfo	0..*	This container contains the configuration (parameters) of one Data.
DcmDspDid	0..*	This container contains the configuration (parameters) of the DID.
DcmDspDidInfo	0..*	This container contains the configuration (parameters) of the DID's Info
DcmDspDidRange	0..*	This container defines the DID Range
DcmDspEcuReset	0..1	This container contains the configuration for DcmDspEcuReset service
DcmDspMemory	0..1	This container contains the configuration of the memory access.
DcmDspMemoryTransfer	0..1	This container contains the configuration of the memory transfer.
DcmDspPeriodic Transmission	0..1	This container contains the configuration (parameters) for Periodic Transmission Scheduler.
DcmDspPid	0..*	This container defines the availability of a PID to the DCM.
DcmDspReadDTC Information	0..1	This container contains the configuration for the UDS service ReadDTCInformation (0x19).
DcmDspRequestControl	0..*	This container contains the configuration (parameters) of the "Request control of on-board system, test or component" service (Service \$08). The DCM will request the control using an R-Port requiring a PortInterface RequestControlServices_{Tid}. The R-Port is named RequestControlServices_{Tid} where {Tid} is the name of the container DcmDspRequestControl.
DcmDspRequestFile Transfer	0..1	This container contains the configuration for RequestFileTransfer. This container only exists if RequestFileTransfer is configured.
DcmDspRoe	0..1	Provide the configuration of the ResponseOnEvent mechanism.

DcmDspRoutine	0..*	This container contains the configuration (parameters) for Routines
DcmDspSecurity	1	This container contains the configuration (DSP parameter) for security level configuration (per security level) Description This container contains Rows of DcmDspSecurityRow
DcmDspSession	1	Parent container holding single rows to configure particular sessions
DcmDspVehInfo	0..*	This container contains the configuration (parameters) for one single VehicleInfoType of service \$09

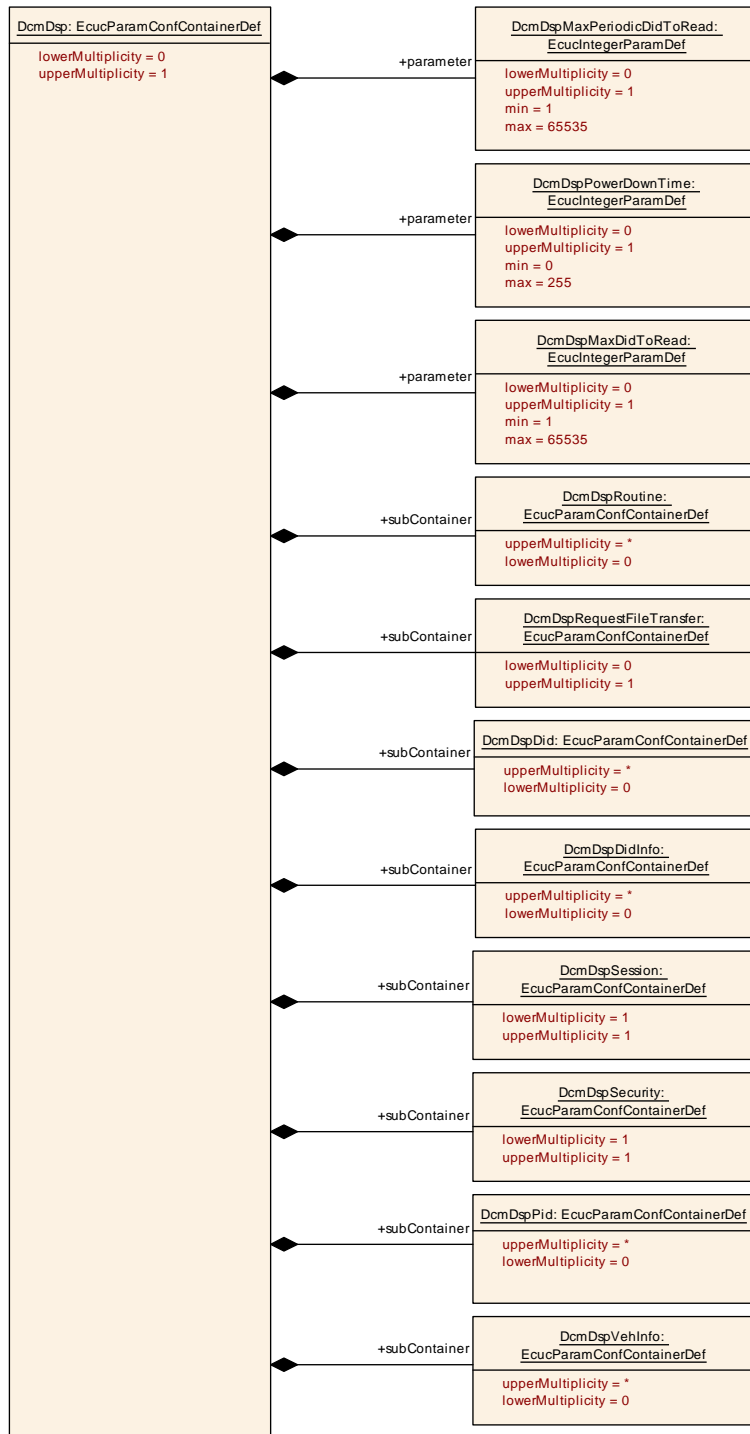


Figure 10.16: DcmDsp configuration overview Part1

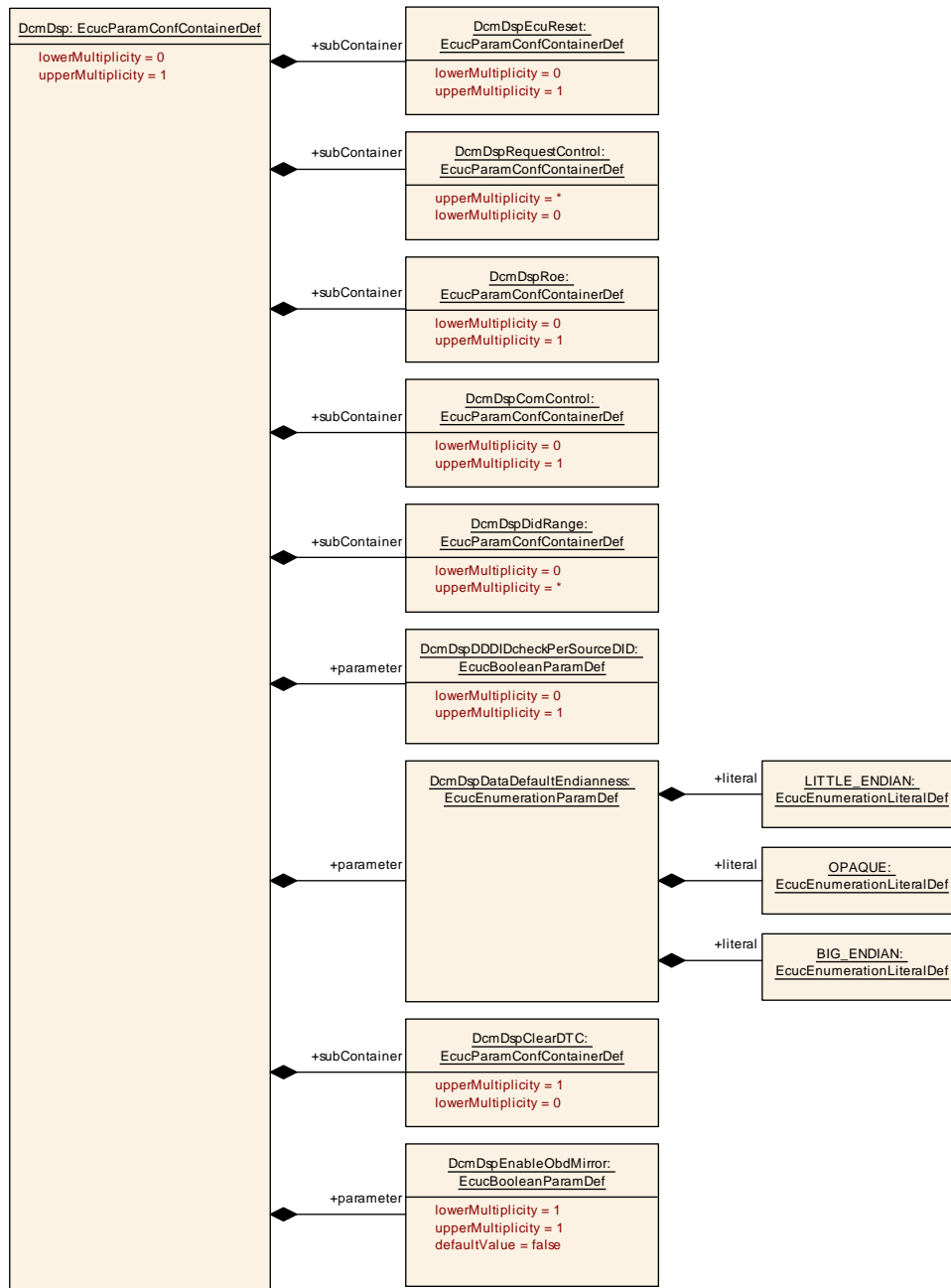


Figure 10.17: DcmDsp configuration overview Part2

10.2.5.1 DcmDspReadDTCInformation

SWS Item	[ECUC_Dcm_01147]
Container Name	DcmDspReadDTCInformation
Parent Container	DcmDsp
Description	This container contains the configuration for the UDS service ReadDTCInformation (0x19).
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspReadDTCInformationUserDefinedFaultMemory	0..255	This container contains the configuration for user defined fault memories in DcmDspReadDTCInformation.

10.2.5.2 DcmDspReadDTCInformationUserDefinedFaultMemory

SWS Item	[ECUC_Dcm_01148]
Container Name	DcmDspReadDTCInformationUserDefinedFaultMemory
Parent Container	DcmDspReadDTCInformation
Description	This container contains the configuration for user defined fault memories in DcmDspReadDTCInformation.
Configuration Parameters	

Name	DcmDspReadDTCInformationUserDefinedFaultMemoryId [ECUC_Dcm_01149]		
Parent Container	DcmDspReadDTCInformationUserDefinedFaultMemory		
Description	Identifier used by external tester to identify the User defined event Memory.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspReadDTCInformationUserDefinedFaultMemoryRole [ECUC_Dcm_01150]		
Parent Container	DcmDspReadDTCInformationUserDefinedFaultMemory		
Description	Bitfield were each bit represents one dedicated role. A user defined fault memory is granted access if the bit value is 1. If a bit value is 0, it is not allowed for that role.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.3 DcmDspAuthentication

SWS Item	[ECUC_Dcm_01151]
Container Name	DcmDspAuthentication
Parent Container	DcmDsp
Description	This container contains the configuration for the UDS service Authentication (0x29).
Configuration Parameters	

Name	DcmDspAuthenticationDeauthenticatedRole [ECUC_Dcm_01153]		
Parent Container	DcmDspAuthentication		
Description	Role used for service authentication verification in the deauthenticated state.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationDefaultSessionTimeOut [ECUC_Dcm_01161]		
Parent Container	DcmDspAuthentication		
Description	Defines the number of seconds after which the Dcm makes a transition to deauthenticated state, in case of no active communication.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. INF[

Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationGeneralNRC [ECUC_Dcm_01159]		
Parent Container	DcmDspAuthentication		
Description	Defines the NRC that shall be send as replacement of all ISO 14229-1 defined NRCs in case of invalid certificate or content.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationRoleSize [ECUC_Dcm_01152]		
Parent Container	DcmDspAuthentication		
Description	Defines the size in bytes for the role element within a certificate.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 4		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationWhiteListDIDMaxSize [ECUC_Dcm_01155]		
Parent Container	DcmDspAuthentication		
Description	Defines the maximum size in bytes for the white list element within a certificate.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationWhiteListMemorySelectionMaxSize [ECUC_Dcm_01157]		
Parent Container	DcmDspAuthentication		
Description	Defines the maximum size in bytes for the white list element within a certificate.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationWhiteListRIDMaxSize [ECUC_Dcm_01156]		
Parent Container	DcmDspAuthentication		
Description	Defines the maximum size in bytes for the white list element within a certificate.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationWhiteListServicesMaxSize [ECUC_Dcm_01154]		
Parent Container	DcmDspAuthentication		
Description	Defines the maximum size in bytes for the white list element within a certificate.		
Multiplicity	0..1		
Type	EcuIntegerParamDef		
Range	1 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationGeneralNRCModeRuleRef [ECUC_Dcm_01158]		
Parent Container	DcmDspAuthentication		
Description	Mode rule that defines if the general NRC shall be send for all failures due to invalid certificate or content.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationPersistStateModeRuleRef [ECUC_Dcm_01160]		
Parent Container	DcmDspAuthentication		
Description	References a mode rule that defines if the authentication state shall be persisted in non-volatile memory.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspAuthenticationConnection	0..*	This container contains the authentication configuration use for a dsl connection.

10.2.5.4 DcmDspAuthenticationConnection

SWS Item	[ECUC_Dcm_01162]
Container Name	DcmDspAuthenticationConnection
Parent Container	DcmDspAuthentication
Description	This container contains the authentication configuration use for a dsl connection.
Configuration Parameters	

Name	DcmDspAuthenticationCertificatePublicKeyStoreJobRef [ECUC_Dcm_01176]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to a CsmJob used to store the public key within the Csm.		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationClientChallengeSignJobRef [ECUC_Dcm_01174]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to a job used to sign the client challenge.		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationConnectionCertificateRef [ECUC_Dcm_01164]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to a KeyMCertificate used to store the certificate within the KeyM.		
Multiplicity	1		
Type	Symbolic name reference to KeyMCertificate		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationConnectionMainConnectionRef [ECUC_Dcm_01163]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to the dsl diagnostic connection that uses this authentication configuration.		
Multiplicity	1		
Type	Reference to DcmDslMainConnection		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationECUCertificateKeyElementRef [ECUC_Dcm_01178]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to a CryptoKeyElement used as server certificate during bi-directional authentication.		
Multiplicity	0..1		
Type	Symbolic name reference to CryptoKeyElement		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationECUCertificateRef [ECUC_Dcm_01177]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to a KeyMCertificate with the server certificate for bi-directional authentication.		
Multiplicity	0..1		
Type	Symbolic name reference to CsmKey		
Post-Build Variant Multiplicity	false		

Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationPublicKeyElementRef [ECUC_Dcm_01166]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to a certificate data element with the public key in the certificate.		
Multiplicity	1		
Type	Symbolic name reference to KeyMCertificateElement		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationRandomJobRef [ECUC_Dcm_01173]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to a certificate parse job used to parse the authentication certificate.		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationRoleElementRef [ECUC_Dcm_01167]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to a certificate data element with the role in the certificate.		
Multiplicity	1		
Type	Symbolic name reference to KeyMCertificateElement		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationTargetIdentificationModeRuleRef [ECUC_Dcm_01172]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to a mode rule that is used to evaluate the target identification.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationVerifyProofOfOwnershipClientJobRef [ECUC_Dcm_01175]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to a CsmJob used to verify the proof of ownership client in the Csm.		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationWhiteListDIDElementRef [ECUC_Dcm_01169]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to a certificate data element with the white list for data identifiers in the certificate.		
Multiplicity	0..1		
Type	Symbolic name reference to KeyMCertificateElement		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationWhiteListMemorySelectionElementRef [ECUC_Dcm_01171]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to a certificate data element with the white list for user defined memory selection in the certificate.		
Multiplicity	0..1		
Type	Symbolic name reference to KeyMCertificateElement		
Post-Build Variant Multiplicity	false		

Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationWhiteListRIDElementRef [ECUC_Dcm_01170]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to a certificate data element with the white list for routine identifiers in the certificate.		
Multiplicity	0..1		
Type	Symbolic name reference to KeyMCertificateElement		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspAuthenticationWhiteListServicesElementRef [ECUC_Dcm_01168]		
Parent Container	DcmDspAuthenticationConnection		
Description	Reference to a certificate data element with the white list in the certificate.		
Multiplicity	1		
Type	Symbolic name reference to KeyMCertificateElement		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.5 Communication Control

10.2.5.5.1 DcmDspComControl

SWS Item	[ECUC_Dcm_00900]
Container Name	DcmDspComControl
Parent Container	DcmDsp
Description	Provides the configuration of the CommunicationControl mechanism.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspComControlAllChannel	0..*	Collection of ComM channels which shall be controlled if all networks are addressed.
DcmDspComControlSetting	0..1	Provide the configuration of the Communication control.
DcmDspComControlSpecificChannel	0..*	Assigns subnet number to ComM channel which will be controlled.
DcmDspComControlSubNode	0..65535	This container gives information about the node identification number and the ComM channel used to address a sub-network.

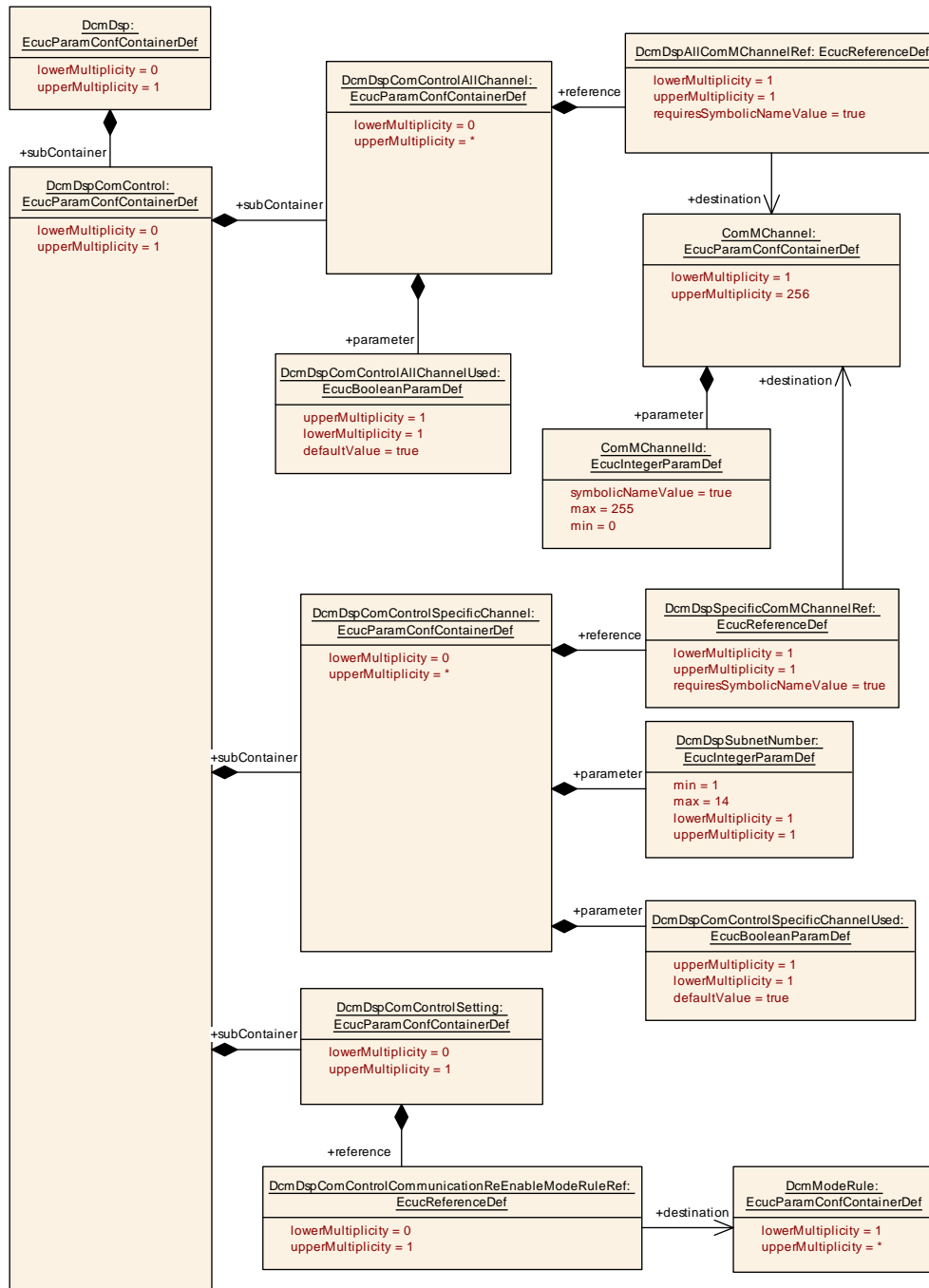


Figure 10.18: DcmDspComControl configuration overview

10.2.5.5.2 DcmDspComControlAllChannel

SWS Item	[ECUC_Dcm_00901]
Container Name	DcmDspComControlAllChannel
Parent Container	DcmDspComControl
Description	Collection of ComM channels which shall be controlled if all networks are addressed.

Configuration Parameters

Name	DcmDspComControlAllChannelUsed [ECUC_Dcm_01045]		
Parent Container	DcmDspComControlAllChannel		
Description	Allow to activate or deactivate the usage of a ComM channel collection to be controlled, for multi purpose ECUs true = ComM channel collection used false = ComM channel collection not used		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	true		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	DcmDspAllComMChannelRef [ECUC_Dcm_00902]		
Parent Container	DcmDspComControlAllChannel		
Description	Reference to ComM channel.		
Multiplicity	1		
Type	Symbolic name reference to ComMChannel		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.5.3 DcmDspComControlSetting

SWS Item	[ECUC_Dcm_00943]
Container Name	DcmDspComControlSetting
Parent Container	DcmDspComControl
Description	Provide the configuration of the Communication control.
Configuration Parameters	

Name	DcmDspComControlCommunicationReEnableModeRuleRef [ECUC_Dcm_00944]		
Parent Container	DcmDspComControlSetting		
Description	Reference to DcmModeRule Mode rule which controls re-enabling of communication by DCM. [ref. SWS_Dcm_00753]		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.5.4 DcmDspComControlSpecificChannel

SWS Item	[ECUC_Dcm_00903]
Container Name	DcmDspComControlSpecificChannel
Parent Container	DcmDspComControl
Description	Assigns subnet number to ComM channel which will be controlled.
Configuration Parameters	

Name	DcmDspComControlSpecificChannelUsed [ECUC_Dcm_01046]		
Parent Container	DcmDspComControlSpecificChannel		
Description	Allow to activate or deactivate the usage of a Subnet assigned to the ComM channel which will be controlled, for multi purpose ECUs. true = Subnet used false = Subnet not used		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	true		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	DcmDspSubnetNumber [ECUC_Dcm_00905]		
Parent Container	DcmDspComControlSpecificChannel		
Description	Subnet Number which controls the specific ComMChannel.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 14		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspSpecificComMChannelRef [ECUC_Dcm_00904]		
Parent Container	DcmDspComControlSpecificChannel		
Description	Reference to ComM channel.		
Multiplicity	1		
Type	Symbolic name reference to ComMChannel		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.5.5 DcmDspComControlSubNode

SWS Item	[ECUC_Dcm_01033]
Container Name	DcmDspComControlSubNode
Parent Container	DcmDspComControl
Description	This container gives information about the node identification number and the ComM channel used to address a sub-network.
Configuration Parameters	

Name	DcmDspComControlSubNodeId [ECUC_Dcm_01031]		
Parent Container	DcmDspComControlSubNode		
Description	The node identification number DcmDspComControlSubNodeId is addressed by the CommunicationControl (0x28) request.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspComControlSubNodeUsed [ECUC_Dcm_01032]		
Parent Container	DcmDspComControlSubNode		
Description	This parameter determines if a node control function is available or not.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	true		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	DcmDspComControlSubNodeComMChannelRef [ECUC_Dcm_01030]		
Parent Container	DcmDspComControlSubNode		
Description	This parameter references a ComM channel where this node is connected to.		
Multiplicity	1		
Type	Symbolic name reference to ComMChannel		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

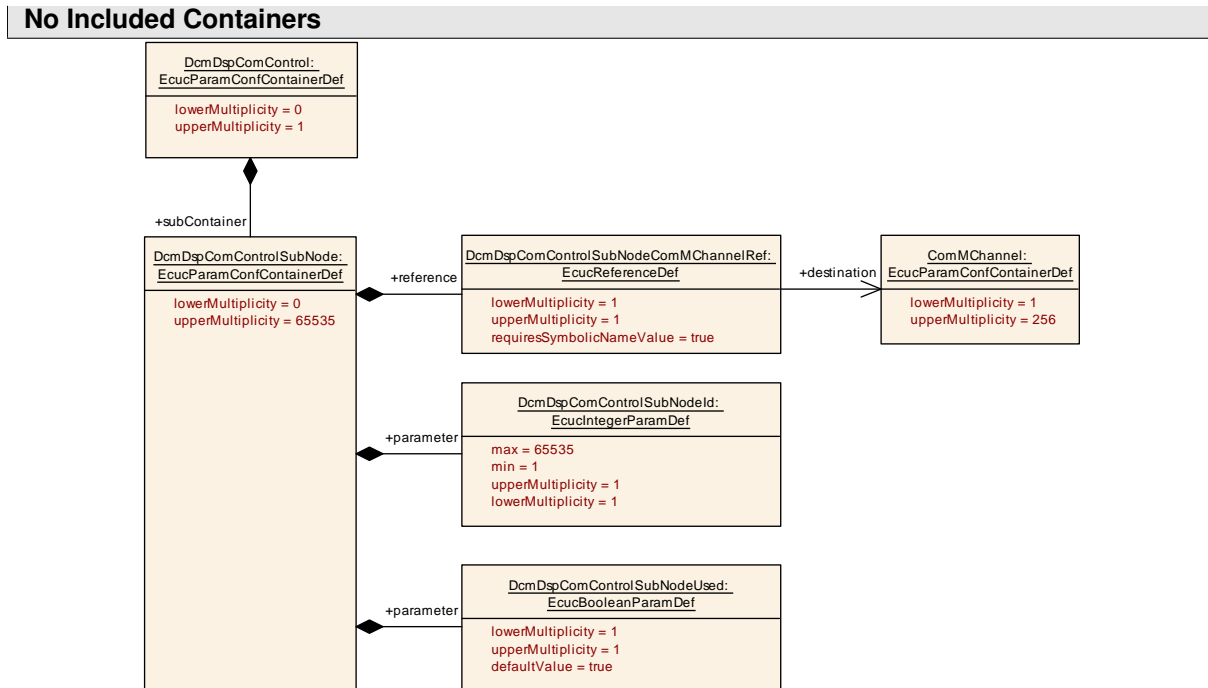


Figure 10.19: DcmDspComControlSubNode configuration overview

10.2.5.6 DcmDspCommonAuthorization

SWS Item	[ECUC_Dcm_01025]
Container Name	DcmDspCommonAuthorization
Parent Container	DcmDsp
Description	This container contains the configuration (parameters) for the common Authorization being equal for several services / sub-services.
Configuration Parameters	

Name	DcmDspCommonAuthorizationModeRuleRef [ECUC_Dcm_01028]		
Parent Container	DcmDspCommonAuthorization		
Description	Reference to DcmModeRule Mode rule which controls this service/ sub-service. If there is no reference, no check of the mode rule shall be done.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspCommonAuthorizationSecurityLevelRef [ECUC_Dcm_01026]		
Parent Container	DcmDspCommonAuthorization		
Description	Reference to DcmDspSecurityRow Security levels allowed to control this service/ sub-service. If there is no reference, no check of security level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSecurityRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspCommonAuthorizationSessionRef [ECUC_Dcm_01027]		
Parent Container	DcmDspCommonAuthorization		
Description	Reference to DcmDspSessionRow Sessions allowed to control this service/ sub-service. If there is no reference, no check of session level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSessionRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

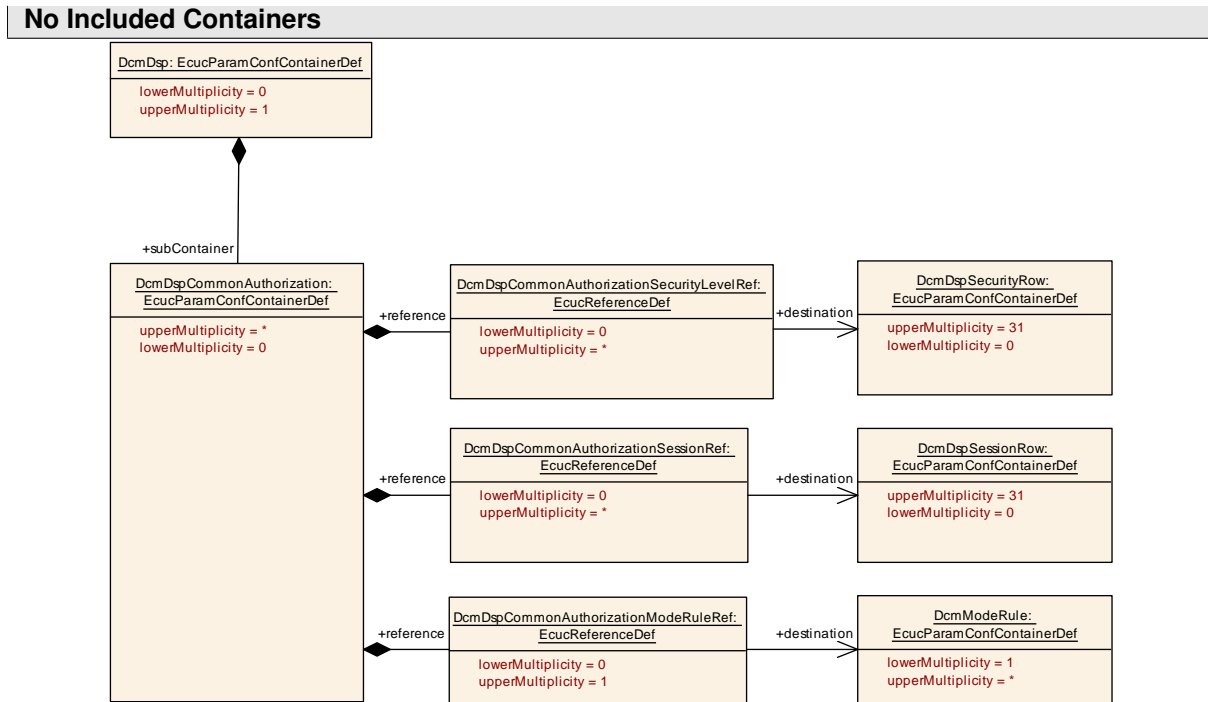


Figure 10.20: DcmDspCommonAuthorization configuration overview

10.2.5.7 DIDs

10.2.5.7.1 DcmDspDid

SWS Item	[ECUC_Dcm_00601]
Container Name	DcmDspDid
Parent Container	DcmDsp
Description	This container contains the configuration (parameters) of the DID.
Configuration Parameters	

Name	DcmDspDidIdentifier [ECUC_Dcm_00602]	
Parent Container	DcmDspDid	
Description	2 byte Identifier of the DID Within each DcmConfigSet all DcmDspDidIdentifier values shall be unique.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 65535	
Default Value		
Post-Build Variant Value	false	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidSize [ECUC_Dcm_01099]		
Parent Container	DcmDspDid		
Description	Length of a DID in byte(s).		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidUsed [ECUC_Dcm_00805]		
Parent Container	DcmDspDid		
Description	Allow to activate or deactivate the usage of a DID, for multi purpose ECUs true = DID available false = DID not available		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	DcmDspDidUsePort [ECUC_Dcm_01122]		
Parent Container	DcmDspDid		
Description	Selects application interface type for DID data elements between a single operation for all data elements or data element specific operations.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	USE_ATOMIC_BNDM	The DID shall be read/written from/to BndM.	

Default Value	USE_ATOMIC_NV_DATA_INTERFACE	A single sender receiver interface with NvData-Ports is used for all data elements of this DID.	
	USE_ATOMIC_SENDER_RECEIVER_INTERFACE	A single sender receiver interface is used to access all data elements of this DID. The sender receiver interface is typed with IsService=false.	
	USE_ATOMIC_SENDER_RECEIVER_INTERFACE_AS_SERVICE	A single sender receiver interface is used to access all data elements of this DID. The sender receiver interface is typed with IsService=true.	
	USE_DATA_ELEMENT_SPECIFIC_INTERFACES	The data elements of this DID are collected by using the data element specific interfaces defined by DcmDspDataUsePort.	
	USE_DATA_ELEMENT_SPECIFIC_INTERFACES		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

Name	DcmDspDidBndMBlockIdRef [ECUC_Dcm_01185]		
Parent Container	DcmDspDid		
Description	Associate this DID with a BndM blockId. Tags: atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to BndMBlockDescriptor		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU dependency: DcmDspDidUsePort == USE_ATOMIC_BNDM		

Name	DcmDspDidInfoRef [ECUC_Dcm_00604]		
Parent Container	DcmDspDid		
Description	Reference to DcmDspDidInfo containing information on this DID.		
Multiplicity	1		
Type	Reference to DcmDspDidInfo		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidRef [ECUC_Dcm_00606]		
Parent Container	DcmDspDid		
Description	Reference to DcmDspDid in case this DID refer to one or several other DID's Attributes: requiresIndex=true		
Multiplicity	0..*		
Type	Reference to DcmDspDid		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspDidSignal	0..*	This container defines the reference to 1 DcmDspData container and position relevant for this DID.
DcmDspDidSupportInfo	0..1	This container defines the support information to declare the usability of the data bytes within the DIDs

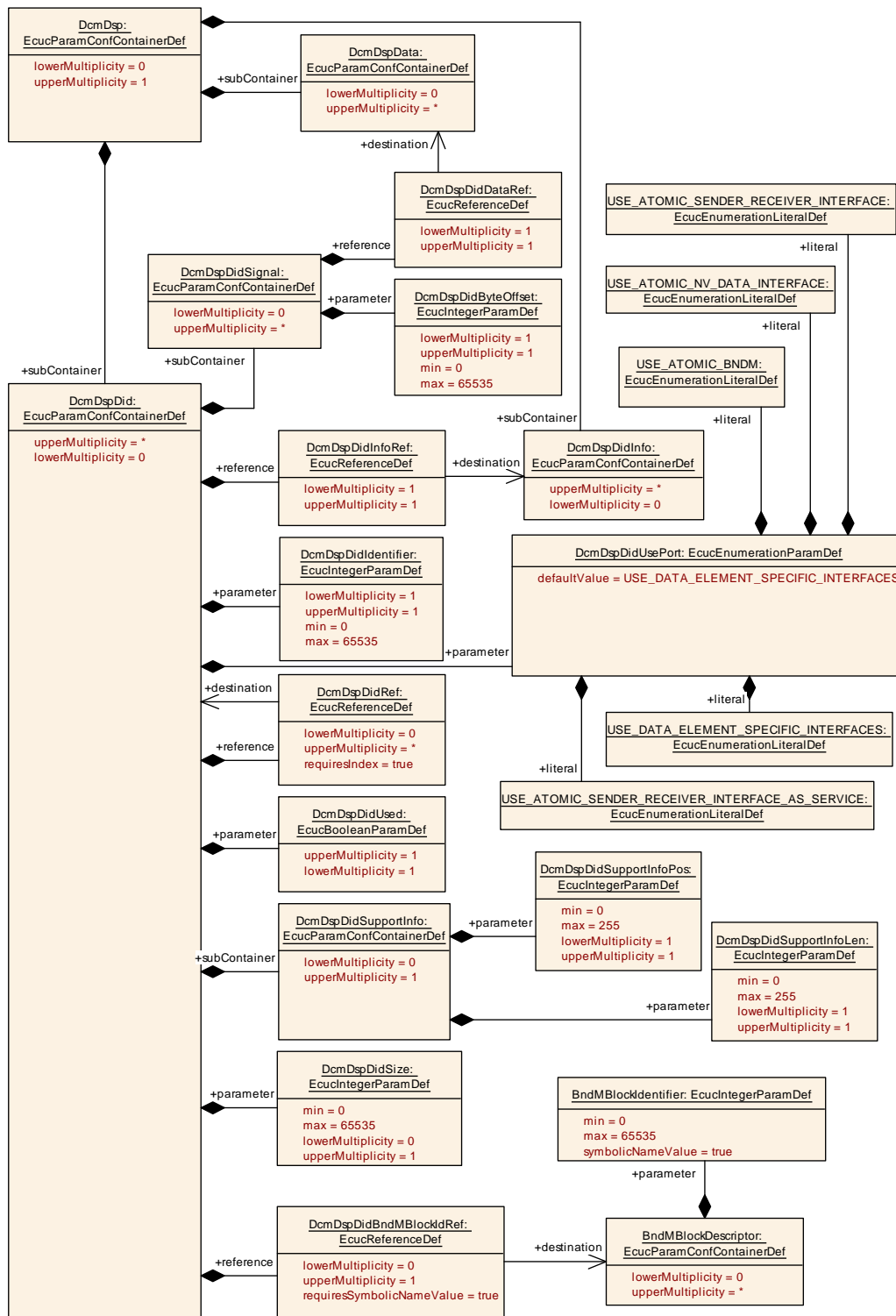


Figure 10.21: DcmDspDid configuration overview

10.2.5.7.2 DcmDspDidInfo

SWS Item	[ECUC_Dcm_00607]
----------	------------------

Container Name	DcmDspDidInfo
Parent Container	DcmDsp
Description	This container contains the configuration (parameters) of the DID's Info
Configuration Parameters	

Name	DcmDspDDDIDMaxElements [ECUC_Dcm_00970]		
Parent Container	DcmDspDidInfo		
Description	Maximum number of source elements of a DDDID.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspDidDynamicallyDefined [ECUC_Dcm_00612]		
Parent Container	DcmDspDidInfo		
Description	Indicates if this DID can be dynamically defined true = DID can be dynamically defined false = DID can not be dynamically defined		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspDidControl	0..1	This container contains the configuration (parameters) of the DID control.
DcmDspDidRead	0..1	This container contains the configuration (parameters) of the DID read.

DcmDspDidWrite	0..1	This container contains the configuration (parameters) of the DID write.
--------------------------------	------	--

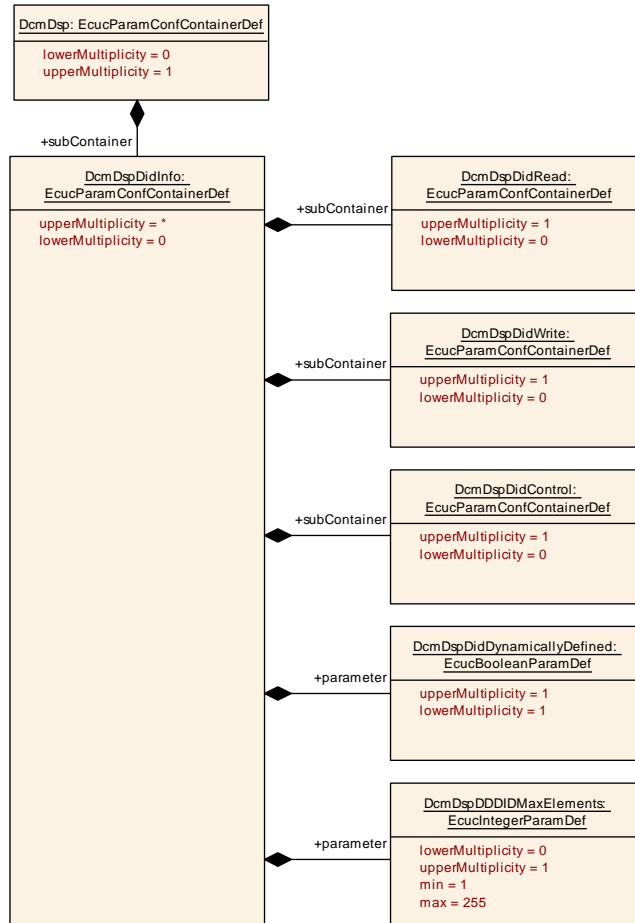


Figure 10.22: DcmDspDidInfo configuration overview

10.2.5.7.3 DcmDspDidRead

SWS Item	[ECUC_Dcm_00613]
Container Name	DcmDspDidRead
Parent Container	DcmDspDidInfo
Description	This container contains the configuration (parameters) of the DID read.
Configuration Parameters	

Name	DcmDspDidReadRole [ECUC_Dcm_01141]		
Parent Container	DcmDspDidRead		
Description	<p>Bitfield where each bit represents one dedicated role. A DID is granted read access if the bit value is 1. If a bit value is 0, the DID is not allowed to be read for that role.</p> <p>Reading DIDs is possible by different UDS services. This setting applies to all possible DID read operations.</p>		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	DcmDspDidReadModeRuleRef [ECUC_Dcm_00917]		
Parent Container	DcmDspDidRead		
Description	<p>Reference to DcmModeRule</p> <p>Mode rule which controls to read this DID. If there is no reference, no check of the mode rule shall be done.</p>		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidReadSecurityLevelRef [ECUC_Dcm_00614]		
Parent Container	DcmDspDidRead		
Description	Reference to DcmDspSecurityRow Referenced security levels are allowed to read this DID. If there is no reference, no check of security level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSecurityRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidReadSessionRef [ECUC_Dcm_00615]		
Parent Container	DcmDspDidRead		
Description	Reference to DcmDspSessionRow Referenced sessions are allowed to read this DID. If there is no reference, no check of session level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSessionRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

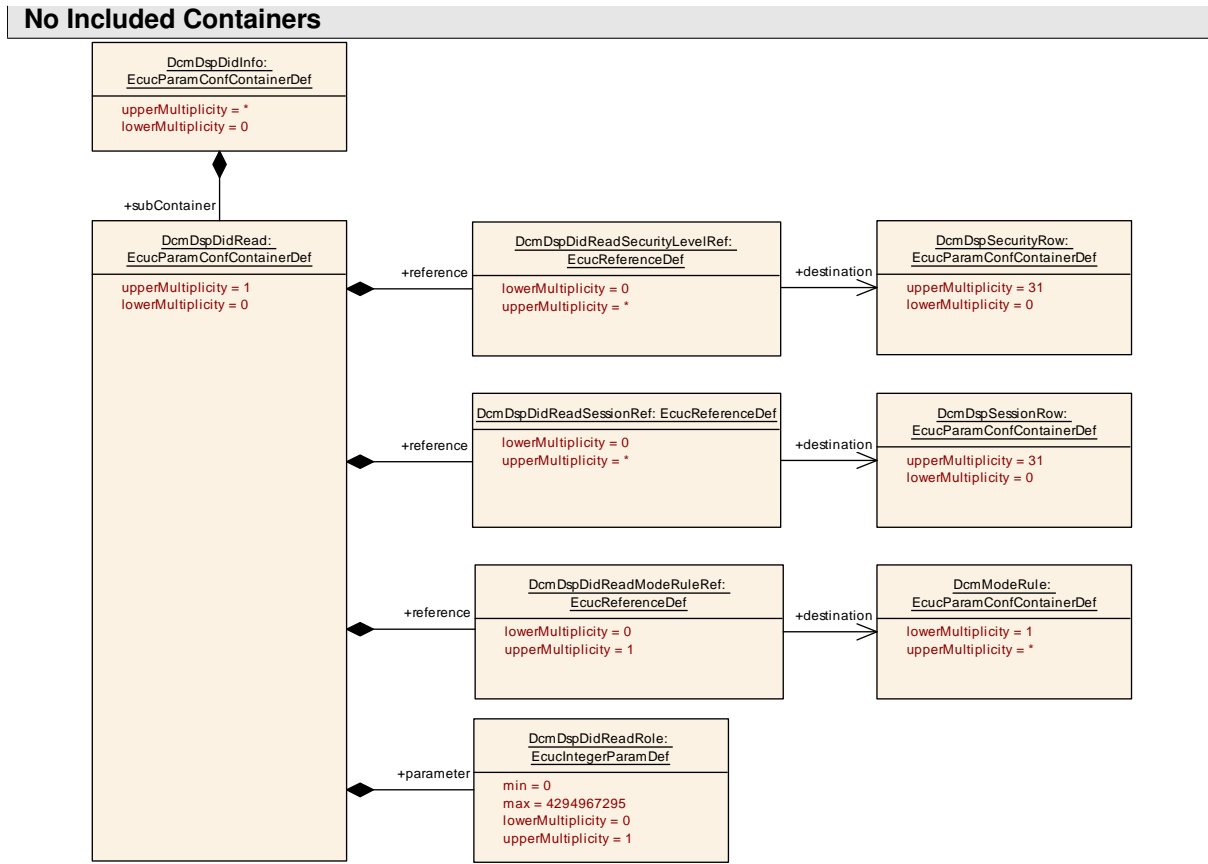


Figure 10.23: DcmDspDidRead configuration overview

10.2.5.7.4 DcmDspDidSignal

SWS Item	[ECUC_Dcm_00813]
Container Name	DcmDspDidSignal
Parent Container	DcmDspDid
Description	This container defines the reference to 1 DcmDspData container and position relevant for this DID.
Configuration Parameters	

Name	DcmDspDidByteOffset [ECUC_Dcm_01105]	
Parent Container	DcmDspDidSignal	
Description	Defines the absolute byte offset of the data defined by DcmDspDidDataRef reference to DcmDspData container in the DID.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 65535	
Default Value		
Post-Build Variant Value	false	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidDataRef [ECUC_Dcm_00808]		
Parent Container	DcmDspDidSignal		
Description	Reference to 1 DcmDspData container relevant for this DID.		
Multiplicity	1		
Type	Reference to DcmDspData		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.7.5 DcmDspDidSupportInfo

SWS Item	[ECUC_Dcm_01102]
Container Name	DcmDspDidSupportInfo
Parent Container	DcmDspDid
Description	This container defines the support information to declare the usability of the data bytes within the DIDs
Configuration Parameters	

Name	DcmDspDidSupportInfoLen [ECUC_Dcm_01103]		
Parent Container	DcmDspDidSupportInfo		
Description	Length of the support information in bytes.		
Multiplicity	1		
Type	EcuIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidSupportInfoPos [ECUC_Dcm_01100]		
Parent Container	DcmDspDidSupportInfo		
Description	Length of the support information in bytes.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.7.6 DcmDspDidRange

SWS Item	[ECUC_Dcm_00937]
Container Name	DcmDspDidRange
Parent Container	DcmDsp
Description	This container defines the DID Range
Configuration Parameters	

Name	DcmDspDidRangeHasGaps [ECUC_Dcm_00941]		
Parent Container	DcmDspDidRange		
Description	Parameter specifying if there are gaps in the DID range (parameter set to TRUE) or not (parameter set to FALSE)		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidRangeIdentifierLowerLimit [ECUC_Dcm_00938]		
Parent Container	DcmDspDidRange		
Description	Lower limit of DID range.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		

Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidRangeIdentifierUpperLimit [ECUC_Dcm_00939]		
Parent Container	DcmDspDidRange		
Description	Upper limit of DID range.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidRangeIsDidAvailableFnc [ECUC_Dcm_00946]		
Parent Container	DcmDspDidRange		
Description	Function name to request from application if a specific DID is available within the range or not. Only relevant if DcmDspDidRangeUsePort is set to false. This parameter is related to the interface Xxx_IsDidAvailable.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidRangeMaxDataLength [ECUC_Dcm_00940]		
Parent Container	DcmDspDidRange		
Description	Maximum data length in bytes		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidRangeReadDataLengthFnc [ECUC_Dcm_01067]		
Parent Container	DcmDspDidRange		
Description	Function name to request from application the length of the data of a range DID. Only relevant if DcmDspDidRangeUsePort is set to false. This parameter is related to the interface Xxx_ReadDidRangeDataLength.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidRangeReadDidFnc [ECUC_Dcm_00947]		
Parent Container	DcmDspDidRange		
Description	Function name to request from application the data range value of a DID.(ReadData-function). Only relevant if DcmDspDidRangeUsePort is set to false. This parameter is related to the interface Xxx_ReadDidData.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			

Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidRangeUsePort [ECUC_Dcm_00945]		
Parent Container	DcmDspDidRange		
Description	When the parameter DcmDspDidRangeUsePort is set to true the DCM will access the Data using an R-Port requiring a PortInterface DataServices_DIDRange. In that case, DcmDspDidRangeIsDidAvailableFnc, DcmDspDidRangeReadDidFnc and DcmDspDidRangeWriteDidFnc are ignored and the RTE APIs are used. When the parameter DcmDspDidRangeUsePort is false, the DCM calls the functions defined in DcmDspDidRangeIsDidAvailableFnc, DcmDspDidRangeReadDidFnc and DcmDspDidRangeWriteDidFnc.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidRangeWriteDidFnc [ECUC_Dcm_00948]		
Parent Container	DcmDspDidRange		
Description	Function name to request application to write the data range value of a DID.(WriteData-function). Only relevant if DcmDspDidRangeUsePort is set to false. This parameter is related to the interface Xxx_WriteDidData.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidRangeInfoRef [ECUC_Dcm_00950]		
Parent Container	DcmDspDidRange		
Description	Reference to DcmDspDidInfo containing information on this DID Range.		
Multiplicity	1		
Type	Reference to DcmDspDidInfo		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

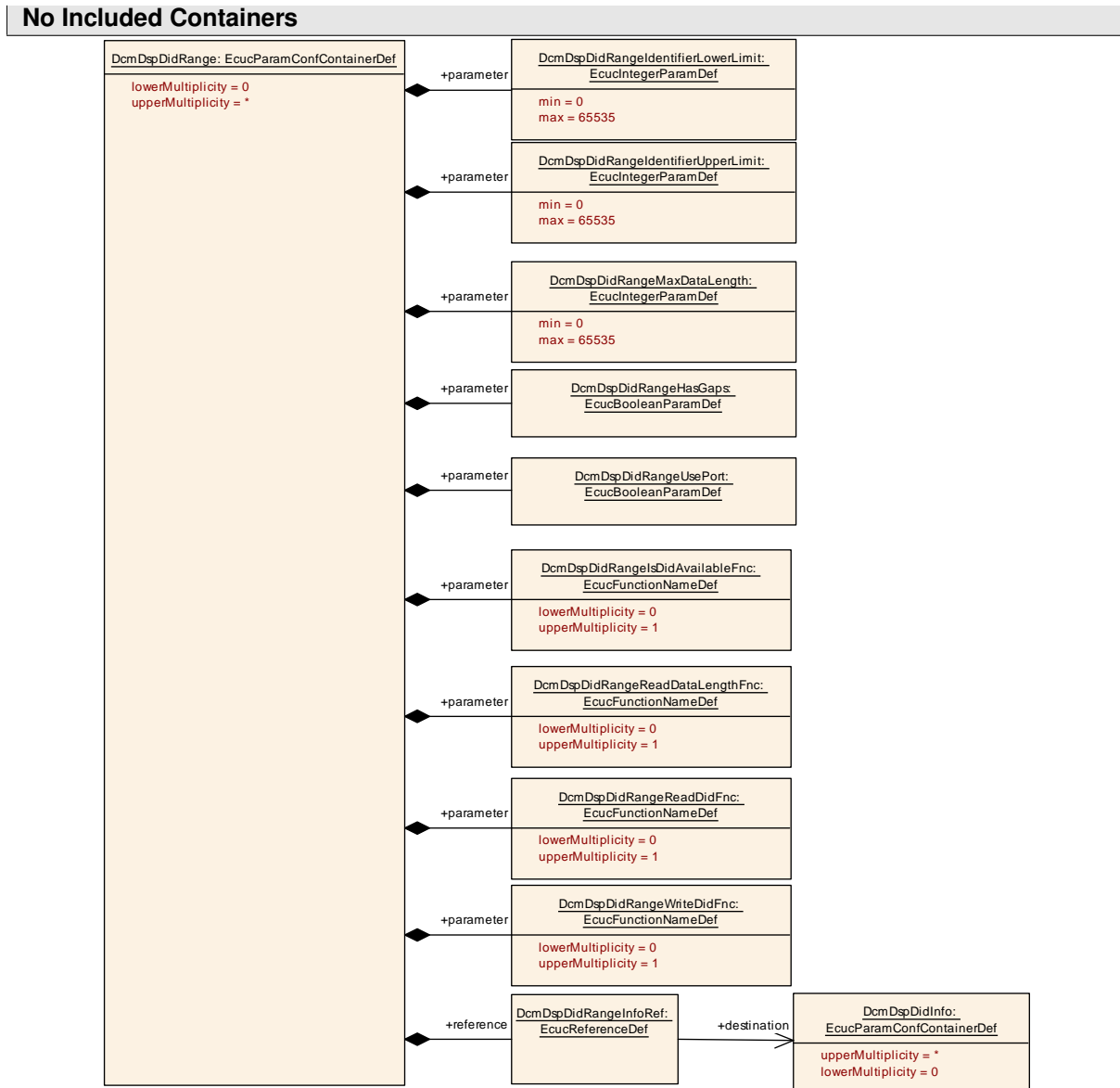


Figure 10.24: DcmDspDidRange configuration overview

10.2.5.7.7 DcmDspDidWrite

SWS Item	[ECUC_Dcm_00616]
Container Name	DcmDspDidWrite
Parent Container	DcmDspDidInfo
Description	This container contains the configuration (parameters) of the DID write.
Configuration Parameters	

Name	DcmDspDidWriteRole [ECUC_Dcm_01142]		
Parent Container	DcmDspDidWrite		
Description	Bitfield where each bit represents one dedicated role. A DID is granted write access if the bit value is 1. If a bit value is 0, the DID is not allowed to be written for that role.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	DcmDspDidWriteModeRuleRef [ECUC_Dcm_00922]		
Parent Container	DcmDspDidWrite		
Description	Reference to DcmModeRule Mode rule which controls to write this DID. If there is no reference, no check of the mode rule shall be done.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidWriteSecurityLevelRef [ECUC_Dcm_00617]		
Parent Container	DcmDspDidWrite		
Description	Reference to DcmDspSecurityRow Referenced security levels are allowed to write this DID. If there is no reference, no check of security level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSecurityRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidWriteSessionRef [ECUC_Dcm_00618]		
Parent Container	DcmDspDidWrite		
Description	Reference to DcmDspSessionRow Referenced sessions are allowed to write this DID. If there is no reference, no check of session level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSessionRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

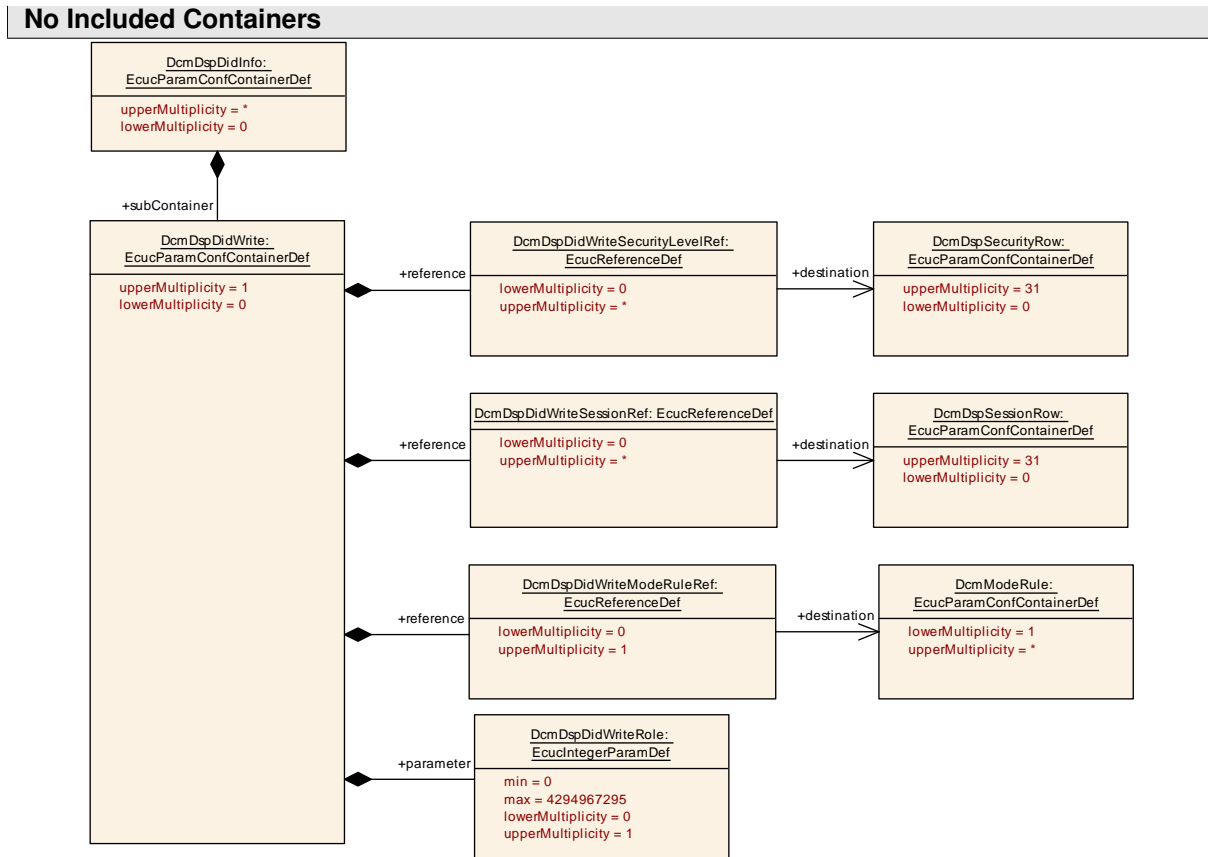


Figure 10.25: DcmDspDidWrite configuration overview

10.2.5.8 DcmDspControlDTCSetting

SWS Item	[ECUC_Dcm_00935]
Container Name	DcmDspControlDTCSetting
Parent Container	DcmDsp
Description	Provide the configuration of the ControlDTCSetting mechanism.
Configuration Parameters	

Name	DcmSupportDTCSettingControlOptionRecord [ECUC_Dcm_00965]
Parent Container	DcmDspControlDTCSetting
Description	This configuration switch defines if the DTCSettingControlOptionRecord is in general supported in the request message or not.
Multiplicity	0..1
Type	EcucBooleanParamDef
Default Value	false
Post-Build Variant Multiplicity	false
Post-Build Variant Value	false

Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspControlDTCSettingReEnableModeRuleRef [ECUC_Dcm_00936]		
Parent Container	DcmDspControlDTCSetting		
Description	Reference to DcmModeRule Mode rule which controls re-enabling of controlDTCsetting by DCM. The DCM module shall execute a ControlDTCSetting.Off (call Dem_EnabledDTCSetting()) in case that the referenced mode rule is not fulfilled anymore.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

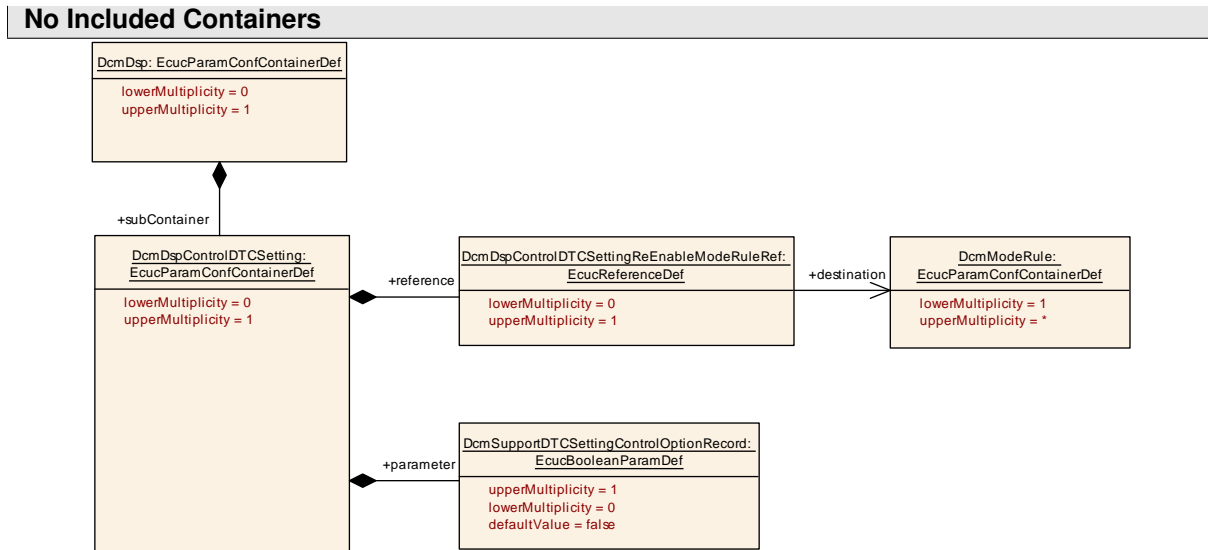


Figure 10.26: DcmDspControlDTCSetting configuration overview

10.2.5.9 Data elements

10.2.5.9.1 DcmDspData

SWS Item	[ECUC_Dcm_00869]
Container Name	DcmDspData
Parent Container	DcmDsp
Description	This container contains the configuration (parameters) of a Data belonging to a DID
Configuration Parameters	

Name	DcmDspDataByteSize [ECUC_Dcm_01106]		
Parent Container	DcmDspData		
Description	Defines the array length in bytes or the the maximum array length for variable datalengths.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataConditionCheckReadFnc [ECUC_Dcm_00677]		
Parent Container	DcmDspData		
Description	Function name to demand application if the conditions (e.g. System state) to read the DID are correct. (ConditionCheckRead-function). Multiplicity shall be equal to parameter DcmDspDataReadFnc. This parameter is related to the interface Xxx_ConditionCheckRead.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataConditionCheckReadFncUsed [ECUC_Dcm_00955]		
Parent Container	DcmDspData		
Description	This parameter determines if a condition check function is available or not.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspDataEcuSignal [ECUC_Dcm_00825]		
Parent Container	DcmDspData		
Description	Function name to control the access to a certain ECU Signal by the DCM. (IoHwAb_Dcm_<symbolic name of ECU signal>-function).		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataEndianness [ECUC_Dcm_00986]		
Parent Container	DcmDspData		
Description	Defines the endianness of the data belonging to a DID in a diagnostic request or response message.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	BIG_ENDIAN	Most significant byte shall be stored at the lowest address.	
	LITTLE_ENDIAN	Most significant byte shall be stored at the highest address.	
	OPAQUE	Opaque data endianness	
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspDataFreezeCurrentStateFnc [ECUC_Dcm_00674]		
Parent Container	DcmDspData		
Description	Function name to request to application to freeze the current state of an IOControl. (FreezeCurrentState-function). This parameter is related to the interface Xxx_FreezeCurrentState.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataGetScalingInfoFnc [ECUC_Dcm_00676]		
Parent Container	DcmDspData		
Description	Function name to request to application the scaling information of the DID. (GetScalingInformation-function). This parameter is related to the interface Xxxx_GetScalingInformation.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataReadDataLengthFnc [ECUC_Dcm_00671]		
Parent Container	DcmDspData		
Description	Function name to request from application the data length of a DID. (ReadDataLength-function). This parameter is related to the interface Xxx_ReadDataLength.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataReadEcuSignal [ECUC_Dcm_00824]		
Parent Container	DcmDspData		
Description	Function name for read access to a certain ECU Signal by the DCM. (IoHwAb_Dcm_Read<EcuSignalName>-function). Only relevant if DcmDspDataUsePort==USE_ECU_SIGNAL.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataReadFnc [ECUC_Dcm_00669]		
Parent Container	DcmDspData		
Description	Function name to request from application the data value of a DID. (ReadData-function). This parameter is related to the interface Xxx_ReadData.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspDataResetToDefaultFnc [ECUC_Dcm_00673]		
Parent Container	DcmDspData		
Description	Function name to request to application to reset an IOControl to default value. (ResetToDefault-function). This parameter is related to the interface Xxx_ResetToDefault.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataReturnControlToEcuFnc [ECUC_Dcm_00672]		
Parent Container	DcmDspData		
Description	Function name to request to application to return control to ECU of an IOControl. (ReturnControlToECU-function). This parameter is related to the interface Xxx_ReturnControlToECU.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataShortTermAdjustmentFnc [ECUC_Dcm_00675]		
Parent Container	DcmDspData		
Description	Function name to request to application to adjust the IO signal. (ShortTermAdjustment-function). This parameter is related to the interface Xxx_ShortTermAdjustment.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataType [ECUC_Dcm_00985]		
Parent Container	DcmDspData		
Description	Provide the implementation data type of data belonging to a DID.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BOOLEAN	Type of the data is boolean.	
	FLOAT	Type of the data is float.	
	FLOAT_N	Type of the data is float array.	
	SINT16	Type of the data is sint16.	
	SINT16_N	Type of the data is sint16 array.	
	SINT32	Type of the data is sint32.	
	SINT32_N	Type of the data is sint32 array.	
	SINT8	Type of the data is sint8.	
	SINT8_N	Type of the data is sint8 array.	
	UINT16	Type of the data is uint16.	
	UINT16_N	Type of the data is uint16 array.	
	UINT32	Type of the data is uint32.	
	UINT32_N	Type of the data is uint32 array.	
	UINT8	Type of the data is uint8.	
	UINT8_DYN	Type of the data is uint8 array with dynamic length.	
	Post-Build Variant Value	false	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataUsePort [ECUC_Dcm_00713]		
Parent Container	DcmDspData		
Description	Defines which interface shall be used to access the data.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	USE_BLOCK_ID	The DCM will access the Data using the NVRAM Apis with the BlockId defined in DcmDspDataBlockId	

USE_DATA_ASYNC_CLIENT_SERVER	The DCM will access the Data using an R-Port requiring a asynchronous ClientServerInterface DataServices_{Data}. The R-Port is named DataServices_{Data} where {Data} is the name of the container DcmDspData.
USE_DATA_ASYNC_CLIENT_SERVER_ERROR	The Dcm will access the Data using an R-Port requiring a asynchronous ClientServerInterface DataServices_{Data}. The parameter ErrorCode can be returned to allow the application to trigger a negative response during the operation. The R-Port is named DataServices_{Data} where {Data} is the name of the container DcmDspData.
USE_DATA_ASYNC_FUNC	The DCM will access the Data using the functions that are defined in parameters of type EcucFunctionNameDef (but without DcmDspDataReadDataLengthFnc) in the DcmDspData container. DCM_E_PENDING return is allowed. OpStatus is existing as IN parameter.
USE_DATA_ASYNC_FUNC_ERROR	The DCM will access the Data using the functions that are defined in parameters of type EcucFunctionNameDef (but without DcmDspDataReadDataLengthFnc) in the DcmDspData container. DCM_E_PENDING return is allowed. OpStatus is existing as IN parameter. The parameter ErrorCode can be returned to allow the application to trigger a negative response during the operation.
USE_DATA_SENDER_RECEIVER	The DCM will access the Data using an Port requiring a SenderReceiverInterface (with isService=false) DataServices_{Data}. The Port is namedDataServices_{Data} where {Data} is the name of the container DcmDspData.
USE_DATA_SENDER_RECEIVER_AS_SERVICE	The DCM will access the Data using an service Port requiring a SenderReceiverInterface (with isService=true) DataServices_{Data} . The Port is namedDataServices_{Data} where {Data} is the name of the container DcmDspData.

Post-Build Variant Value	USE_DATA_SYNCH_CLIENT_SERVER	The DCM will access the Data using an R-Port requiring a synchronous ClientServerInterface DataServices_{Data}. The R-Port is named DataServices_{Data} where {Data} is the name of the container DcmDspData.	
	USE_DATA_SYNCH_FNC	The DCM will access the Data using the functions that are defined in parameters of type EcucFunctionNameDef (but without DcmDspDataReadDataLengthFnc) in the DcmDspData container. DCM_E_PENDING return value is not allowed and OpStatus parameter is not existing in the prototype.	
	USE_ECU_SIGNAL	The DCM will access the Data using a direct access to IoHwAb	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataWriteFnc [ECUC_Dcm_00670]		
Parent Container	DcmDspData		
Description	Function name to request application to write the data value of a DID. (WriteData-function). This parameter is related to the interface Xxx_WriteData.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspOdxDataDescription [ECUC_Dcm_00988]		
Parent Container	DcmDspData		
Description	Defines additional description for ODX documentation		
Multiplicity	0..1		
Type	EcucAddInfoParamDef		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspDataBlockIdRef [ECUC_Dcm_00809]		
Parent Container	DcmDspData		
Description	NRAM blockId to access the data. Only relevant if DcmDspDataUsePort==USE_BLOCK_ID.		
Multiplicity	0..1		
Type	Symbolic name reference to NvMBlockDescriptor		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDataInfoRef [ECUC_Dcm_00811]		
Parent Container	DcmDspData		
Description	Reference to 1 DcmDspDataInfo		
Multiplicity	0..1		
Type	Reference to DcmDspDataInfo		
Post-Build Variant Value	false		

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspDiagnosisScaling	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation. Out if this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.
DcmDspDidDataSupportInfo	0..1	This container defines the supported information.
DcmDspExternalSRDataElementClass	0..1	<p>This container defines the source of data in a provided port which shall be read respectively the target of data in a required port which shall be written.</p> <p>This container shall contain either one DcmSubElementInDataElementInstance OR DcmDataElementInstance OR DcmSubElementInImplDataElementInstance reference.</p>

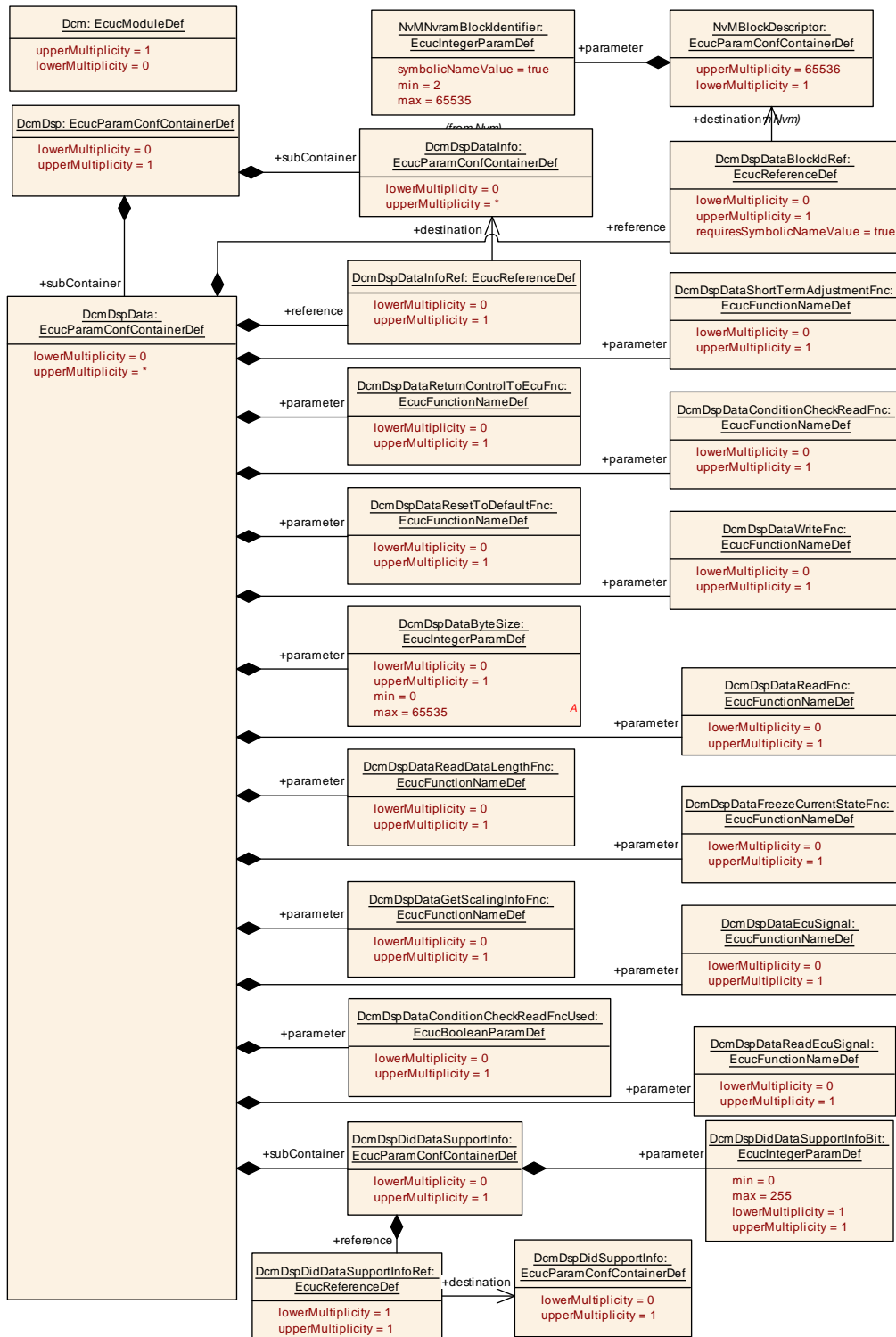


Figure 10.27: DcmDspData1 configuration overview

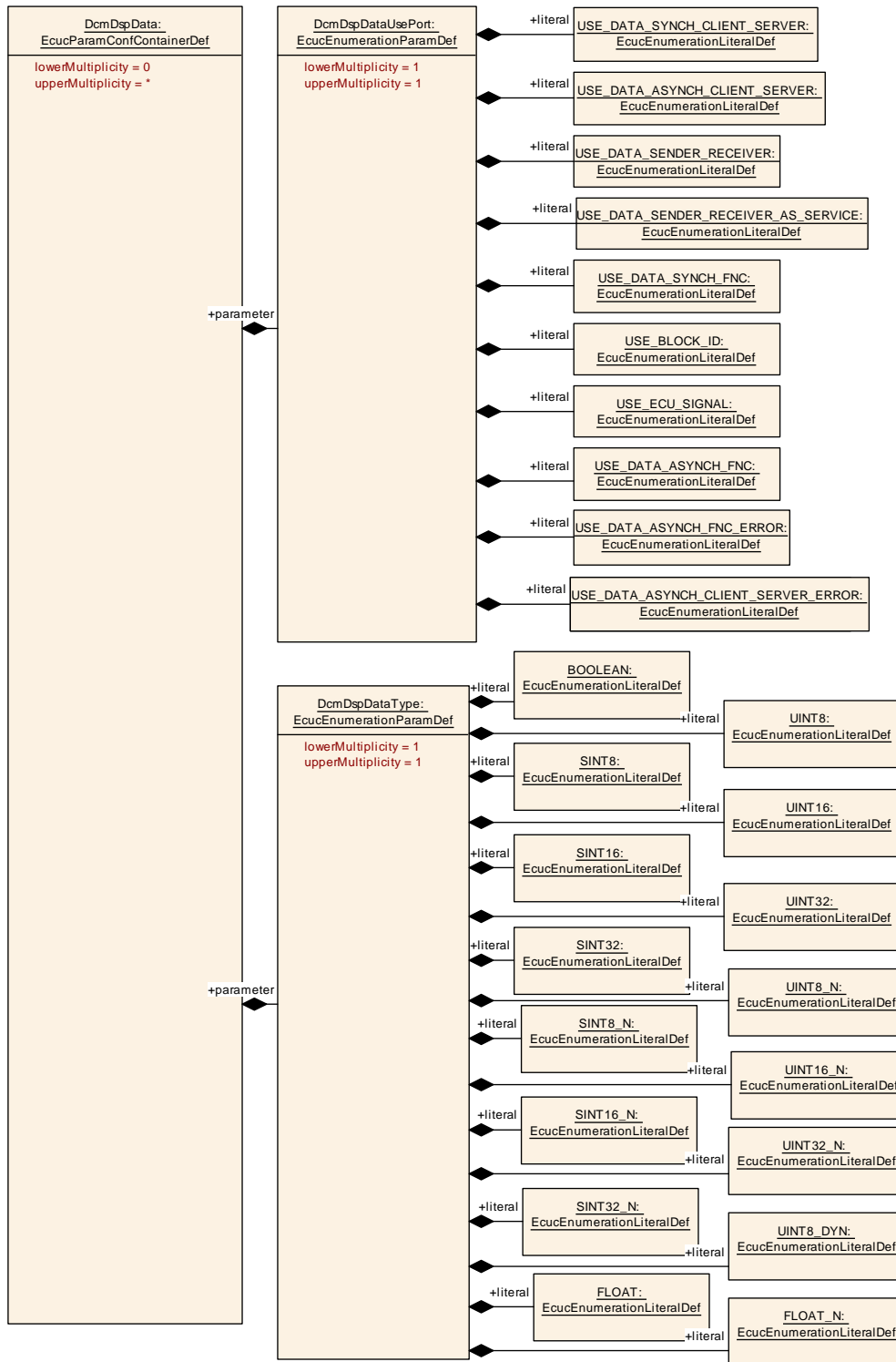


Figure 10.28: DcmDspData2 configuration overview

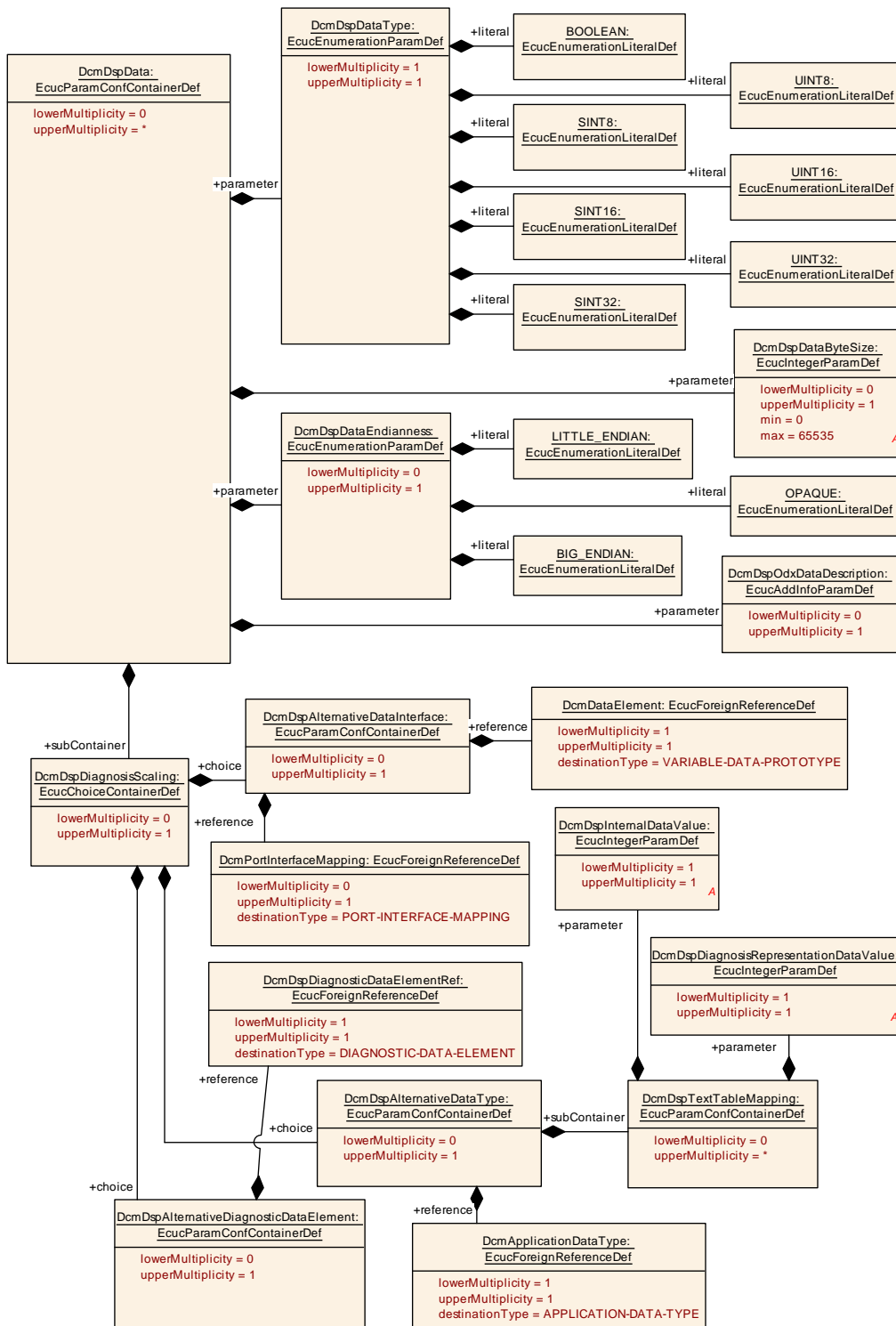


Figure 10.29: DcmDspSRData1 configuration overview

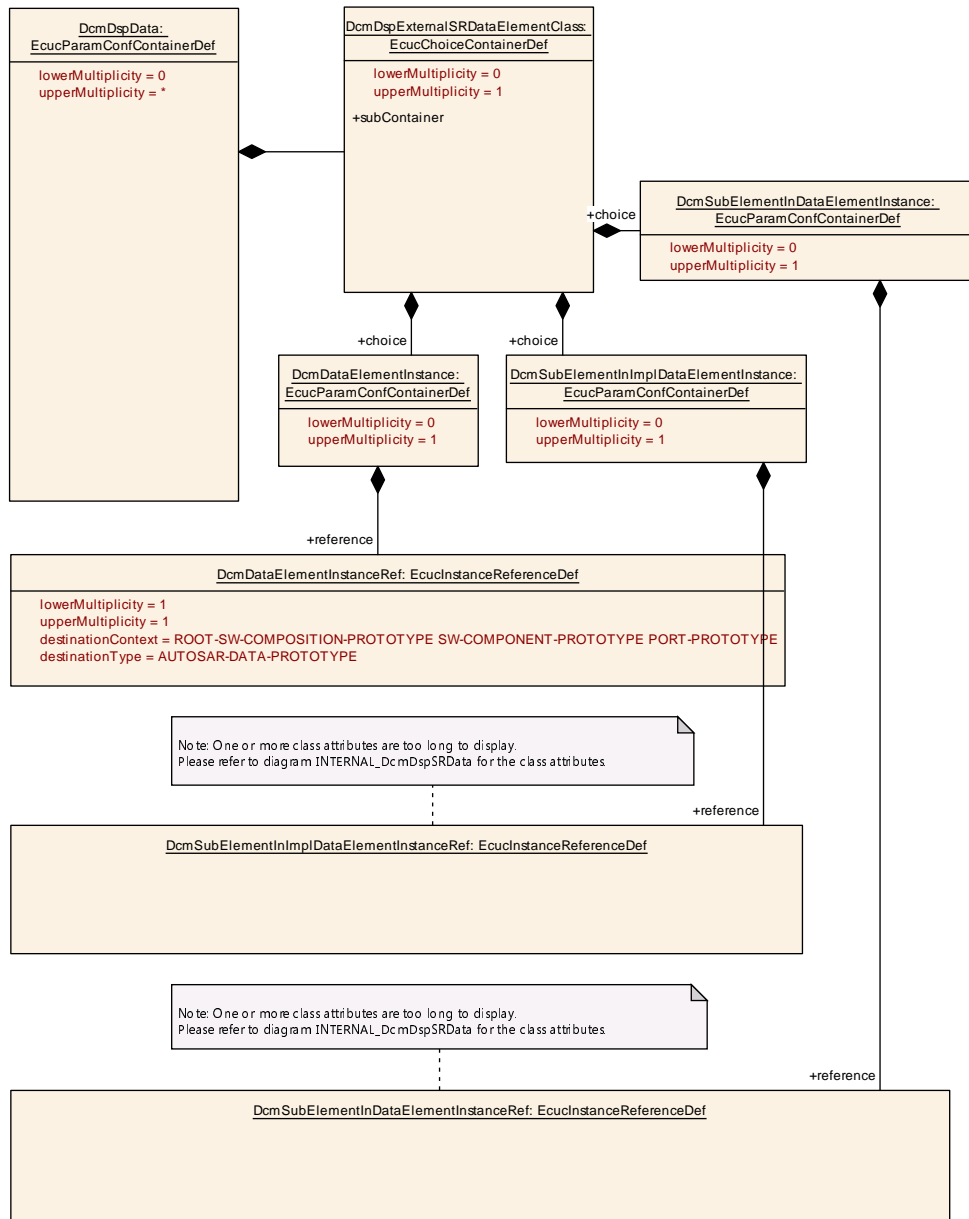


Figure 10.30: DcmDspSRData2 configuration overview

10.2.5.9.2 DcmDspDiagnosisScaling

SWS Item	[ECUC_Dcm_00993]
Container Name	DcmDspDiagnosisScaling
Parent Container	DcmDspData , DcmDspPidService01
Description	This container contains the configuration (parameters) of an alternative Diagnosis Representation. Out if this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.
Configuration Parameters	

Container Choices		
Container Name	Multiplicity	Scope / Dependency
DcmDspAlternativeDataInterface	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of a VariableDataPrototype in a DataInterface. Additionally a reference to PortInterfaceMapping can be defined which provide already the mapping rules between the VariableDataPrototype in a DataInterface used by the software component (DcmDspExternalSRDataElementClass) and the intended Diagnosis Representation defined by DcmDataElement.
DcmDspAlternativeDataType	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of an ApplicationDataType. Additionally the definition of a text table mapping can be defined for ApplicationDataTypes with a CompuMethod of category TEXTTABLE and SCALE_LINEAR_AND_TEXTTABLE.
DcmDspAlternativeDiagnosticDataElement	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of Diagnostic Extract.

10.2.5.9.3 DcmDspArgumentScaling

SWS Item	[ECUC_Dcm_01062]
Container Name	DcmDspArgumentScaling
Parent Container	DcmDspRequestRoutineResultsInSignal , DcmDspRequestRoutineResultsOutSignal , DcmDspStartRoutineInSignal , DcmDspStartRoutineOutSignal , DcmDspStopRoutineInSignal , DcmDspStopRoutineOutSignal
Description	This container contains the configuration (arguments) of an alternative Diagnosis Representation. Out if this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.
Configuration Parameters	

Container Choices		
Container Name	Multiplicity	Scope / Dependency
DcmDspAlternativeArgumentData	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of a ArgumentDataPrototype.
DcmDspAlternativeDataType	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of an ApplicationDataType. Additionally the definition of a text table mapping can be defined for ApplicationDataTypes with a CompuMethod of category TEXTTABLE and SCALE_LINEAR_AND_TEXTTABLE.

DcmDspAlternativeDiagnosticDataElement	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of Diagnostic Extract.
--	------	---

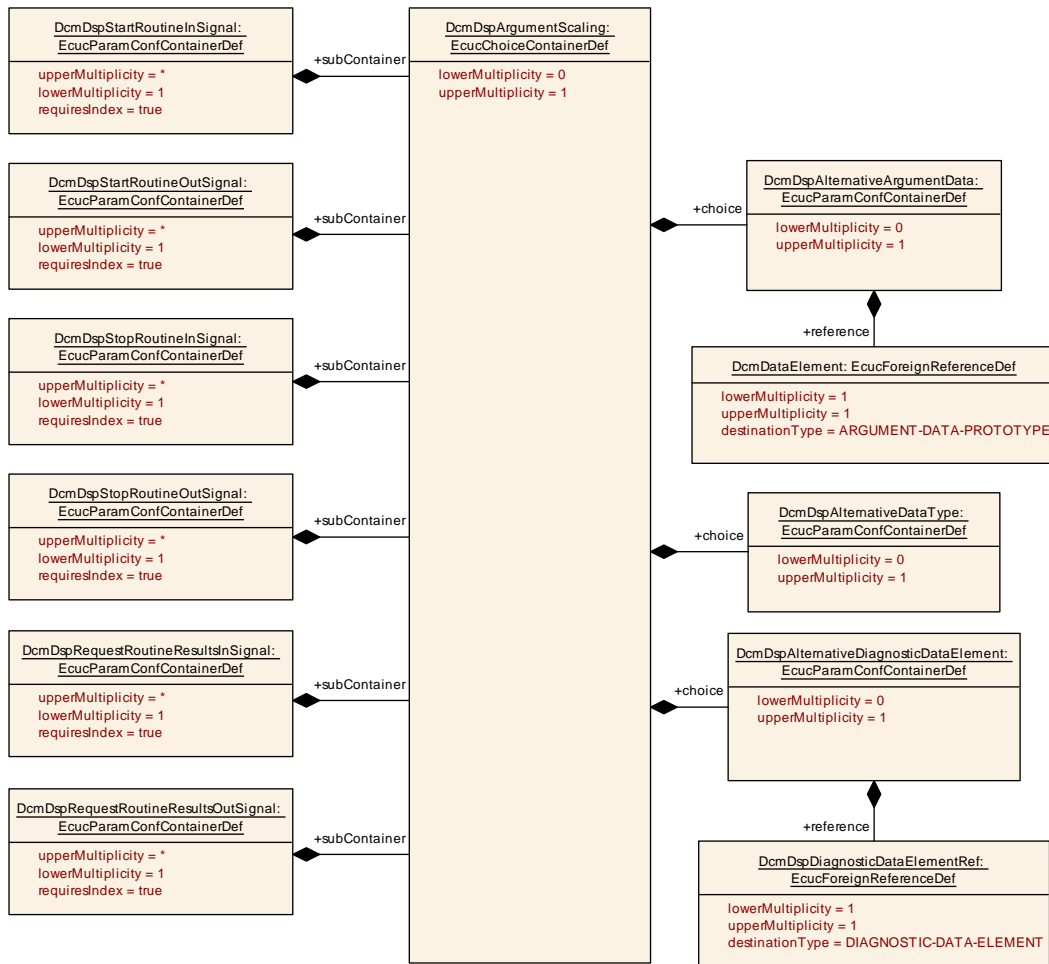


Figure 10.31: DcmDspArgumentScaling configuration overview

10.2.5.9.4 DcmDspAlternativeArgumentData

SWS Item	[ECUC_Dcm_01055]
Container Name	DcmDspAlternativeArgumentData
Parent Container	DcmDspArgumentScaling , DcmDspArgumentScaling , DcmDspArgumentScaling , DcmDspArgumentScaling , DcmDspArgumentScaling
Description	This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of a ArgumentDataPrototype.
Configuration Parameters	

Name	DcmDataElement [ECUC_Dcm_01056]		
Parent Container	DcmDspAlternativeArgumentData		
Description	<p>Alternative Diagnosis Representation for the data defined by the means of a ArgumentDataPrototype.</p> <p>The CompuMethod of the data type of the referenced ArgumentDataPrototype will be applied to the data type of the ArgumentDataPrototype in the interface used by the Dcm.</p>		
Multiplicity	1		
Type	Foreign reference to ARGUMENT-DATA-PROTOTYPE		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

No Included Containers

10.2.5.9.5 DcmDspTextTableMapping

SWS Item	[ECUC_Dcm_00999]
Container Name	DcmDspTextTableMapping
Parent Container	DcmDspAlternativeDataType , DcmDspAlternativeDataType , DcmDspAlternativeDataType , DcmDspAlternativeDataType , DcmDspAlternativeDataType , DcmDspAlternativeDataType , DcmDspAlternativeDataType , DcmDspAlternativeDataType
Description	<p>The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists).</p> <p>In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.</p>
Configuration Parameters	

Name	DcmDspDiagnosisRepresentationDataValue [ECUC_Dcm_01001]		
Parent Container	DcmDspTextTableMapping		
Description	The data value in the diagnosis representation.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspInternalDataValue [ECUC_Dcm_01000]		
Parent Container	DcmDspTextTableMapping		
Description	The ECU internal data value.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.9.6 DcmDspAlternativeDataInterface

SWS Item	[ECUC_Dcm_00994]
Container Name	DcmDspAlternativeDataInterface
Parent Container	DcmDspDiagnosisScaling , DcmDspDiagnosisScaling
Description	<p>This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of a VariableDataPrototype in a DataInterface.</p> <p>Additionally a reference to PortInterfaceMapping can be defined which provide already the mapping rules between the VariableDataPrototype in a DataInterface used by the software component (DcmDspExternalSRDataElementClass) and the intended Diagnosis Representation defined by DcmDataElement.</p>

Configuration Parameters

Name	DcmDataElement [ECUC_Dcm_00995]		
Parent Container	DcmDspAlternativeDataInterface		
Description	<p>Alternative Diagnosis Representation for the data defined by the means of a VariableDataPrototype in a DataInterface.</p> <p>The CompuMethod of the data type of the referenced VariableDataPrototype will be applied to the data type of the VariableDataPrototype in the interface used by the Dcm.</p>		
Multiplicity	1		
Type	Foreign reference to VARIABLE-DATA-PROTOTYPE		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency			

Name	DcmPortInterfaceMapping [ECUC_Dcm_00996]		
Parent Container	DcmDspAlternativeDataInterface		
Description	<p>Optional reference to PortInterfaceMapping which defines the mapping rules.</p> <p>The PortInterfaceMapping is used to get the DataPrototypeMapping that describes a conversion between the data prototype referenced by DcmDataElement and the data prototype referenced from DcmDspExternalSRDataElementClass.</p>		
Multiplicity	0..1		
Type	Foreign reference to PORT-INTERFACE-MAPPING		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency			

No Included Containers

10.2.5.9.7 DcmDspAlternativeDataType

SWS Item	[ECUC_Dcm_00997]
Container Name	DcmDspAlternativeDataType
Parent Container	DcmDspArgumentScaling , DcmDspArgumentScaling , DcmDspArgumentScaling , DcmDspArgumentScaling , DcmDspArgumentScaling , DcmDspArgumentScaling , DcmDspArgumentScaling , DcmDspArgumentScaling , DcmDspDiagnosisScaling , DcmDspDiagnosisScaling
Description	<p>This container contains the configuration (parameters) of an alternative Diagnosis Representation by the means of an ApplicationDataType.</p> <p>Additionally the definition of a text table mapping can be defined for ApplicationDataTypes with a CompuMethod of category TEXTTABLE and SCALE_LINEAR_AND_TEXTTABLE.</p>
Configuration Parameters	

Name	DcmApplicationDataType [ECUC_Dcm_00998]		
Parent Container	DcmDspAlternativeDataType		
Description	<p>Alternative Diagnosis Representation for the data defined by the means of a ApplicationDataType of category VALUE, BOOLEAN or ARRAY.</p> <p>The CompuMethod that applies to the referenced ApplicationDataType in case of category VALUE or BOOLEAN will be applied to the data type of the VariableDataPrototype in the interface used by the Dcm.</p>		
Multiplicity	1		
Type	Foreign reference to APPLICATION-DATA-TYPE		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspTextTableMapping	0..*	<p>The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists).</p> <p>In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.</p>

Name	DcmDataElementInstanceRef [ECUC_Dcm_00991]		
Parent Container	DcmDataElementInstance		
Description	Instance Reference to the primitive or array data which shall be read or written. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationPrimitiveDataType of category VALUE or BOOLEAN or ApplicationArrayDataType or if the AutosarDataPrototype is typed with a ImplementationDataType of category VALUE, ARRAY or TYPE_REFERENCE that in turn boils down to VALUE or ARRAY		
Multiplicity	1		
Type	Instance reference to AUTOSAR-DATA-PROTOTYPE context: ROOT-SW-COMPOSITION-PROTOTYPE SW-COMPONENT-PROTOTYPE PORT-PROTOTYPE		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

No Included Containers

10.2.5.9.10 DcmSubElementInDataElementInstance

SWS Item	[ECUC_Dcm_01009]
Container Name	DcmSubElementInDataElementInstance
Parent Container	DcmDspExternalSRDataElementClass, DcmDspPidService01ExternalSRDataElementClass
Description	Instance Reference to the primitive sub-element (at any level) of composite data in a port where the data element is typed with an ApplicationCompositeDataType.
Configuration Parameters	

Name	DcmSubElementInDataElementInstanceRef [ECUC_Dcm_00990]		
Parent Container	DcmSubElementInDataElementInstance		
Description	<p>Instance Reference to the primitive sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationCompositeDataType.</p> <p>Note: One or more class attributes are too long to display. Please refer to diagram INTERNAL_DcmDspSRData for the class attributes.</p>		
Multiplicity	1		
Type	Instance reference to APPLICATION-COMPOSITE-ELEMENT-DATA-PROTOTYPE context: ROOT-SW-COMPOSITION-PROTOTYPE SW-COMPONENT-PROTOTYPE PORT-PROTOTYPE AUTOSAR-DATA-PROTOTYPE APPLICATION-COMPOSITE-ELEMENT-DATA-PROTOTYPE*		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

No Included Containers

10.2.5.9.11 DcmSubElementInImplDataElementInstance

SWS Item	[ECUC_Dcm_01011]
Container Name	DcmSubElementInImplDataElementInstance
Parent Container	DcmDspExternalSRDataElementClass, DcmDspPidService01ExternalSRDataElementClass
Description	Instance Reference to the primitive sub-element (at any level) of composite data in a port where the data element is typed with an ImplementationDataType.
Configuration Parameters	

Name	DcmSubElementInImplDataElementInstanceRef [ECUC_Dcm_00992]		
Parent Container	DcmSubElementInImplDataElementInstance		
Description	<p>Instance Reference to the primitive sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ImplementationDataType of category STRUCTURE or ARRAY. Please note that in case of ARRAY the index attribute in the target reference has to be set to select a single array element.</p> <p>Note: One or more class attributes are too long to display. Please refer to diagram INTERNAL_DcmDspSRData for the class attributes.</p>		
Multiplicity	1		
Type	Instance reference to IMPLEMENTATION-DATA-TYPE-ELEMENT context: ROOT-SW-COMPOSITION-PROTOTYPE SW-COMPONENT-PROTOTYPE PORT-PROTOTYPE AUTOSAR-DATA-PROTOTYPE IMPLEMENTATION-DATA-TYPE-ELEMENT*		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

No Included Containers

10.2.5.9.12 DcmDspDidDataSupportInfo

SWS Item	[ECUC_Dcm_01104]
Container Name	DcmDspDidDataSupportInfo
Parent Container	DcmDspData
Description	This container defines the supported information.
Configuration Parameters	

Name	DcmDspDidDataSupportInfoBit [ECUC_Dcm_01097]	
Parent Container	DcmDspDidDataSupportInfo	
Description	Referenced Bit of the SupportInfo	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default Value		
Post-Build Variant Value	false	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidDataSupportInfoRef [ECUC_Dcm_01098]		
Parent Container	DcmDspDidDataSupportInfo		
Description	Reference to DcmDspDidSupportInfo		
Multiplicity	1		
Type	Reference to DcmDspDidSupportInfo		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.9.13 DcmDspDataInfo

SWS Item	[ECUC_Dcm_00810]
Container Name	DcmDspDataInfo
Parent Container	DcmDsp
Description	This container contains the configuration (parameters) of one Data.
Configuration Parameters	

Name	DcmDspDataScalingInfoSize [ECUC_Dcm_00611]	
Parent Container	DcmDspDataInfo	
Description	If Scaling information service is available for this Data, it provides the size in bytes of the scaling information.	
Multiplicity	0..1	
Type	EcuIntegerParamDef	
Range	0 .. 4294967295	
Default Value		
Post-Build Variant Multiplicity	false	

Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

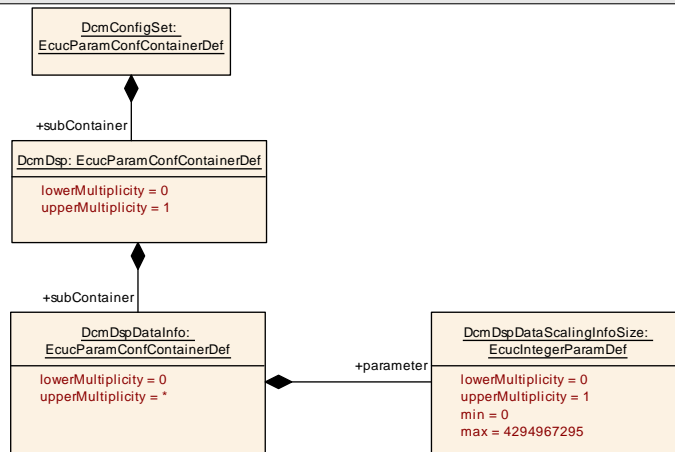


Figure 10.32: DcmDspDataInfo configuration overview

10.2.5.10 DcmDspDidControl

SWS Item	[ECUC_Dcm_00619]
Container Name	DcmDspDidControl
Parent Container	DcmDspDidInfo
Description	This container contains the configuration (parameters) of the DID control.
Configuration Parameters	

Name	DcmDspDidControlMask [ECUC_Dcm_01059]	
Parent Container	DcmDspDidControl	
Description	This indicates the presence of "controlEnableMask" in SWC service interfaces and defines how the Dcm treats a service request.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DCM_CONTROLMASK_E XTERNAL	The control enable mask record shall be forwarded within each interface and is handled externally.

	DCM_CONTROLMASK_INTERNAL	The control enable mask record is handled internally and Dcm controls only the included signals.	
	DCM_CONTROLMASK_NO	No control enable mask handling.	
Default Value	DCM_CONTROLMASK_INTERNAL		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidControlMaskSize [ECUC_Dcm_01060]		
Parent Container	DcmDspDidControl		
Description	The value defines the size of the controlEnableMaskRecord in bytes.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 4294967294		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidControlRole [ECUC_Dcm_01143]		
Parent Container	DcmDspDidControl		
Description	Bitfield were each bit represents one dedicated role. A DID is granted IOcontrol access if the bit value is 1. If a bit value is 0, the DID is not allowed to be controlled for that role.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	DcmDspDidFreezeCurrentState [ECUC_Dcm_00624]		
Parent Container	DcmDspDidControl		
Description	This indicates the presence of "FreezeCurrentState".		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidResetToDefault [ECUC_Dcm_00623]		
Parent Container	DcmDspDidControl		
Description	This indicates the presence of "ResetToDefault".		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidShortTermAdjustment [ECUC_Dcm_00625]		
Parent Container	DcmDspDidControl		
Description	This indicates the presence of "ShortTermAdjustment".		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidControlModeRuleRef [ECUC_Dcm_00923]		
Parent Container	DcmDspDidControl		
Description	Reference to DcmModeRule Mode rule which controls this DID. If there is no reference, no check of the mode rule shall be done.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidControlSecurityLevelRef [ECUC_Dcm_00620]		
Parent Container	DcmDspDidControl		
Description	Reference to DcmDspSecurityRow Security levels allowed to control this DID. If there is no reference, no check of security level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSecurityRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspDidControlSessionRef [ECUC_Dcm_00621]		
Parent Container	DcmDspDidControl		
Description	Reference to DcmDspSessionRow Sessions allowed to control this DID. If there is no reference, no check of session level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSessionRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspDidControl EnableMask	0..32	The shortname of the container value defines the symbol of the controlMask.

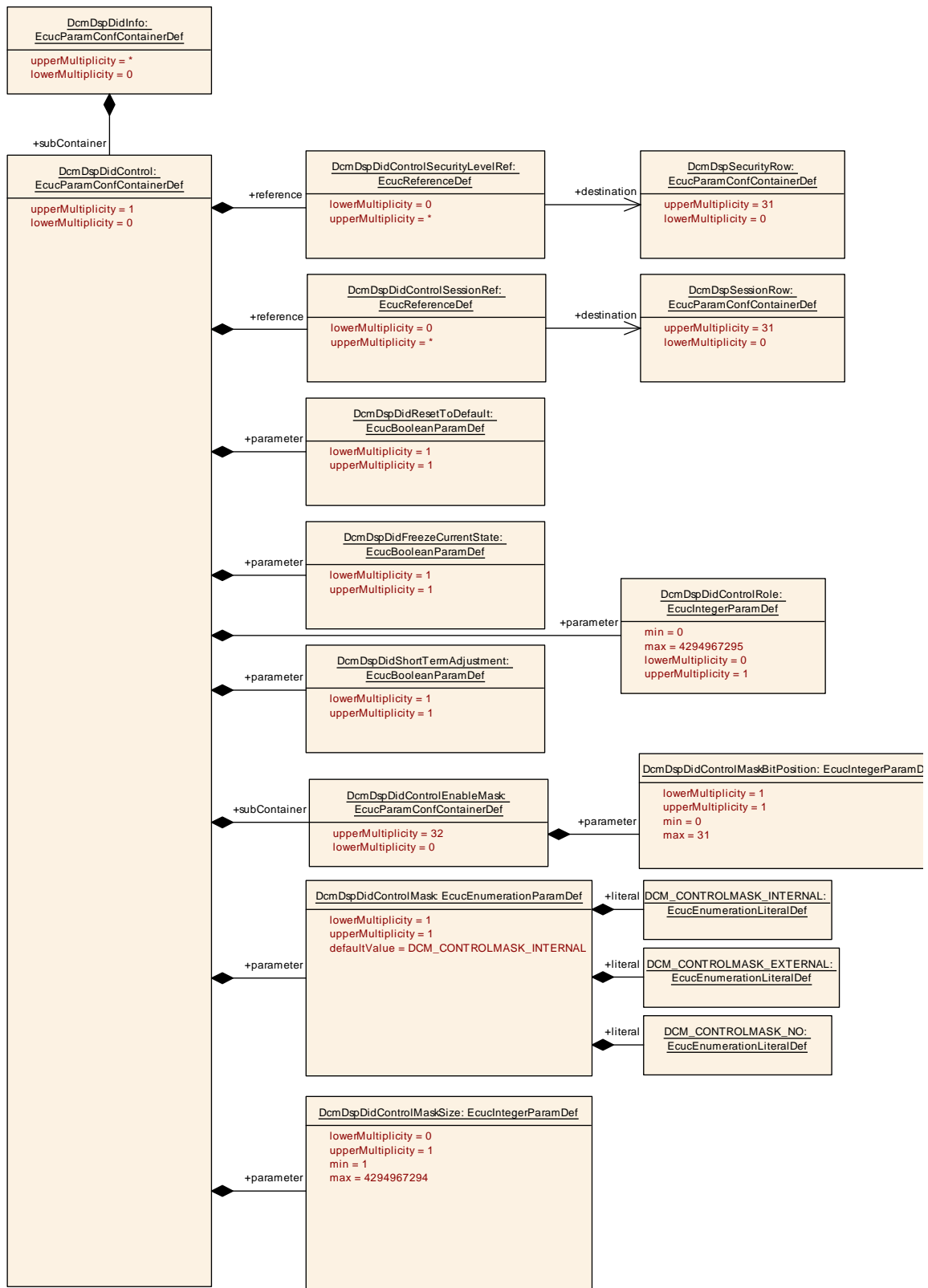


Figure 10.33: DcmDspDidControl configuration overview

10.2.5.11 DcmDspDidControlEnableMask

SWS Item	[ECUC_Dcm_01057]
Container Name	DcmDspDidControlEnableMask
Parent Container	DcmDspDidControl
Description	The shortname of the container value defines the symbol of the controlMask.
Configuration Parameters	

Name	DcmDspDidControlMaskBitPosition [ECUC_Dcm_01058]		
Parent Container	DcmDspDidControlEnableMask		
Description	Defines the position of the bit in the controlMask starting. The counting order is from the most significant bit (MSB first) to least significant bit. A value of 0 identifies the most significant bit in the MSB of the control enable mask. A value of 0 represents also the first controlState parameter. The bit endianness is identical to the controlMask in UDS. The number of possible and available bits is given by DcmDspDidControlMaskSize.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 31		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.12 Ecu Reset

10.2.5.12.1 DcmDspEcuReset

SWS Item	[ECUC_Dcm_01111]
Container Name	DcmDspEcuReset
Parent Container	DcmDsp
Description	This container contains the configuration for DcmDspEcuReset service
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspEcuResetRow	1..*	This container contains the configuration for each DcmDspEcuReset subservice.

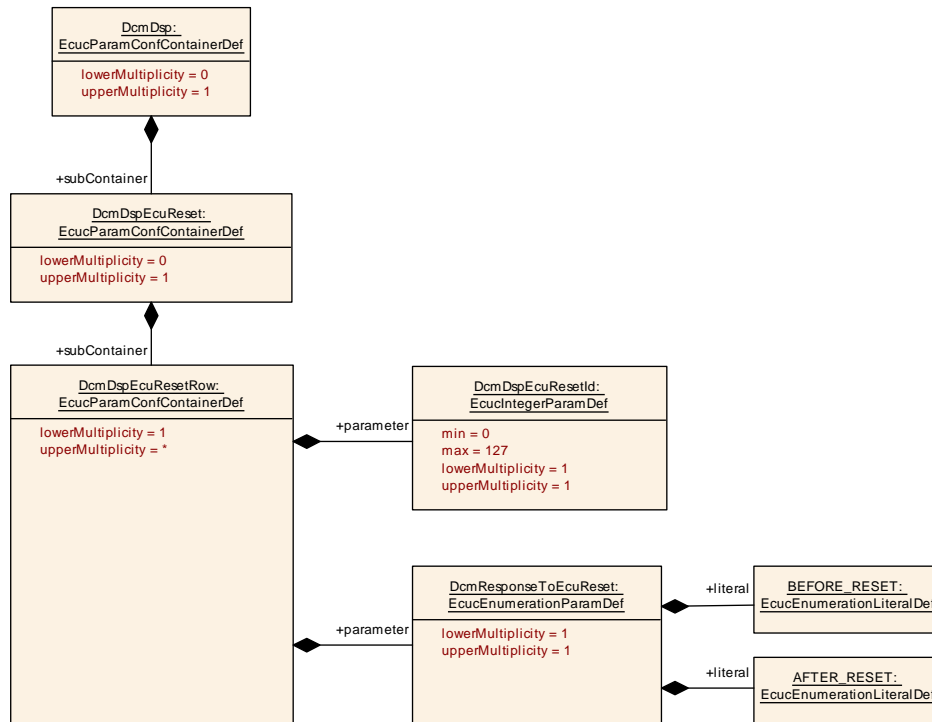


Figure 10.34: DcmDspEcuReset configuration overview

10.2.5.12.2 DcmDspEcuResetRow

SWS Item	[ECUC_Dcm_01112]
Container Name	DcmDspEcuResetRow
Parent Container	DcmDspEcuReset
Description	This container contains the configuration for each DcmDspEcuReset subservice.
Configuration Parameters	

Name	DcmDspEcuResetId [ECUC_Dcm_01113]	
Parent Container	DcmDspEcuResetRow	
Description	Defines the subfunction ID	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 127	
Default Value		
Post-Build Variant	false	
Multiplicity		

Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmResponseToEcuReset [ECUC_Dcm_01039]		
Parent Container	DcmDspEcuResetRow		
Description	Defines the answer to EcuReset service should come: Before or after the reset.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	AFTER_RESET	Answer to EcuReset service should come after the reset.	
	BEFORE_RESET	Answer to EcuReset service should come before the reset.	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.13 Memory

10.2.5.13.1 DcmDspMemory

SWS Item	[ECUC_Dcm_00784]
Container Name	DcmDspMemory
Parent Container	DcmDsp
Description	This container contains the configuration of the memory access.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspAddressAndLengthFormatIdentifier	0..1	This container contains the configuration of the supported AddressAndLengthFormatIdentifiers for memory access.
DcmDspMemoryIdInfo	1..*	Provides the value of memory identifier used to select the desired memory device This container contains the configuration of the memory access requested through diagnostic services : ReadMemoryByAddress, WriteMemoryByAddress, and DynamicallyDefineDataIdentifier

10.2.5.13.2 DcmDspMemoryTransfer

SWS Item	[ECUC_Dcm_01132]
Container Name	DcmDspMemoryTransfer
Parent Container	DcmDsp
Description	This container contains the configuration of the memory transfer.
Configuration Parameters	

Name	DcmDspMemoryTransferFnc [ECUC_Dcm_01134]		
Parent Container	DcmDspMemoryTransfer		
Description	Function name for memory transfer services. Only relevant if DcmDspMemoryTransferUsePort is set to false.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspMemoryTransferUsePort [ECUC_Dcm_01133]		
Parent Container	DcmDspMemoryTransfer		
Description	If this parameter is set to true, the Dcm uses a port requiring a PortInterface UploadDownload. If the parameter is false, the DCM uses the according C-API callouts.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspAddressAndLengthFormatIdentifier	0..1	This container contains the configuration of the supported AddressAndLengthFormatIdentifiers for memory access.
DcmDspMemoryTransferIdInfo	1..*	Provides the value of memory identifier used to select the desired memory device This container contains the configuration of the memory access requested through diagnostic services : RequestDownload, RequestUpload, TransferData, RequestTransferExit

10.2.5.13.3 DcmDspAddressAndLengthFormatIdentifier

SWS Item	[ECUC_Dcm_00963]
Container Name	DcmDspAddressAndLengthFormatIdentifier
Parent Container	DcmDspMemory , DcmDspMemoryTransfer
Description	This container contains the configuration of the supported AddressAndLengthFormatIdentifiers for memory access.
Configuration Parameters	

Name	DcmDspSupportedAddressAndLengthFormatIdentifier [ECUC_Dcm_00964]	
Parent Container	DcmDspAddressAndLengthFormatIdentifier	
Description	This parameter defines the supported AddressAndLengthFormatIdentifier of the request message.	
Multiplicity	1..*	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default Value		
Post-Build Variant Multiplicity	false	

Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

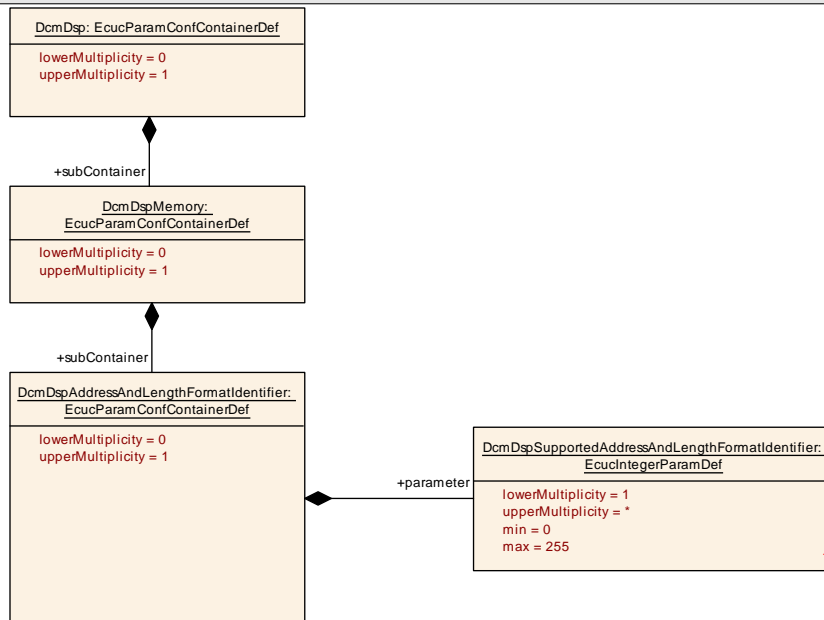


Figure 10.35: DcmDspMemoryFormatIdentifier configuration overview

10.2.5.13.4 DcmDspMemoryIdInfo

SWS Item	[ECUC_Dcm_00911]
Container Name	DcmDspMemoryIdInfo
Parent Container	DcmDspMemory
Description	Provides the value of memory identifier used to select the desired memory device This container contains the configuration of the memory access requested through diagnostic services : ReadMemoryByAddress, WriteMemoryByAddress, and DynamicallyDefineDataIdentifier
Configuration Parameters	

Name	DcmDspMemoryIdValue [ECUC_Dcm_00913]		
Parent Container	DcmDspMemoryIdInfo		
Description	<p>Value of the memory device identifier used.</p> <p>Each DcmDspMemoryIdInfo should have a unique ID.</p> <p>The MemoryIdValue is retrieved from the request messages (RMBA,WMBA,RD,RU,DDDI) according to ISO-14229-1 with the most significant byte of the request parameter memoryAddress.</p>		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspReadMemoryRangeByLabelInfo	0..*	Provides a memory range allowed for reading via labels (lower and higher address configured as strings).
DcmDspReadMemoryRangeInfo	0..*	Provides the range of memory address allowed for reading
DcmDspWriteMemoryRangeByLabelInfo	0..*	Provides a memory range allowed for writing via labels (lower and higher address configured as strings).
DcmDspWriteMemoryRangeInfo	0..*	Provides the range of memory address allowed for writing.

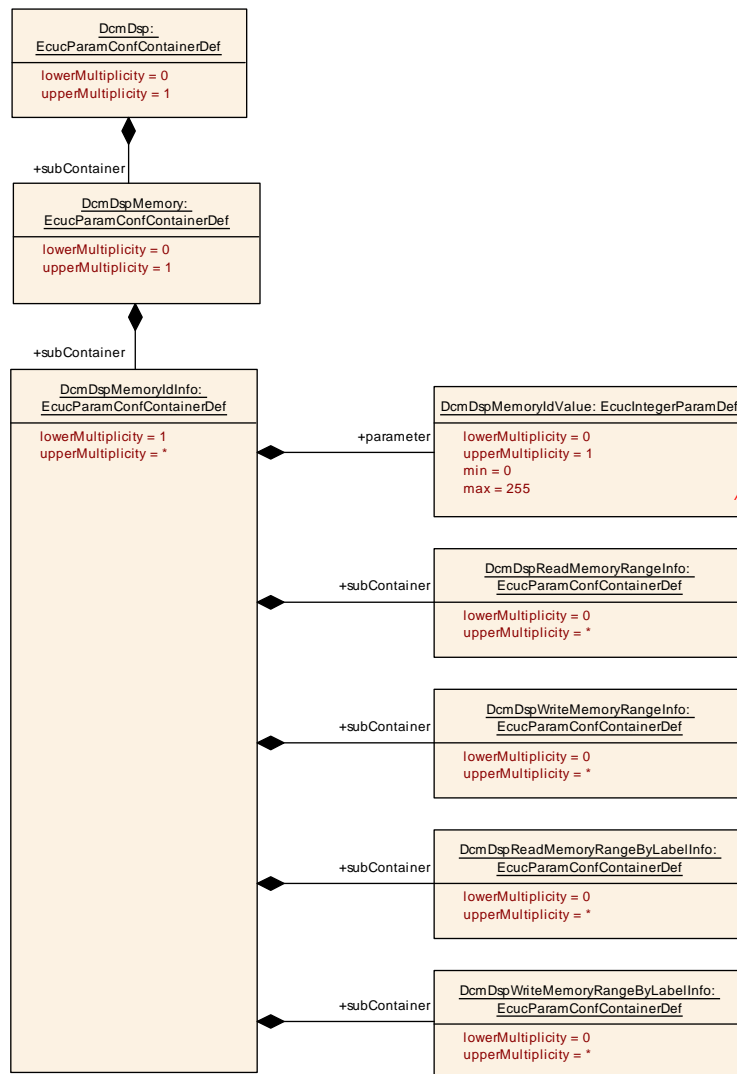


Figure 10.36: DcmDspMemoryIdInfo configuration overview

10.2.5.13.5 DcmDspMemoryTransferIdInfo

SWS Item	[ECUC_Dcm_01137]
Container Name	DcmDspMemoryTransferIdInfo
Parent Container	DcmDspMemoryTransfer
Description	<p>Provides the value of memory identifier used to select the desired memory device</p> <p>This container contains the configuration of the memory access requested through diagnostic services : RequestDownload, RequestUpload, TransferData, RequestTransferExit</p>
Configuration Parameters	

Name	DcmDspMemoryIdValue [ECUC_Dcm_01138]		
Parent Container	DcmDspMemoryTransferIdInfo		
Description	<p>Value of the memory device identifier used.</p> <p>Each DcmDspMemoryIdInfo should have a unique ID.</p> <p>The MemoryIdValue is retrieved from the request messages (RMBA,WMBA,RD,RU,DDDI) according to ISO-14229-1 with the most significant byte of the request parameter memoryAddress.</p>		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.13.6 DcmDspReadMemoryRangeByLabelInfo

SWS Item	[ECUC_Dcm_01068]
Container Name	DcmDspReadMemoryRangeByLabelInfo
Parent Container	DcmDspMemoryIdInfo
Description	Provides a memory range allowed for reading via labels (lower and higher address configured as strings).
Configuration Parameters	

Name	DcmDspReadMemoryRangeByLabelHigh [ECUC_Dcm_01070]
Parent Container	DcmDspReadMemoryRangeByLabelInfo
Description	High memory address as label (string) of a range allowed for reading.
Multiplicity	1
Type	EcucStringParamDef
Default Value	
Regular Expression	[a-zA-Z0-9_]([a-zA-Z0-9\._])*
Post-Build Variant Value	false

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspReadMemoryRangeByLabelLow [ECUC_Dcm_01069]		
Parent Container	DcmDspReadMemoryRangeByLabelInfo		
Description	Low memory address as label (string) of a range allowed for reading.		
Multiplicity	1		
Type	EcucStringParamDef		
Default Value			
Regular Expression	[a-zA-Z0-9_]([a-zA-Z0-9\._])*		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspReadMemoryRangeModeRuleRef [ECUC_Dcm_01072]		
Parent Container	DcmDspReadMemoryRangeByLabelInfo		
Description	Reference to DcmModeRule		
	Mode rule which controls read access on this memory address. If there is no reference, no check of the mode rule shall be done.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspReadMemoryRangeSecurityLevelRef [ECUC_Dcm_01071]		
Parent Container	DcmDspReadMemoryRangeByLabelInfo		
Description	Link to the Security Access Levels needed for read access on this memory address. If there is no reference, no check of security level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSecurityRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspReadMemoryRangeSessionLevelRef [ECUC_Dcm_01088]		
Parent Container	DcmDspReadMemoryRangeByLabelInfo		
Description	Link to the session level needed for access to this memory address range. If there is no reference, no check of session level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSessionRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.13.7 DcmDspReadMemoryRangeInfo

SWS Item	[ECUC_Dcm_00785]
-----------------	------------------

Container Name	DcmDspReadMemoryRangeInfo
Parent Container	DcmDspMemoryIdInfo
Description	Provides the range of memory address allowed for reading
Configuration Parameters	

Name	DcmDspReadMemoryRangeHigh [ECUC_Dcm_00787]		
Parent Container	DcmDspReadMemoryRangeInfo		
Description	High memory address of a range allowed for reading		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967294		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspReadMemoryRangeLow [ECUC_Dcm_00786]		
Parent Container	DcmDspReadMemoryRangeInfo		
Description	Low memory address of a range allowed for reading		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967294		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspReadMemoryRangeModeRuleRef [ECUC_Dcm_00920]		
Parent Container	DcmDspReadMemoryRangeInfo		
Description	Reference to DcmModeRule Mode rule which controls read access on this memory address. If there is no reference, no check of the mode rule shall be done.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspReadMemoryRangeSecurityLevelRef [ECUC_Dcm_00788]		
Parent Container	DcmDspReadMemoryRangeInfo		
Description	Link to the Security Access Levels needed for read access on this memory address. If there is no reference, no check of security level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSecurityRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspReadMemoryRangeSessionLevelRef [ECUC_Dcm_01086]		
Parent Container	DcmDspReadMemoryRangeInfo		
Description	Link to the session level needed for access to this memory address range. If there is no reference, no check of session level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSessionRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.13.8 DcmDspWriteMemoryRangeByLabelInfo

SWS Item	[ECUC_Dcm_01073]
Container Name	DcmDspWriteMemoryRangeByLabelInfo
Parent Container	DcmDspMemoryIdInfo
Description	Provides a memory range allowed for writing via labels (lower and higher address configured as strings).
Configuration Parameters	

Name	DcmDspWriteMemoryRangeByLabelHigh [ECUC_Dcm_01075]		
Parent Container	DcmDspWriteMemoryRangeByLabelInfo		
Description	High memory address as label (string) of a range allowed for writing.		
Multiplicity	1		
Type	EcucStringParamDef		
Default Value			
Regular Expression	[a-zA-Z0-9_]([a-zA-Z0-9\._])*		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspWriteMemoryRangeByLabelLow [ECUC_Dcm_01074]		
Parent Container	DcmDspWriteMemoryRangeByLabelInfo		
Description	Low memory address as label (string) of a range allowed for writing.		
Multiplicity	1		
Type	EcucStringParamDef		
Default Value			
Regular Expression	[a-zA-Z0-9_]([a-zA-Z0-9\._])*		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Scope / Dependency	scope: local
---------------------------	--------------

Name	DcmDspWriteMemoryRangeModeRuleRef [ECUC_Dcm_01077]		
Parent Container	DcmDspWriteMemoryRangeByLabelInfo		
Description	Reference to DcmModeRule Mode rule which controls write access on this memory address. If there is no reference, no check of the mode rule shall be done.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspWriteMemoryRangeSecurityLevelRef [ECUC_Dcm_01076]		
Parent Container	DcmDspWriteMemoryRangeByLabelInfo		
Description	Link to the Security Access Levels needed for write access on this memory address. If there is no reference, no check of security level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSecurityRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspWriteMemoryRangeSessionLevelRef [ECUC_Dcm_01089]		
Parent Container	DcmDspWriteMemoryRangeByLabelInfo		
Description	Link to the session level needed for access to this memory address range. If there is no reference, no check of session level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSessionRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

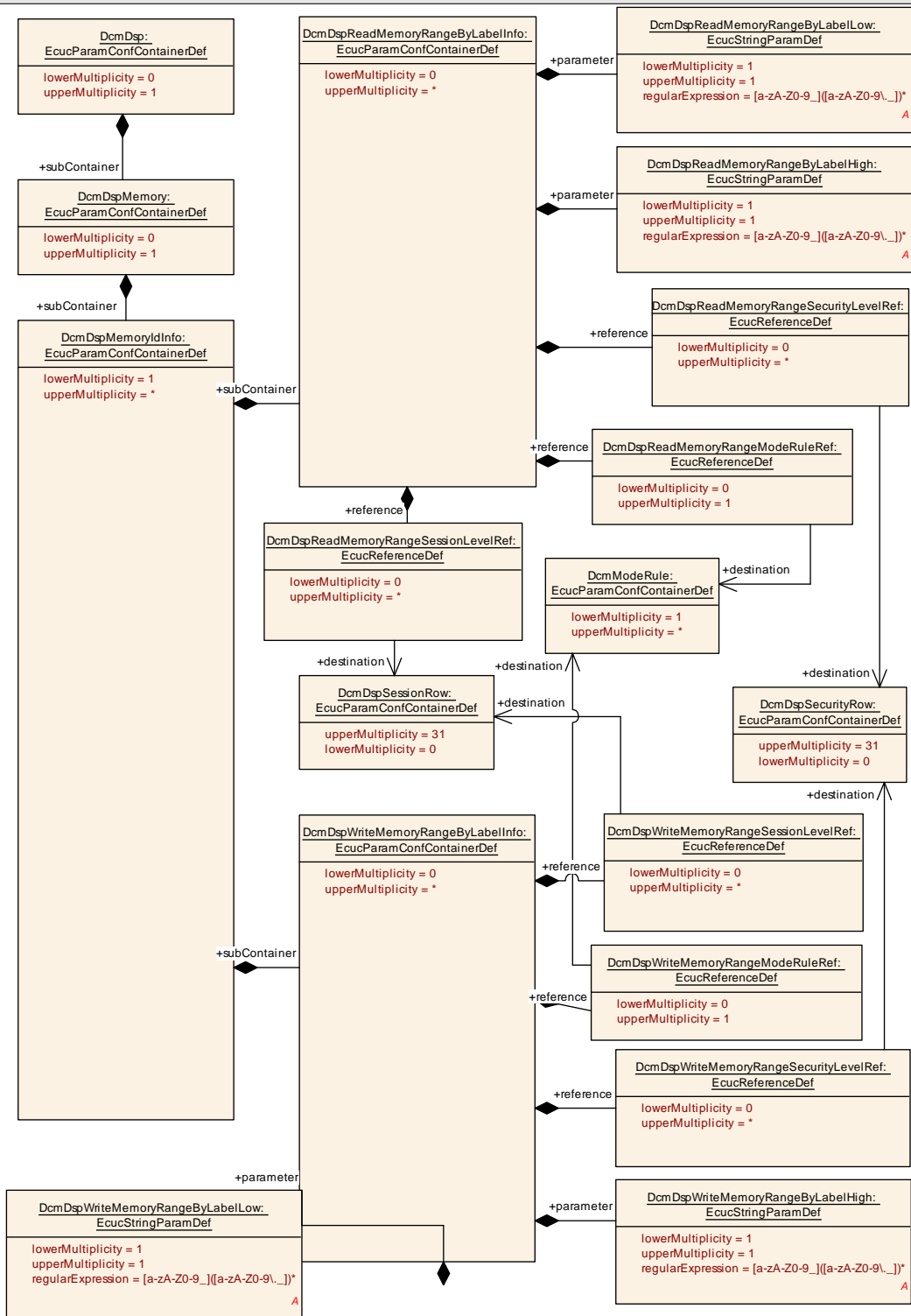


Figure 10.37: DcmDspMemoryReadWriteByLabel configuration overview

10.2.5.13.9 DcmDspWriteMemoryRangeInfo

SWS Item	[ECUC_Dcm_00789]
Container Name	DcmDspWriteMemoryRangeInfo
Parent Container	DcmDspMemoryIdInfo
Description	Provides the range of memory address allowed for writing.
Configuration Parameters	

Name	DcmDspWriteMemoryRangeHigh [ECUC_Dcm_00791]		
Parent Container	DcmDspWriteMemoryRangeInfo		
Description	High memory address of a range allowed for writing.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967294		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspWriteMemoryRangeLow [ECUC_Dcm_00790]		
Parent Container	DcmDspWriteMemoryRangeInfo		
Description	Low memory address of a range allowed for writing		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967294		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspWriteMemoryRangeModeRuleRef [ECUC_Dcm_00916]		
Parent Container	DcmDspWriteMemoryRangeInfo		
Description	Reference to DcmModeRule Mode rule which controls write access on this memory address. If there is no reference, no check of the mode rule shall be done.		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspWriteMemoryRangeSecurityLevelRef [ECUC_Dcm_00793]		
Parent Container	DcmDspWriteMemoryRangeInfo		
Description	Link to the Security Access Levels needed for write access on this memory address. If there is no reference, no check of security level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSecurityRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspWriteMemoryRangeSessionLevelRef [ECUC_Dcm_01087]		
Parent Container	DcmDspWriteMemoryRangeInfo		
Description	Link to the session level needed for access to this memory address range. If there is no reference, no check of session level shall be done.		
Multiplicity	0..*		
Type	Reference to DcmDspSessionRow		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

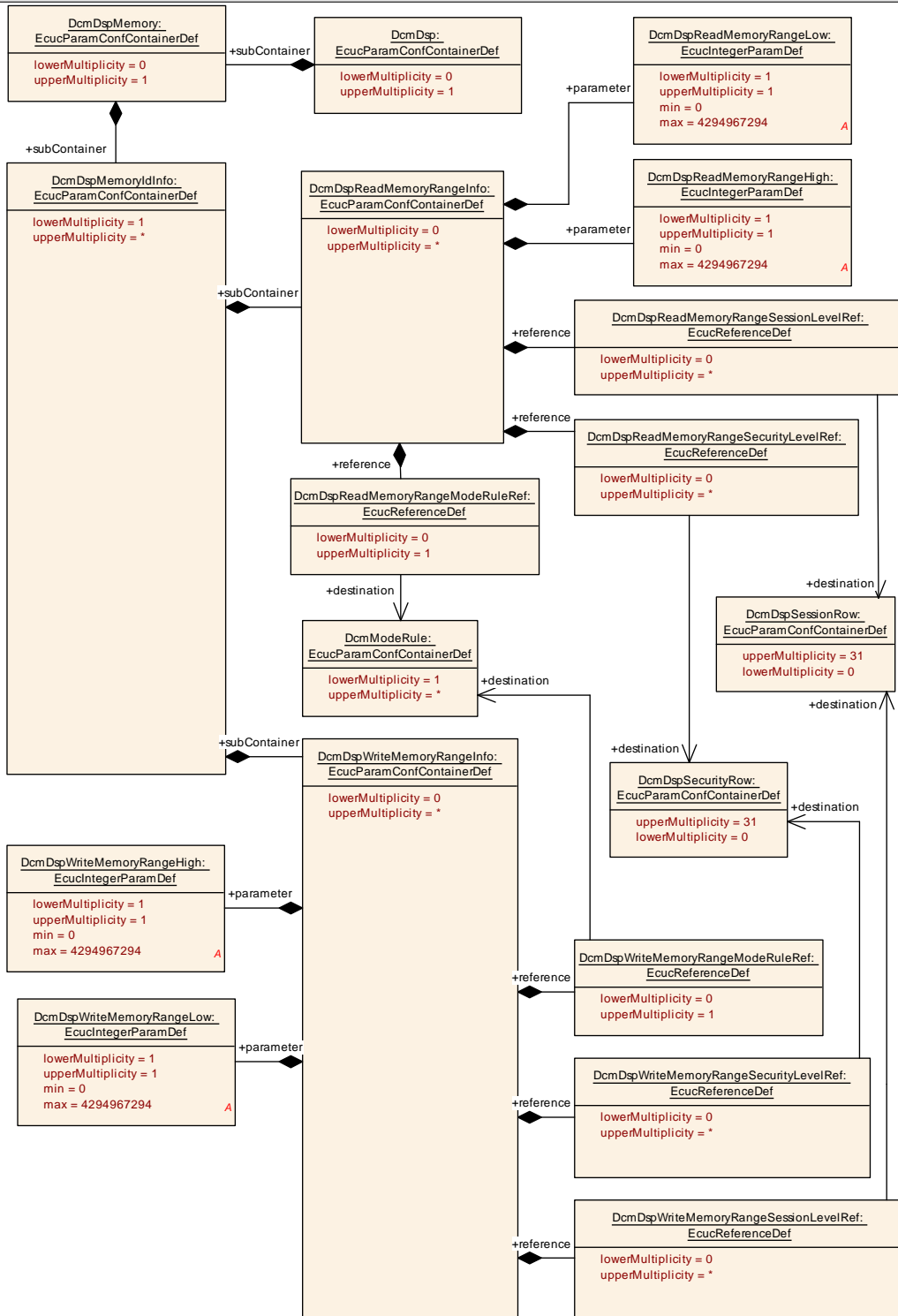


Figure 10.38: DcmDspMemoryReadWrite configuration overview

10.2.5.14 PIDs

10.2.5.14.1 DcmDspPid

SWS Item	[ECUC_Dcm_00626]
Container Name	DcmDspPid
Parent Container	DcmDsp
Description	This container defines the availability of a PID to the DCM.
Configuration Parameters	

Name	DcmDspPidIdentifier [ECUC_Dcm_00627]		
Parent Container	DcmDspPid		
Description	1 byte Identifier of the PID Within each DcmConfigSet all DcmDspPidIdentifier values shall be unique.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspPidService [ECUC_Dcm_00893]		
Parent Container	DcmDspPid		
Description	Indicates if a PID is used with service \$01 and/or \$02		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DCM_SERVICE_01	A PID is used with service \$01 only.	
	DCM_SERVICE_01_02	A PID is used with service \$01 and \$02. Allowed with a PID configuration containing data elements on byte basis.	
	DCM_SERVICE_02	A PID is used with service \$02 only. Allowed with a PID configuration containing data elements on byte basis.	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspPidSize [ECUC_Dcm_00870]		
Parent Container	DcmDspPid		
Description	Length of a PID in byte(s).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspPidUsed [ECUC_Dcm_00806]		
Parent Container	DcmDspPid		
Description	Allow to activate or deactivate the usage of a PID, for multi purpose ECUs true = PID is available false = PID is not available		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspPidData	1..*	This container defines the parameter for a Signal in the PID.
DcmDspPidSupportInfo	0..*	This container defines the support information (typically byte A) to declare the usability of the data bytes within the so-called packeted PIDs (e.g. PID\$68).

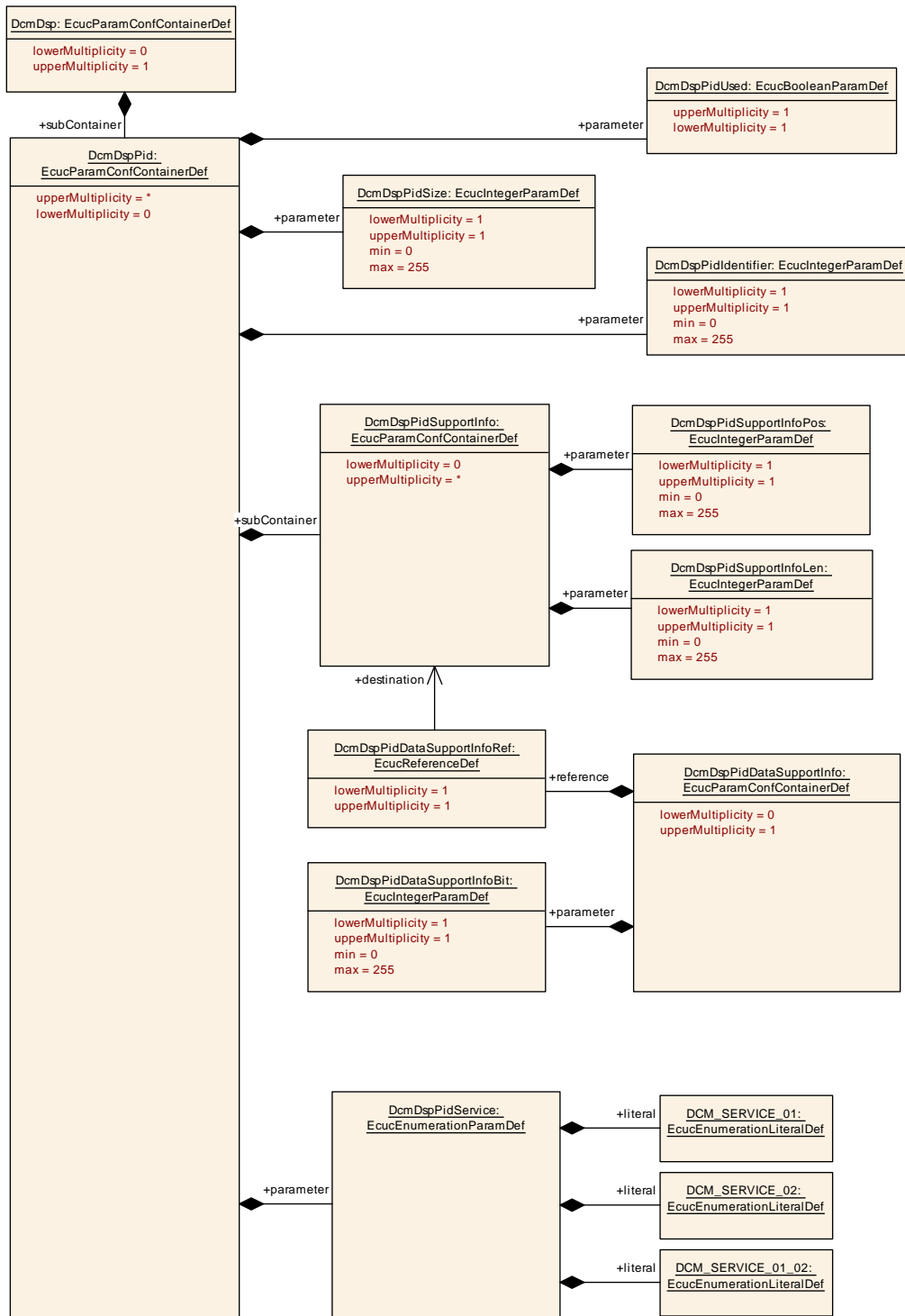


Figure 10.39: DcmDspPid1 configuration overview

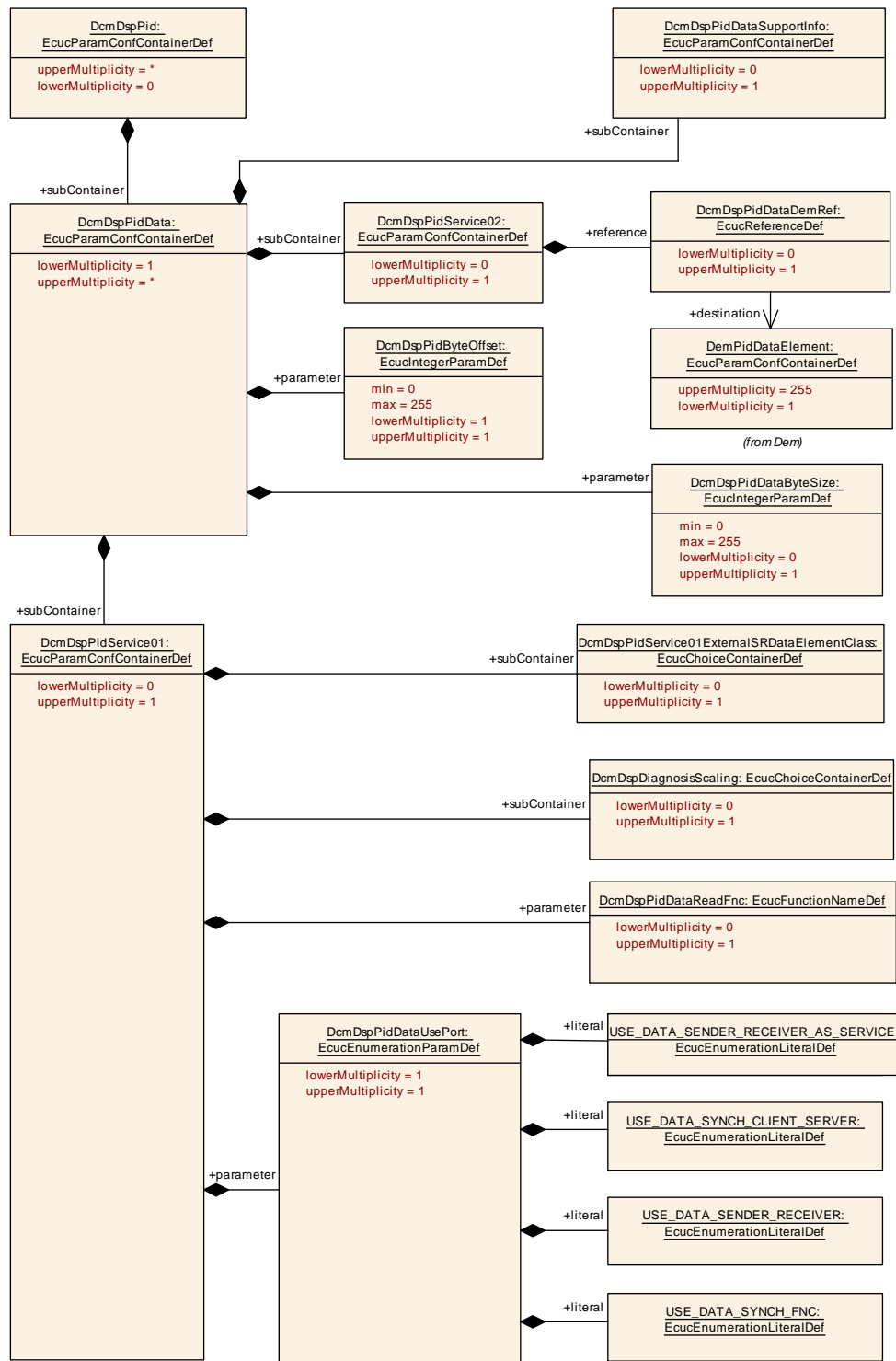


Figure 10.40: DcmDspPid2 configuration overview

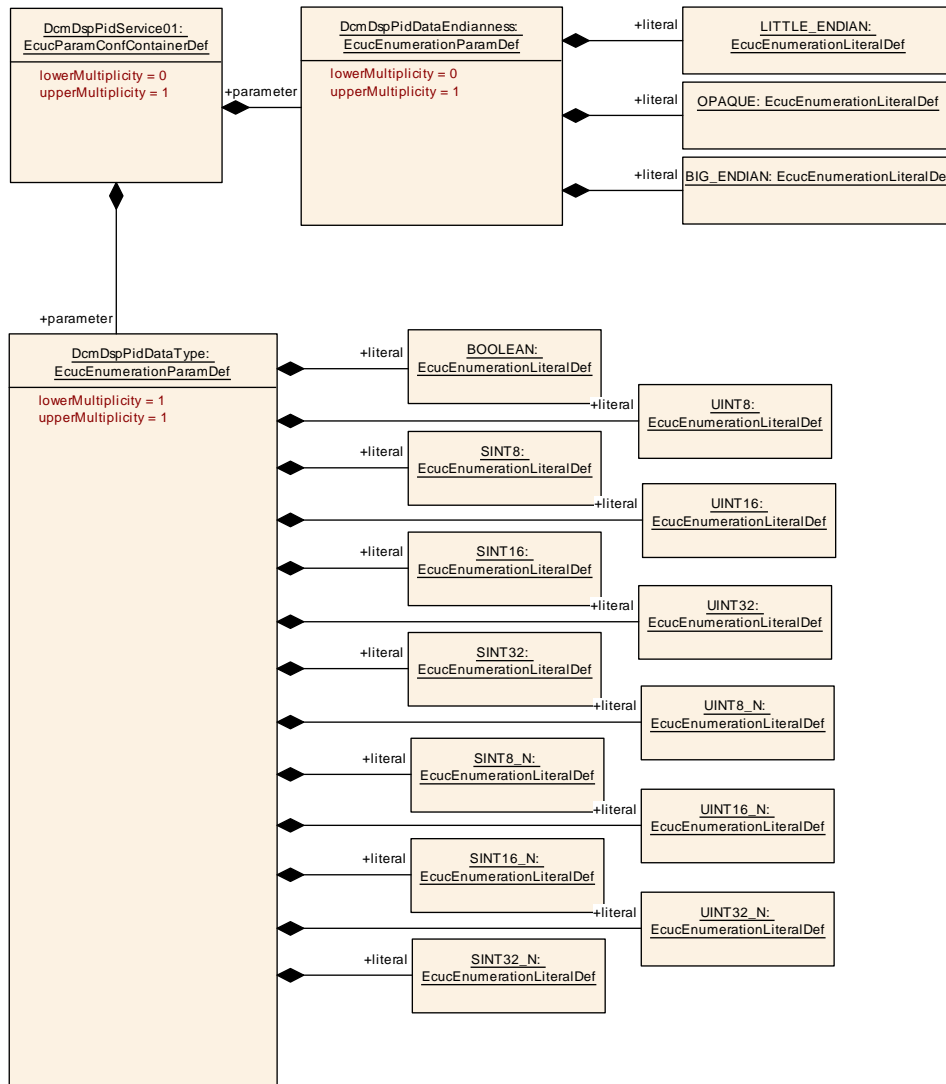


Figure 10.41: DcmDspPid3 configuration overview

10.2.5.14.2 DcmDspPidSupportInfo

SWS Item	[ECUC_Dcm_00871]
Container Name	DcmDspPidSupportInfo
Parent Container	DcmDspPid
Description	This container defines the support information (typically byte A) to declare the usability of the data bytes within the so-called packeted PIDs (e.g. PID\$68).
Configuration Parameters	

Name	DcmDspPidSupportInfoLen [ECUC_Dcm_00873]		
Parent Container	DcmDspPidSupportInfo		
Description	Length of the support information in bytes.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspPidSupportInfoPos [ECUC_Dcm_00872]		
Parent Container	DcmDspPidSupportInfo		
Description	Position of the support information in bytes.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.14.3 DcmDspPidData

SWS Item	[ECUC_Dcm_00865]
Container Name	DcmDspPidData
Parent Container	DcmDspPid
Description	This container defines the parameter for a Signal in the PID.
Configuration Parameters	

Name	DcmDspPidByteOffset [ECUC_Dcm_01107]		
Parent Container	DcmDspPidData		
Description	This is the position in bytes of the PID structure and will not start at position 0 in case a support information is available (for packeted PIDs).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspPidDataByteSize [ECUC_Dcm_01108]		
Parent Container	DcmDspPidData		
Description	Defines the array length in bytes or the the maximum array length for variable datalengths.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspPidDataSupport Info	0..1	This container defines the supported information.
DcmDspPidService01	0..1	Contains specific configuration parameter of PID for service \$01. This container exists only if DcmDspPidService is set to DCM_SERVICE_01 or DCM_SERVICE_01_02.
DcmDspPidService02	0..1	Contains specific configuration parameter of PID for service \$02. This container exists only if DcmDspPidService is set to DCM_SERVICE_02 or DCM_SERVICE_01_02.

10.2.5.14.4 DcmDspPidService01

SWS Item	[ECUC_Dcm_00894]
Container Name	DcmDspPidService01
Parent Container	DcmDspPidData
Description	Contains specific configuration parameter of PID for service \$01. This container exists only if DcmDspPidService is set to DCM_SERVICE_01 or DCM_SERVICE_01_02.
Configuration Parameters	

Name	DcmDspPidDataEndianness [ECUC_Dcm_01012]		
Parent Container	DcmDspPidService01		
Description	<p>Defines the endianness of the data belonging to a PID in a diagnostic response message.</p> <p>If no DcmDspPidDataEndianness is defined the value of DcmDspDataDefaultEndianness is applicable.</p>		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	BIG_ENDIAN	Most significant byte shall be stored at the lowest address.	
	LITTLE_ENDIAN	Most significant byte shall be stored at the highest address	
	OPAQUE	Opaque data endianness	
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspPidDataReadFnc [ECUC_Dcm_00629]		
Parent Container	DcmDspPidService01		
Description	<p>Function name for reading PID data value. This is only relevant if DcmDspPidDataUsePort==USE_DATA_SYNCH_FNC.</p> <p>This parameter is related to the interface Xxx_ReadData.</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		

Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspPidDataType [ECUC_Dcm_01018]		
Parent Container	DcmDspPidService01		
Description	Provide the implementation data type of data belonging to a PID.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BOOLEAN	Type of the data is boolean.	
	SINT16	Type of the data is sint16.	
	SINT16_N	Type of the data is sint16 array.	
	SINT32	Type of the data is sint32.	
	SINT32_N	Type of the data is sint32 array.	
	SINT8	Type of the data is sint8.	
	SINT8_N	Type of the data is sint8 array.	
	UINT16	Type of the data is uint16.	
	UINT16_N	Type of the data is uint16 array.	
	UINT32	Type of the data is uint32.	
	UINT32_N	Type of the data is uint32 array.	
	UINT8	Type of the data is uint8.	
	UINT8_N	Type of the data is uint8 array.	
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspPidDataUsePort [ECUC_Dcm_00720]		
Parent Container	DcmDspPidService01		
Description	<p>If this parameter is set to USE_DATA_SYNCH_FNC, the Dcm will use the function defined in DcmDspPidDataReadFnc to get the PID data value.</p> <p>If this parameter is set to USE_DATA_SYNCH_CLIENT_SERVER, the Dcm will have an R-Port requiring the interface DataServices_{Data}.</p> <p>If this parameter is set to USE_DATA_SENDER_RECEIVER or USE_DATA_SENDER_RECEIVER_AS_SERVICE, the DCM will have an R-Port requiring a SenderReceiverInterface.</p> <p>The R-Port is named DataServices_{Data} where {Data} is the name of the container DcmDspPidData.</p>		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	USE_DATA_SENDER_RECEIVER		
	USE_DATA_SENDER_RECEIVER_AS_SERVICE		
	USE_DATA_SYNCH_CLIENT_SERVER		
	USE_DATA_SYNCH_FNC		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspDiagnosisScaling	0..1	This container contains the configuration (parameters) of an alternative Diagnosis Representation. Out if this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.
DcmDspPidService01 ExternalSRDataElement Class	0..1	<p>This container defines the source of data in a provided port which shall be read respectively the target of data in a required port which shall be written.</p> <p>This container shall contain either one DcmSubElementInDataElementInstance OR DcmDataElementInstance OR DcmSubElementInImplDataElementInstance reference.</p>

10.2.5.14.5 DcmDspPidService02

SWS Item	[ECUC_Dcm_00895]
Container Name	DcmDspPidService02
Parent Container	DcmDspPidData

Description	Contains specific configuration parameter of PID for service \$02. This container exists only if DcmDspPidService is set to DCM_SERVICE_02 or DCM_SERVICE_01_02.
Configuration Parameters	

Name	DcmDspPidDataDemRef [ECUC_Dcm_00887]		
Parent Container	DcmDspPidService02		
Description	Reference to DemPidDataElement in DEM configuration. Allows to link the DCM PID and DEM PID configuration for Mode \$02.		
Multiplicity	0..1		
Type	Reference to DemPidDataElement		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.14.6 DcmDspPidDataSupportInfo

SWS Item	[ECUC_Dcm_00874]
Container Name	DcmDspPidDataSupportInfo
Parent Container	DcmDspPidData
Description	This container defines the supported information.
Configuration Parameters	

Name	DcmDspPidDataSupportInfoBit [ECUC_Dcm_00876]	
Parent Container	DcmDspPidDataSupportInfo	
Description	Referenced Bit of the SupportInfo	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default Value		
Post-Build Variant Value	false	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspPidDataSupportInfoRef [ECUC_Dcm_00875]		
Parent Container	DcmDspPidDataSupportInfo		
Description	Reference to DcmDspPidSupportInfo		
Multiplicity	1		
Type	Reference to DcmDspPidSupportInfo		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.15 DcmDspRequestControl

SWS Item	[ECUC_Dcm_00637]
Container Name	DcmDspRequestControl
Parent Container	DcmDsp
Description	This container contains the configuration (parameters) of the "Request control of on-board system, test or component" service (Service \$08). The DCM will request the control using an R-Port requiring a PortInterface RequestControlServices_{Tid}. The R-Port is named RequestControlServices_{Tid} where {Tid} is the name of the container DcmDspRequestControl.
Configuration Parameters	

Name	DcmDspRequestControlInBufferSize [ECUC_Dcm_00722]	
Parent Container	DcmDspRequestControl	
Description	Number of bytes to be provided in the input buffer of the interface RequestControlServices_{Tid} for OBD Service \$08	
Multiplicity	1	
Type	EcuIntegerParamDef	
Range	0 .. 255	
Default Value		
Post-Build Variant Value	false	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRequestControlInfoByte [ECUC_Dcm_01078]		
Parent Container	DcmDspRequestControl		
Description	Manufacturer specific value reported to the tester for the record identifiers 0xE000 to 0xE1FF. (WWH-OBD use cases)		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRequestControlOutBufferSize [ECUC_Dcm_00723]		
Parent Container	DcmDspRequestControl		
Description	Number of bytes to be provided in the output buffer of the interface RequestControlServices_{Tid} for OBD Service \$08		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRequestControlTestId [ECUC_Dcm_00656]		
Parent Container	DcmDspRequestControl		
Description	Test Id for Service \$08		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: ECU		

No Included Containers

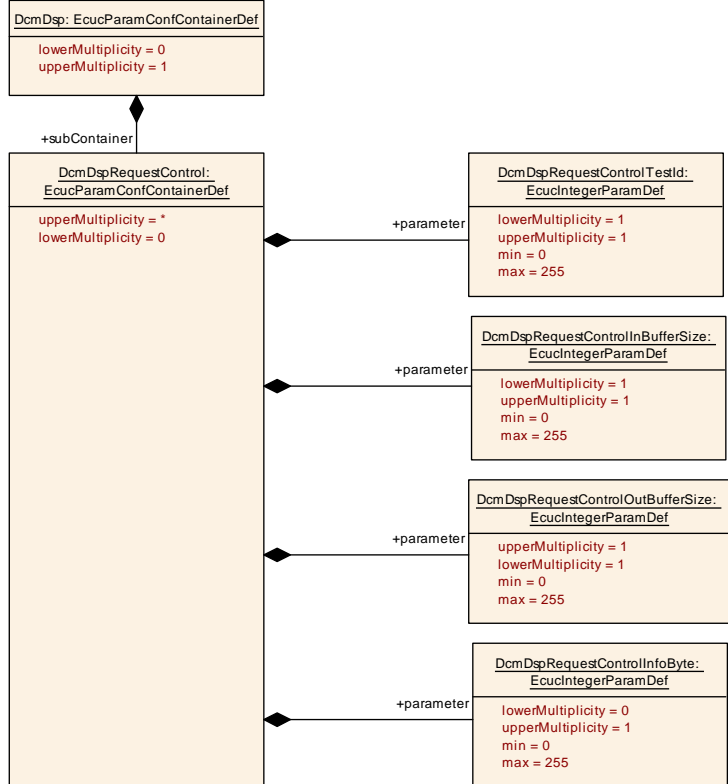


Figure 10.42: DcmDspRequestControl configuration overview

10.2.5.16 DcmDspRequestFileTransfer

SWS Item	[ECUC_Dcm_01034]
Container Name	DcmDspRequestFileTransfer
Parent Container	DcmDsp
Description	This container contains the configuration for RequestFileTransfer. This container only exists if RequestFileTransfer is configured.
Configuration Parameters	

Name	DcmRequestFileTransferFileSizeOrDirInfoParameterLength [ECUC_Dcm_01035]		
Parent Container	DcmDspRequestFileTransfer		
Description	Defines the length (number of bytes, i.e. the value of fileSizeOrDirInfoParameterLength) of the fileSizeUncompressedOrDirInfoLength and fileSizeCompressed in the response of RequestFileTransfer.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 4		
Default Value	4		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmRequestFileTransferLengthFormatIdentifier [ECUC_Dcm_01036]		
Parent Container	DcmDspRequestFileTransfer		
Description	Defines the length (number of bytes) of the maxNumberOfBlockLength parameter in the response of RequestFileTransfer.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 4		
Default Value	4		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmRequestFileTransferMaxFileAndDirName [ECUC_Dcm_01130]		
Parent Container	DcmDspRequestFileTransfer		
Description	Defines the maximum size allowed for the FileAndDirName parameter with RTE interfaces used for RequestFileTransfer.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default Value	1		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmRequestFileTransferUsePort [ECUC_Dcm_01131]		
Parent Container	DcmDspRequestFileTransfer		
Description	Defines if a C/S or C function call shall be used for RequestFileTransfer processing.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

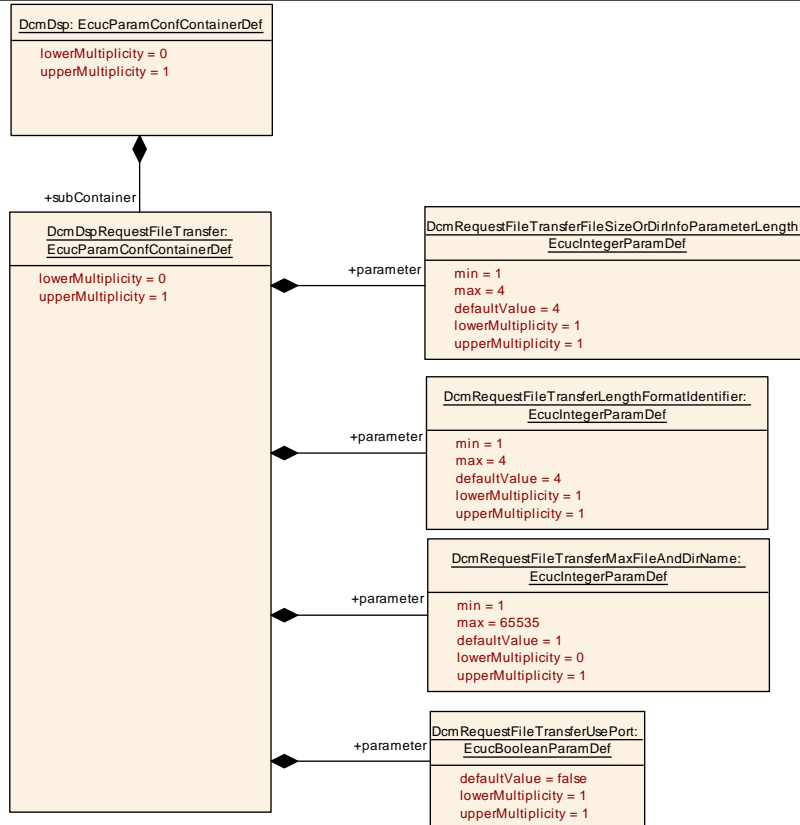


Figure 10.43: DcmDspRequestFileTransfer configuration overview

10.2.5.17 Response on Event

10.2.5.17.1 DcmDspRoe

SWS Item	[ECUC_Dcm_00858]
Container Name	DcmDspRoe

Parent Container	DcmDsp
Description	Provide the configuration of the ResponseOnEvent mechanism.
Configuration Parameters	

Name	DcmDspRoelnterMessageTime [ECUC_Dcm_00856]		
Parent Container	DcmDspRoe		
Description	Provide the minimum time in seconds between two transmissions of ROE event. It is used for the delay between two different consecutive Roe transmissions.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. 5]		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspRoeEvent	1..255	This container contains a list of all supported Roe eventTypeRecords which are accepted by this ECU. At most one DcmDspRoeEvent container is allowed to define a DcmDspRoeEventProperties container with the choice DcmDspRoeOnDTCStatusChange.
DcmDspRoeEventWindowTime	1..*	This container configures the available EventWindowTime in this Ecu. This container contains a sub-set of EventWindowTimes supported by the Dcm, to limit the Ecu resources.

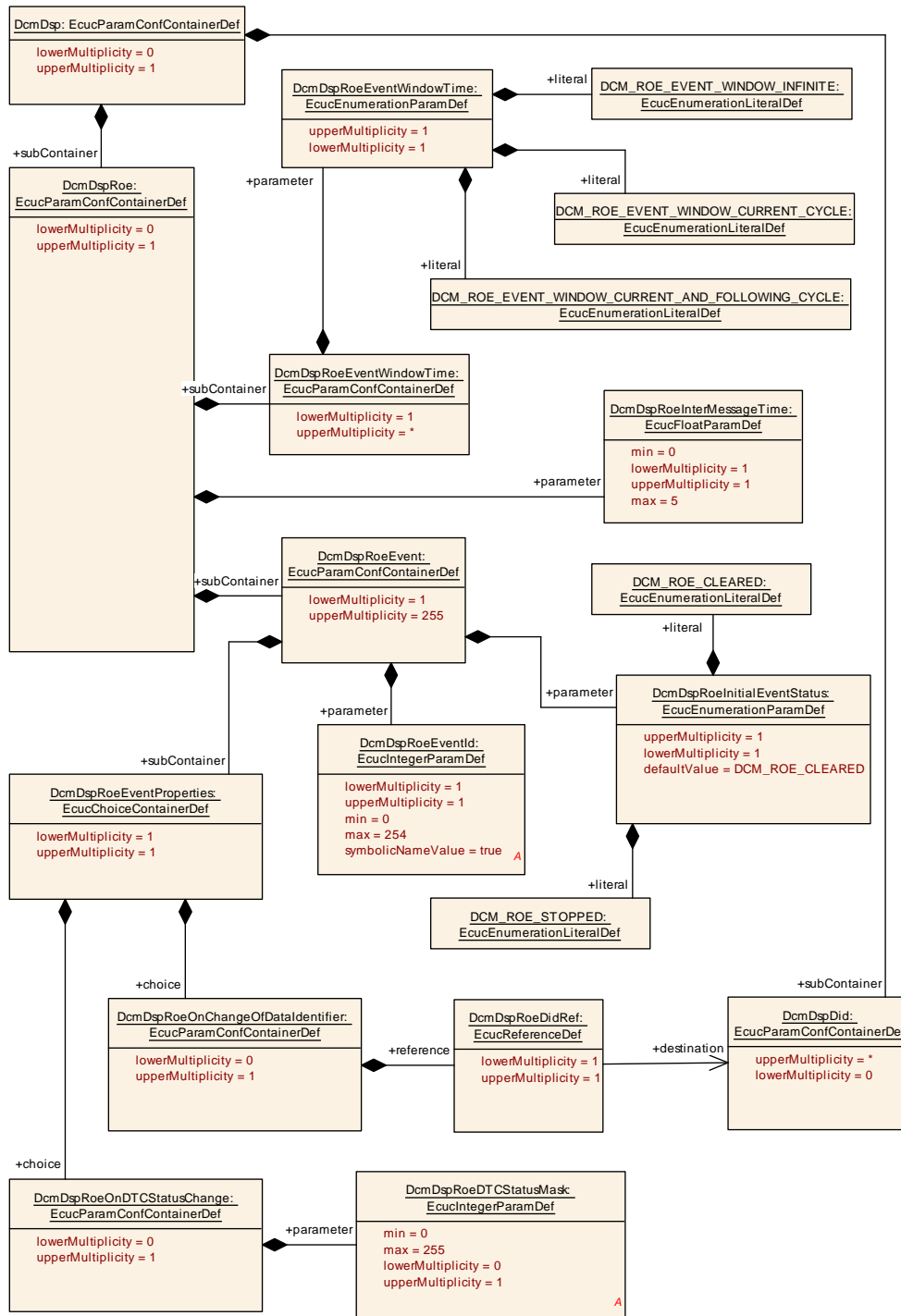


Figure 10.44: DcmDspRoe configuration overview

10.2.5.17.2 DcmDspRoeEvent

SWS Item	[ECUC_Dcm_00973]
Container Name	DcmDspRoeEvent
Parent Container	DcmDspRoe

Description	<p>This container contains a list of all supported Roe eventTypeRecords which are accepted by this ECU.</p> <p>At most one DcmDspRoeEvent container is allowed to define a DcmDspRoeEventProperties container with the choice DcmDspRoeOnDTCSStatusChange.</p>
Configuration Parameters	

Name	DcmDspRoeEventId [ECUC_Dcm_00976]		
Parent Container	DcmDspRoeEvent		
Description	<p>EventId for a global identification of this ROE event it is used within APIs Dcm_TriggerOnEvent() and the ModeDeclarationGroup.</p> <p>The ratio Ids should be sequentially ordered beginning with 0 and no gaps in between.</p>		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 254		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspRoeInitialEventStatus [ECUC_Dcm_00980]		
Parent Container	DcmDspRoeEvent		
Description	Initial Roe status of this RoeEvent		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DCM_ROE_CLEARED		
	DCM_ROE_STOPPED		
Default Value	DCM_ROE_CLEARED		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspRoeEventProperties	1	<p>This container contains the properties of Roe eventTypeRecords.</p> <p>In one DcmDspRoeEventProperties container one DcmDspRoeOnDTCStatusChange or DcmDspRoeOnChangeOfDataIdentifier container shall be defined.</p>

10.2.5.17.3 DcmDspRoeEventProperties

SWS Item	[ECUC_Dcm_00978]
Container Name	DcmDspRoeEventProperties
Parent Container	DcmDspRoeEvent
Description	<p>This container contains the properties of Roe eventTypeRecords.</p> <p>In one DcmDspRoeEventProperties container one DcmDspRoeOnDTCStatusChange or DcmDspRoeOnChangeOfDataIdentifier container shall be defined.</p>
Configuration Parameters	

Container Choices		
Container Name	Multiplicity	Scope / Dependency
DcmDspRoeOnChangeOfDataIdentifier	0..1	This container contains the eventTypeRecord supported for onChangeOfDataIdentifier eventType.
DcmDspRoeOnDTCStatusChange	0..1	This container contains the eventTypeRecord supported for onDTCStatusChange eventType.

10.2.5.17.4 DcmDspRoeOnChangeOfDataIdentifier

SWS Item	[ECUC_Dcm_00975]
Container Name	DcmDspRoeOnChangeOfDataIdentifier
Parent Container	DcmDspRoeEventProperties
Description	This container contains the eventTypeRecord supported for onChangeOfDataIdentifier eventType.
Configuration Parameters	

Name	DcmDspRoeDidRef [ECUC_Dcm_00979]
Parent Container	DcmDspRoeOnChangeOfDataIdentifier
Description	Reference to a Did which is watched.
Multiplicity	1
Type	Reference to DcmDspDid
Post-Build Variant Value	false

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.5.17.5 DcmDspRoeOnDTCCStatusChange

SWS Item	[ECUC_Dcm_00974]
Container Name	DcmDspRoeOnDTCCStatusChange
Parent Container	DcmDspRoeEventProperties
Description	This container contains the eventTypeRecord supported for onDTCCStatusChange eventType.
Configuration Parameters	

Name	DcmDspRoeDTCCStatusMask [ECUC_Dcm_01109]		
Parent Container	DcmDspRoeOnDTCCStatusChange		
Description	Value of the relevant DTCCStatusMask		
Multiplicity	0..1		
Type	EcuIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.5.17.6 DcmDspRoeEventWindowTime

SWS Item	[ECUC_Dcm_00981]
Container Name	DcmDspRoeEventWindowTime
Parent Container	DcmDspRoe
Description	This container configures the available EventWindowTime in this Ecu. This container contains a sub-set of EventWindowTimes supported by the Dcm, to limit the Ecu resources.

Configuration Parameters

Name	DcmDspRoeEventWindowTime [ECUC_Dcm_00982]		
Parent Container	DcmDspRoeEventWindowTime		
Description	Value of the EventWindowTime		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DCM_ROE_EVENT_WINDOW_CURRENT_AND_FOLLOWING_CYCLE		
	DCM_ROE_EVENT_WINDOW_CURRENT_CYCLE		
	DCM_ROE_EVENT_WINDOW_INFINITE		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.5.18 Routines

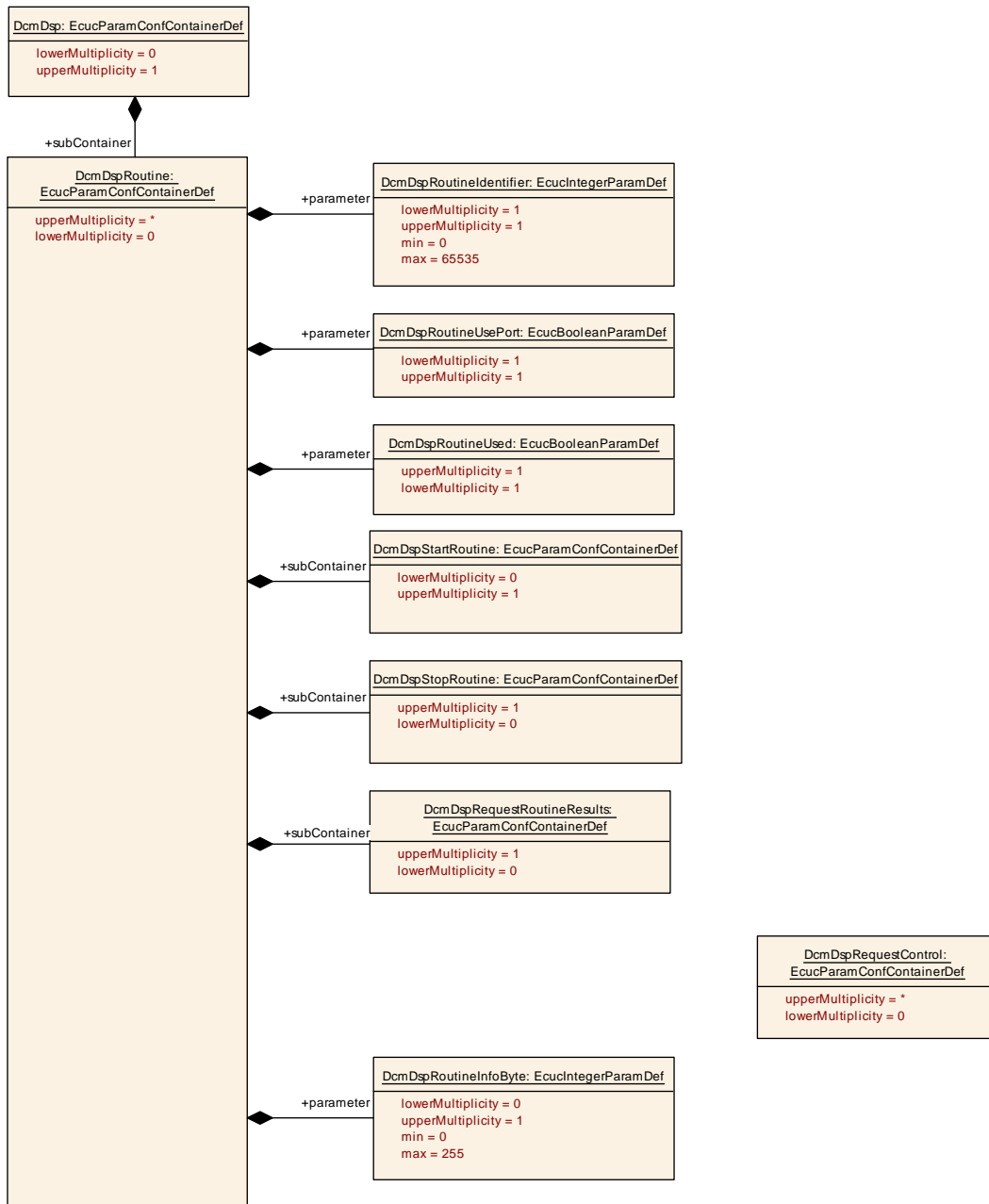


Figure 10.45: DcmDspRoutine configuration overview

10.2.5.18.1 DcmDspRoutine

SWS Item	[ECUC_Dcm_00640]
Container Name	DcmDspRoutine
Parent Container	DcmDsp
Description	This container contains the configuration (parameters) for Routines
Configuration Parameters	

Name	DcmDspRoutineIdentifier [ECUC_Dcm_00641]		
Parent Container	DcmDspRoutine		
Description	2 bytes Identifier of the RID Within each DcmConfigSet all DcmDspRoutineIdentifier values shall be unique.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineInfoByte [ECUC_Dcm_01063]		
Parent Container	DcmDspRoutine		
Description	Manufacturer specific value reported to the tester for the record identifiers 0xE000 to 0xE1FF. (OBD use cases)		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineUsed [ECUC_Dcm_00807]		
Parent Container	DcmDspRoutine		
Description	Allow to activate or deactivate the usage of a Routine, for multi purpose ECUs True = Routine is available False = Routine is not available		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineUsePort [ECUC_Dcm_00724]		
Parent Container	DcmDspRoutine		
Description	<p>If this parameter is set to true, the DCM uses a port requiring a PortInterface RoutineServices_{RoutineName}.</p> <p>The R-Port is named RoutineServices_{RoutineName} where {RoutineName} is the name of the container DcmDspRoutine In that case, the configuration must not provide function names in DcmDspStartRoutineFnc, DcmDspStopRoutineFnc or DcmDspRequestResultsRoutineFnc. If this is false, the DCM expects to find the names of the functions to be used in DcmDspStartRoutineFnc, DcmDspStopRoutineFnc or DcmDspRequestResultsRoutineFnc.</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspRequestRoutineResults	0..1	Provides the configuration of RequestResult subservice for RoutineControl service. Existence indicates that the RequestRoutineResults in the RoutineControl is supported.
DcmDspStartRoutine	0..1	Provides the configuration of Start subservice for RoutineControl service.
DcmDspStopRoutine	0..1	Provides the configuration of Stop subservice for RoutineControl service. Existence indicates that the StopRoutine in the RoutineControl is supported.

10.2.5.18.2 DcmDspRequestRoutineResults

SWS Item	[ECUC_Dcm_01023]
Container Name	DcmDspRequestRoutineResults
Parent Container	DcmDspRoutine
Description	Provides the configuration of RequestResult subservice for RoutineControl service. Existence indicates that the RequestRoutineResults in the RoutineControl is supported.
Configuration Parameters	

Name	DcmDspRequestRoutineResultsConfirmationEnabled [ECUC_Dcm_01091]		
Parent Container	DcmDspRequestRoutineResults		
Description	Allows to enable/disable the confirmation function to indicate the transmission of a response to a RequestRoutineResults request		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRequestRoutineResultsConfirmationFnc [ECUC_Dcm_01090]		
Parent Container	DcmDspRequestRoutineResults		
Description	C-function to call if a transmission confirmation is needed by the issuer (BSW module)		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRequestRoutineResultsFnc [ECUC_Dcm_00753]		
Parent Container	DcmDspRequestRoutineResults		
Description	Function name for request to application the results of a routine. (Routine_RequestResults-function) This parameter is related to the interface Xxx_RequestResults.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRequestRoutineResultsRole [ECUC_Dcm_01146]		
Parent Container	DcmDspRequestRoutineResults		
Description	Bitfield where each bit represents one dedicated role. A RoutineControl with sub-function requestResults is granted access if the bit value is 1. If a bit value is 0, the routine is not allowed for that role.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineInterfaceArgumentIntegrity [ECUC_Dcm_01184]		
Parent Container	DcmDspRequestRoutineResults		
Description	Defines the value of ClientServerOperation.diagArgIntegrity for the created C/S interface of this routine		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRequestRoutineResultsCommonAuthorizationRef [ECUC_Dcm_01054]		
Parent Container	DcmDspRequestRoutineResults		
Description	Reference to DcmDspCommonAuthorization Common authorization configuration taken from the referenced DcmDspRequestRoutineResultsCommonAuthorizationRef. If there is no reference, no check on the commonly defined authorization conditions shall be done to get the routine result.		
Multiplicity	0..1		
Type	Reference to DcmDspCommonAuthorization		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspRequestRoutineResultsIn	0..1	Provide description of input parameter of RequestResult subservice for RoutineControl service.

DcmDspRequestRoutineResultsOut	0..1	Provide description of output parameter of RequestResult subservice for RoutineControl service.
--	------	---

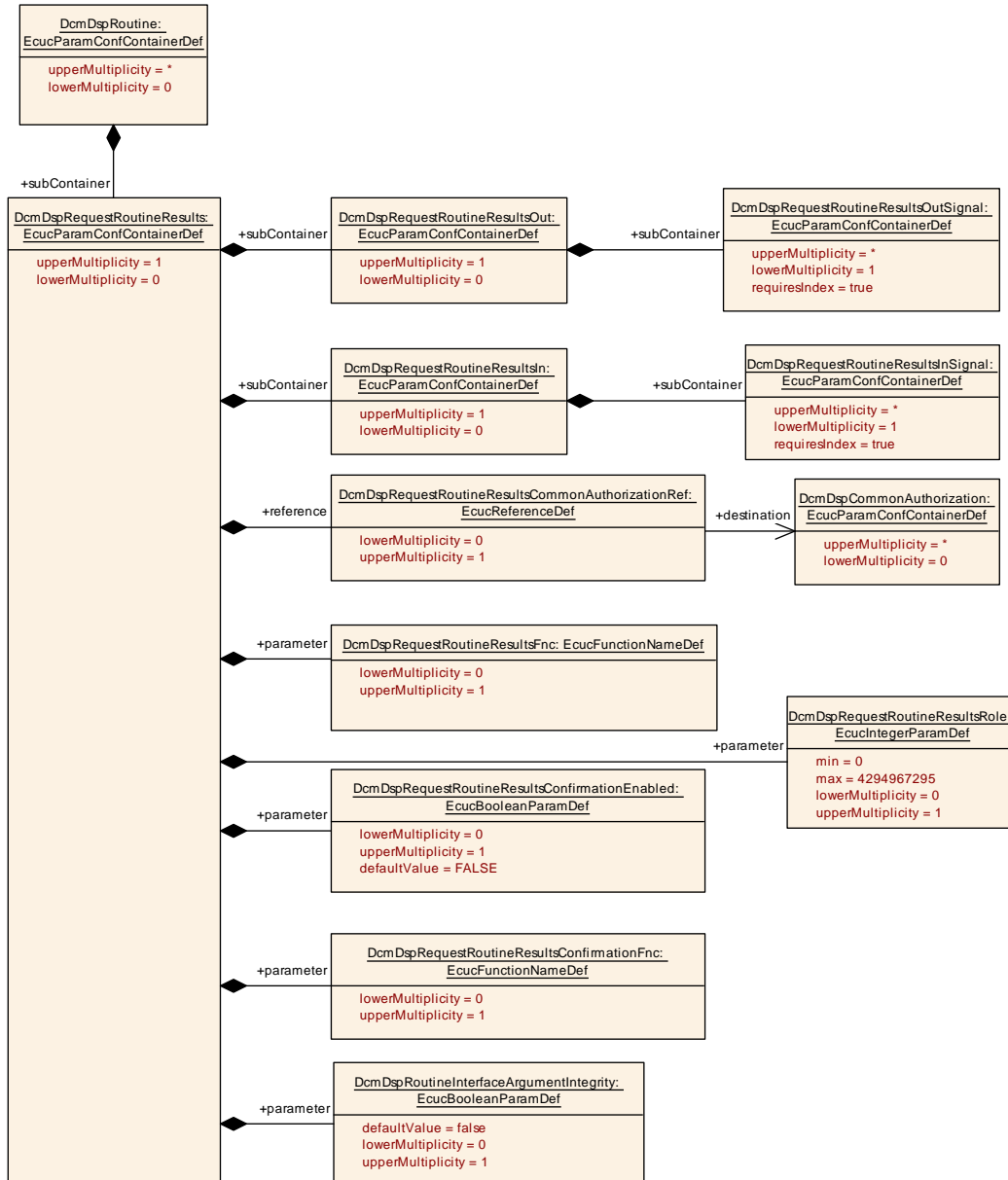


Figure 10.46: DcmDspRequestRoutineResults configuration overview

10.2.5.18.3 DcmDspRequestRoutineResultsIn

SWS Item	[ECUC_Dcm_01116]
Container Name	DcmDspRequestRoutineResultsIn
Parent Container	DcmDspRequestRoutineResults
Description	Provide description of input parameter of RequestResult subservice for RoutineControl service.

Configuration Parameters

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspRequestRoutineResultsInSignal	1..*	Provides description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataIn_n elements in the XXX_RequestResult function call.

10.2.5.18.4 DcmDspRequestRoutineResultsInSignal

SWS Item	[ECUC_Dcm_01117]
Container Name	DcmDspRequestRoutineResultsInSignal
Parent Container	DcmDspRequestRoutineResultsIn
Description	Provides description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataIn_n elements in the XXX_RequestResult function call. Attributes: requiresIndex=true
Configuration Parameters	

Name	DcmDspRoutineParameterSize [ECUC_Dcm_01119]		
Parent Container	DcmDspRequestRoutineResultsInSignal		
Description	Provide the size of a RoutineControl parameter in bytes		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalEndianness [ECUC_Dcm_01121]		
Parent Container	DcmDspRequestRoutineResultsInSignal		
Description	Defines the endianness of the data belonging to a Routine In Signal for RequestResult subfunction.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	BIG_ENDIAN		Most significant byte shall be stored at the lowest address.
	LITTLE_ENDIAN		Most significant byte shall be stored at the highest address
	OPAQUE	false	Opaque data endianness
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalPos [ECUC_Dcm_01118]		
Parent Container	DcmDspRequestRoutineResultsInSignal		
Description	Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalType [ECUC_Dcm_01120]		
Parent Container	DcmDspRequestRoutineResultsInSignal		
Description	Provide the type of the signal in the RoutineControl request/response.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BOOLEAN		Type of the signal is boolean.
	FLOAT		Type of the data is float.
	FLOAT_N		Type of the data is float array.
	SINT16		Type of the signal is sint16.

Post-Build Variant Value	SINT16_N	Type of the signal is sint16 array.	
	SINT32	Type of the signal is sint32.	
	SINT32_N	Type of the signal is sint32 array.	
	SINT8	Type of the signal is sint8.	
	SINT8_N	Type of the signal is sint8 array.	
	UINT16	Type of the signal is uint16.	
	UINT16_N	Type of the signal is uint16 array.	
	UINT32	Type of the signal is uint32.	
	UINT32_N	Type of the signal is uint32 array.	
	UINT8	Type of the signal is uint8.	
	UINT8_N	Type of the signal is uint8 array.	
	VARIABLE_LENGTH	Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspArgumentScaling	0..1	This container contains the configuration (arguments) of an alternative Diagnosis Representation. Out if this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.

10.2.5.18.5 DcmDspRequestRoutineResultsOut

SWS Item	[ECUC_Dcm_00831]
Container Name	DcmDspRequestRoutineResultsOut
Parent Container	DcmDspRequestRoutineResults
Description	Provide description of output parameter of RequestResult subservice for RoutineControl service.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspRequestRoutineResultsOutSignal	1..*	Provides description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_RequestResult function call.

10.2.5.18.6 DcmDspRequestRoutineResultsOutSignal

SWS Item	[ECUC_Dcm_00836]
Container Name	DcmDspRequestRoutineResultsOutSignal
Parent Container	DcmDspRequestRoutineResultsOut
Description	<p>Provides description of a routine signal used in RoutineControl service.</p> <p>The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_RequestResult function call.</p> <p>Attributes: requiresIndex=true</p>
Configuration Parameters	

Name	DcmDspRoutineParameterSize [ECUC_Dcm_00838]		
Parent Container	DcmDspRequestRoutineResultsOutSignal		
Description	Provide the size of a RoutineControl parameter in bytes		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalEndianness [ECUC_Dcm_01013]	
Parent Container	DcmDspRequestRoutineResultsOutSignal	
Description	Defines the endianness of the data belonging to a Routine Out Signal for RequestResult subfunction.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	BIG_ENDIAN	Most significant byte shall be stored at the lowest address.
	LITTLE_ENDIAN	Most significant byte shall be stored at the highest address
	OPAQUE	Opaque data endianness
Post-Build Variant Multiplicity	false	

Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalPos [ECUC_Dcm_00837]		
Parent Container	DcmDspRequestRoutineResultsOutSignal		
Description	Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalType [ECUC_Dcm_00881]	
Parent Container	DcmDspRequestRoutineResultsOutSignal	
Description	Provide the type of the signal in the RoutineControl request/response.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BOOLEAN	Type of the signal is boolean.
	FLOAT	Type of the data is float.
	FLOAT_N	Type of the data is float array.
	SINT16	Type of the signal is sint16.
	SINT16_N	Type of the signal is sint16 array.
	SINT32	Type of the signal is sint32.
	SINT32_N	Type of the signal is sint32 array.
	SINT8	Type of the signal is sint8.
	SINT8_N	Type of the signal is sint8 array.
	UINT16	Type of the signal is uint16.
	UINT16_N	Type of the signal is uint16 array.
	UINT32	Type of the signal is uint32.
	UINT32_N	Type of the signal is uint32 array.
	UINT8	Type of the signal is uint8.
	UINT8_N	Type of the signal is uint8 array.

Post-Build Variant Value	VARIABLE_LENGTH false	Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspArgument Scaling	0..1	This container contains the configuration (arguments) of an alternative Diagnosis Representation. Out if this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.

10.2.5.18.7 DcmDspStartRoutine

SWS Item	[ECUC_Dcm_01021]
Container Name	DcmDspStartRoutine
Parent Container	DcmDspRoutine
Description	Provides the configuration of Start subservice for RoutineControl service.
Configuration Parameters	

Name	DcmDspRoutineInterfaceArgumentIntegrity [ECUC_Dcm_01184]		
Parent Container	DcmDspStartRoutine		
Description	Defines the value of ClientServerOperation.diagArgIntegrity for the created C/S interface of this routine		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Scope / Dependency	scope: ECU
---------------------------	------------

Name	DcmDspStartRoutineConfirmationEnabled [ECUC_Dcm_01093]		
Parent Container	DcmDspStartRoutine		
Description	Allows to enable/disable the confirmation function to indicate the transmission of a response to a StartRoutine request		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspStartRoutineConfirmationFnc [ECUC_Dcm_01094]		
Parent Container	DcmDspStartRoutine		
Description	C-function to call if a transmission confirmation is needed by the issuer (BSW module)		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspStartRoutineFnc [ECUC_Dcm_00664]		
Parent Container	DcmDspStartRoutine		
Description	Function name for request to application to start a routine. (Routine_Start-function) This parameter is related to the interface Xxx_Start.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspStartRoutineRole [ECUC_Dcm_01144]		
Parent Container	DcmDspStartRoutine		
Description	Bitfield were each bit represents one dedicated role. A RoutineControl with sub-function startRoutine is granted access if the bit value is 1. If a bit value is 0, the routine is not allowed for that role.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	DcmDspStartRoutineCommonAuthorizationRef [ECUC_Dcm_01052]		
Parent Container	DcmDspStartRoutine		
Description	Reference to DcmDspCommonAuthorization Common authorization configuration taken from the referenced DcmDspStartRoutineCommonAuthorizationRef. If there is no reference, no check on the commonly defined authorization conditions shall be done to start the routine.		
Multiplicity	0..1		
Type	Reference to DcmDspCommonAuthorization		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspStartRoutineIn	0..1	Provide description of input parameter of Start subservice for RoutineControl service
DcmDspStartRoutineOut	0..1	Provide description of output parameter of Start subservice for RoutineControl service.

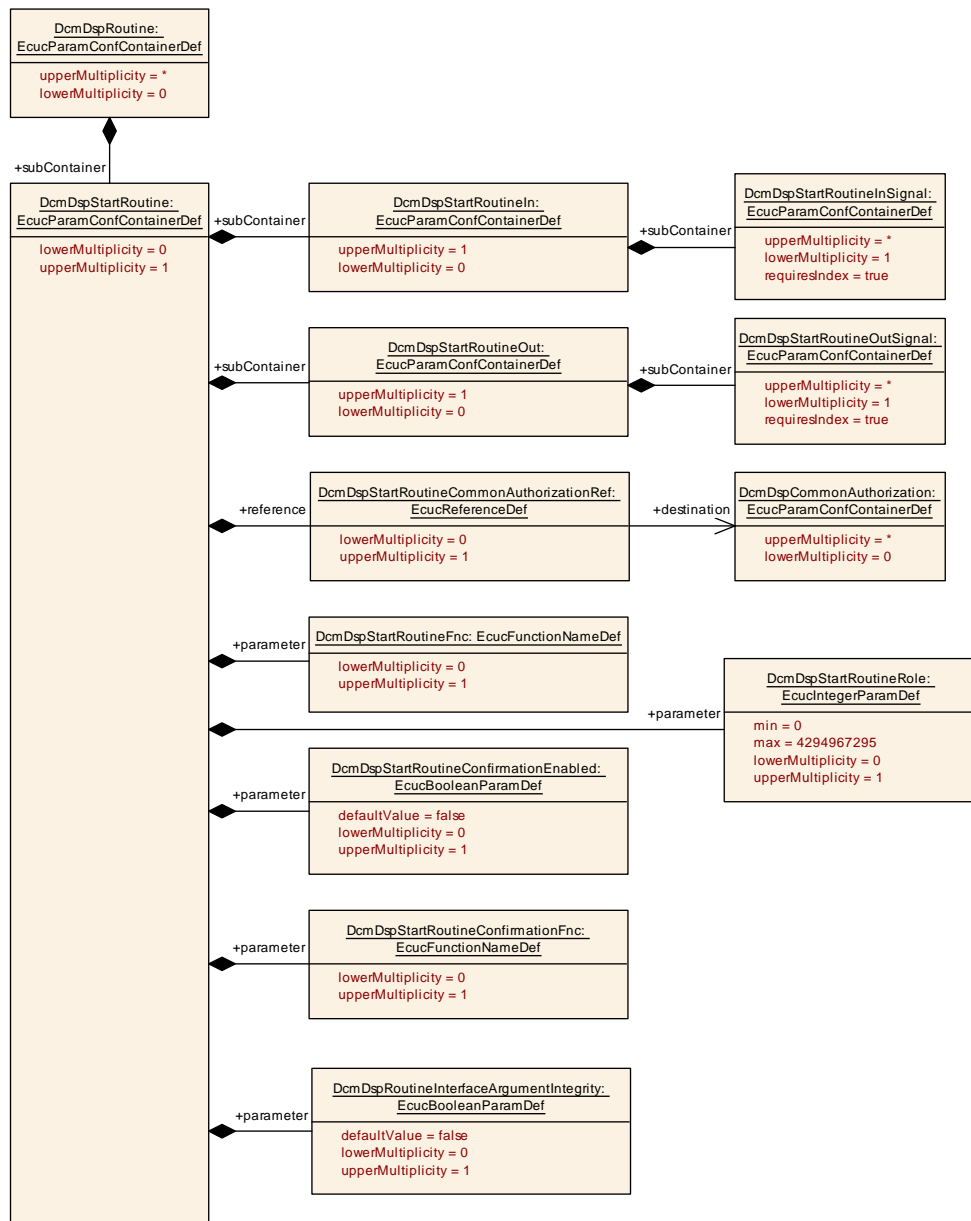


Figure 10.47: DcmDspStartRoutine configuration overview

10.2.5.18.8 DcmDspStartRoutineIn

SWS Item	[ECUC_Dcm_00834]
Container Name	DcmDspStartRoutineIn
Parent Container	DcmDspStartRoutine
Description	Provide description of input parameter of Start subservice for RoutineControl service
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspStartRoutineInSignal	1..*	<p>Provide description of a routine signal used in RoutineControl service.</p> <p>The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataInN elements in the XXX_Start function call.</p>

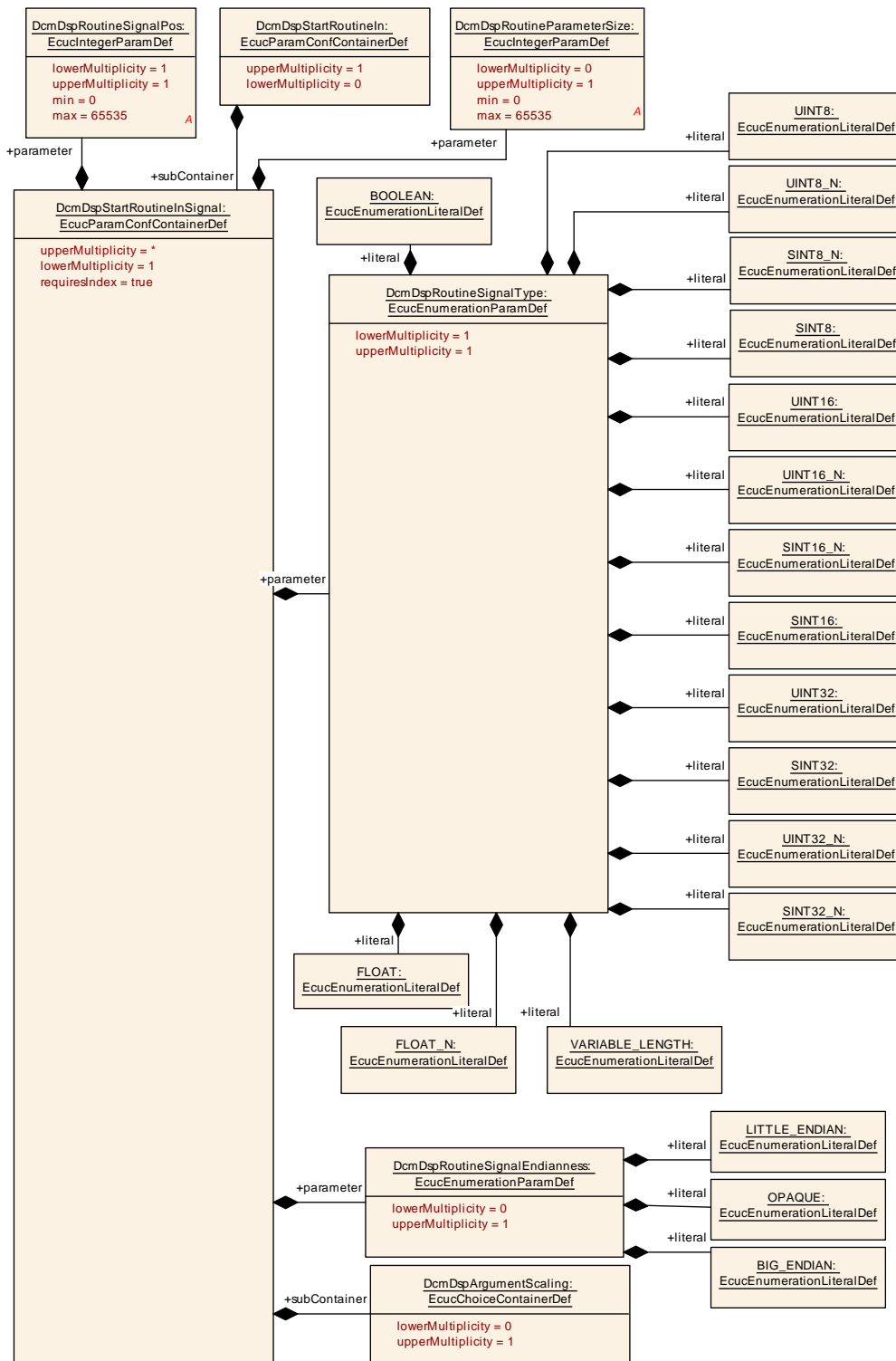


Figure 10.48: DcmDspStartRoutineIn configuration overview

10.2.5.18.9 DcmDspStartRoutineInSignal

SWS Item	[ECUC_Dcm_00845]
----------	------------------

Container Name	DcmDspStartRoutineInSignal
Parent Container	DcmDspStartRoutineIn
Description	<p>Provide description of a routine signal used in RoutineControl service.</p> <p>The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataInN elements in the XXX_Start function call.</p> <p>Attributes: requiresIndex=true</p>
Configuration Parameters	

Name	DcmDspRoutineParameterSize [ECUC_Dcm_00847]		
Parent Container	DcmDspStartRoutineInSignal		
Description	Provide the size of a RoutineControl parameter in bytes		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalEndianness [ECUC_Dcm_01016]	
Parent Container	DcmDspStartRoutineInSignal	
Description	Defines the endianness of the data belonging to a Routine In Signal for Start subfunction.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	BIG_ENDIAN	Most significant byte shall be stored at the lowest address.
	LITTLE_ENDIAN	Most significant byte shall be stored at the highest address
	OPAQUE	Opaque data endianness
Post-Build Variant Multiplicity	false	
Post-Build Variant Value	false	

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalPos [ECUC_Dcm_00846]		
Parent Container	DcmDspStartRoutineInSignal		
Description	Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalType [ECUC_Dcm_00884]	
Parent Container	DcmDspStartRoutineInSignal	
Description	Provide the type of the signal in the RoutineControl request/response.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BOOLEAN	Type of the signal is boolean.
	FLOAT	Type of the data is float.
	FLOAT_N	Type of the data is float array.
	SINT16	Type of the signal is sint16.
	SINT16_N	Type of the signal is sint16 array.
	SINT32	Type of the signal is sint32.
	SINT32_N	Type of the signal is sint32 array.
	SINT8	Type of the signal is sint8.
	SINT8_N	Type of the signal is sint8 array.
	UINT16	Type of the signal is uint16.
	UINT16_N	Type of the signal is uint16 array.
	UINT32	Type of the signal is uint32.
	UINT32_N	Type of the signal is uint32 array.
	UINT8	Type of the signal is uint8.
	UINT8_N	Type of the signal is uint8 array.

Post-Build Variant Value	VARIABLE_LENGTH false	Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspArgument Scaling	0..1	This container contains the configuration (arguments) of an alternative Diagnosis Representation. Out if this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.

10.2.5.18.10 DcmDspStartRoutineOut

SWS Item	[ECUC_Dcm_00835]
Container Name	DcmDspStartRoutineOut
Parent Container	DcmDspStartRoutine
Description	Provide description of output parameter of Start subservice for RoutineControl service.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspStartRoutineOut Signal	1..*	Provide description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_Start function call.

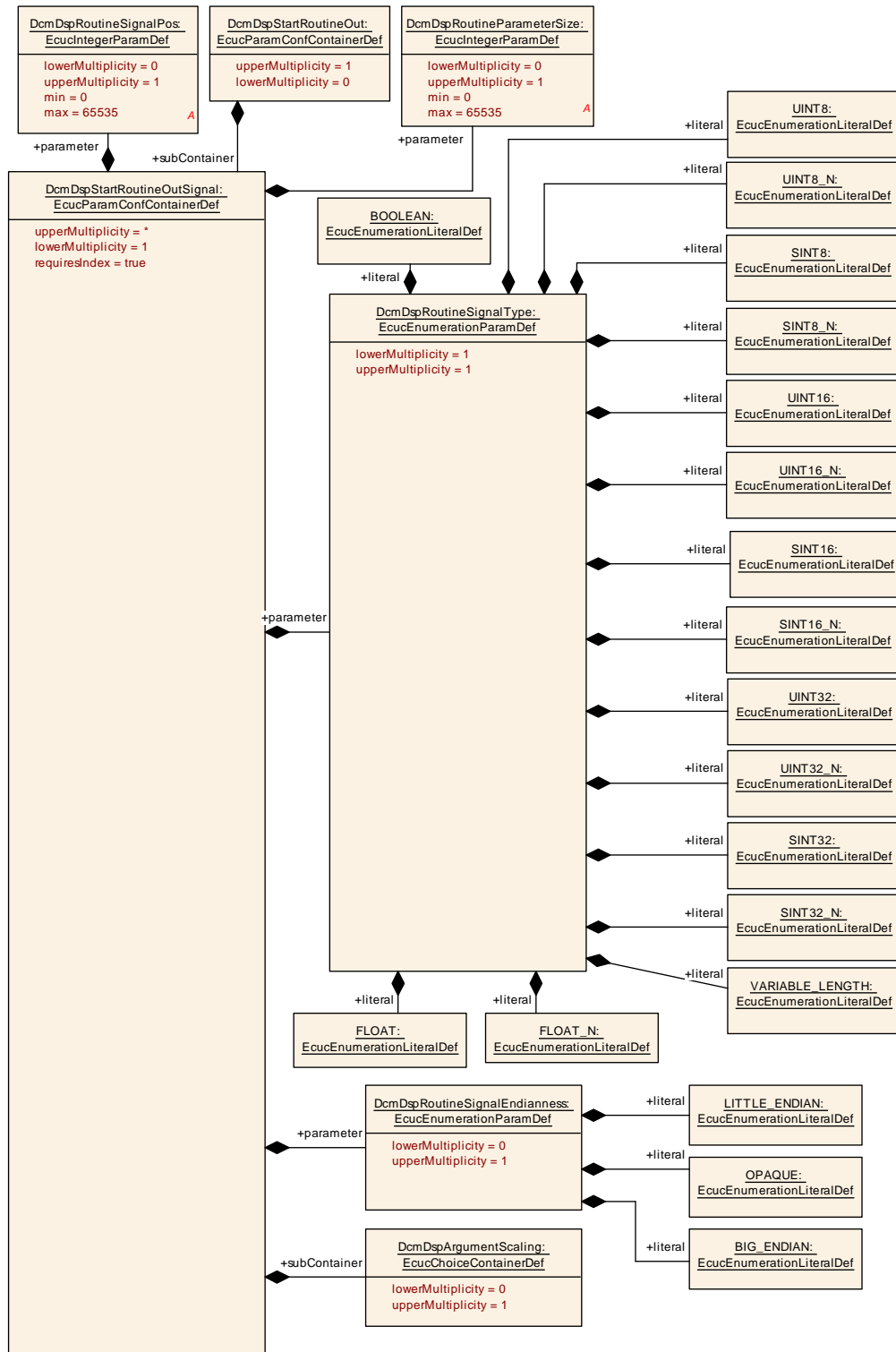


Figure 10.49: DcmDspStartRoutineOut configuration overview

10.2.5.18.11 DcmDspStartRoutineOutSignal

SWS Item	[ECUC_Dcm_00848]
----------	------------------

Container Name	DcmDspStartRoutineOutSignal
Parent Container	DcmDspStartRoutineOut
Description	<p>Provide description of a routine signal used in RoutineControl service.</p> <p>The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_Start function call.</p> <p>Attributes: requiresIndex=true</p>
Configuration Parameters	

Name	DcmDspRoutineParameterSize [ECUC_Dcm_00850]		
Parent Container	DcmDspStartRoutineOutSignal		
Description	Provide the size of a RoutineControl parameter in bytes		
Multiplicity	0..1		
Type	EcuIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalEndianness [ECUC_Dcm_01017]	
Parent Container	DcmDspStartRoutineOutSignal	
Description	Defines the endianness of the data belonging to a Routine Out Signal for Start subfunction.	
Multiplicity	0..1	
Type	EcuEnumerationParamDef	
Range	BIG_ENDIAN	Most significant byte shall be stored at the lowest address.
	LITTLE_ENDIAN	Most significant byte shall be stored at the highest address
	OPAQUE	Opaque data endianness
Post-Build Variant Multiplicity	false	
Post-Build Variant Value	false	

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalPos [ECUC_Dcm_00867]		
Parent Container	DcmDspStartRoutineOutSignal		
Description	Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalType [ECUC_Dcm_00885]	
Parent Container	DcmDspStartRoutineOutSignal	
Description	Provide the type of the signal in the RoutineControl request/response.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BOOLEAN	Type of the signal is boolean.
	FLOAT	Type of the data is float.
	FLOAT_N	Type of the data is float array.
	SINT16	Type of the signal is sint16.
	SINT16_N	Type of the signal is sint16 array.
	SINT32	Type of the signal is sint32.
	SINT32_N	Type of the signal is sint32 array.
	SINT8	Type of the signal is sint8.
	SINT8_N	Type of the signal is sint8 array.
	UINT16	Type of the signal is uint16.
	UINT16_N	Type of the signal is uint16 array.
	UINT32	Type of the signal is uint32.

	UINT32_N	Type of the signal is uint32 array.	
	UINT8	Type of the signal is uint8.	
	UINT8_N	Type of the signal is uint8 array.	
	VARIABLE_LENGTH	Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspArgumentScaling	0..1	This container contains the configuration (arguments) of an alternative Diagnosis Representation. Out if this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.

10.2.5.18.12 DcmDspStopRoutine

SWS Item	[ECUC_Dcm_01022]
Container Name	DcmDspStopRoutine
Parent Container	DcmDspRoutine
Description	Provides the configuration of Stop subservice for RoutineControl service. Existence indicates that the StopRoutine in the RoutineControl is supported.
Configuration Parameters	

Name	DcmDspRoutineInterfaceArgumentIntegrity [ECUC_Dcm_01184]		
Parent Container	DcmDspStopRoutine		
Description	Defines the value of ClientServerOperation.diagArgIntegrity for the created C/S interface of this routine		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspStopRoutineConfirmationEnabled [ECUC_Dcm_01095]		
Parent Container	DcmDspStopRoutine		
Description	Allows to enable/disable the confirmation function to indicate the transmission of a response to a StopRoutine request		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspStopRoutineConfirmationFnc [ECUC_Dcm_01096]		
Parent Container	DcmDspStopRoutine		
Description	C-function to call if a transmission confirmation is needed by the issuer (BSW module)		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspStopRoutineFnc [ECUC_Dcm_00752]		
Parent Container	DcmDspStopRoutine		
Description	Function name for request to application to stop a routine. (Routine_Stop-function) This parameter is related to the interface Xxx_Stop.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspStopRoutineRole [ECUC_Dcm_01145]		
Parent Container	DcmDspStopRoutine		
Description	Bitfield were each bit represents one dedicated role. A RoutineControl with sub-function stopRoutine is granted access if the bit value is 1. If a bit value is 0, the routine is not allowed for that role.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	DcmDspStopRoutineCommonAuthorizationRef [ECUC_Dcm_01053]		
Parent Container	DcmDspStopRoutine		
Description	Reference to DcmDspCommonAuthorization Common authorization configuration taken from the referenced DcmDspStopRoutineCommonAuthorizationRef. If there is no reference, no check on the commonly defined authorization conditions shall be done to stop the routine.		
Multiplicity	0..1		
Type	Reference to DcmDspCommonAuthorization		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspStopRoutineIn	0..1	Provide description of input parameter of Stop subservice for RoutineControl service.
DcmDspStopRoutineOut	0..1	Provide description of output parameter of Stop subservice for RoutineControl service.

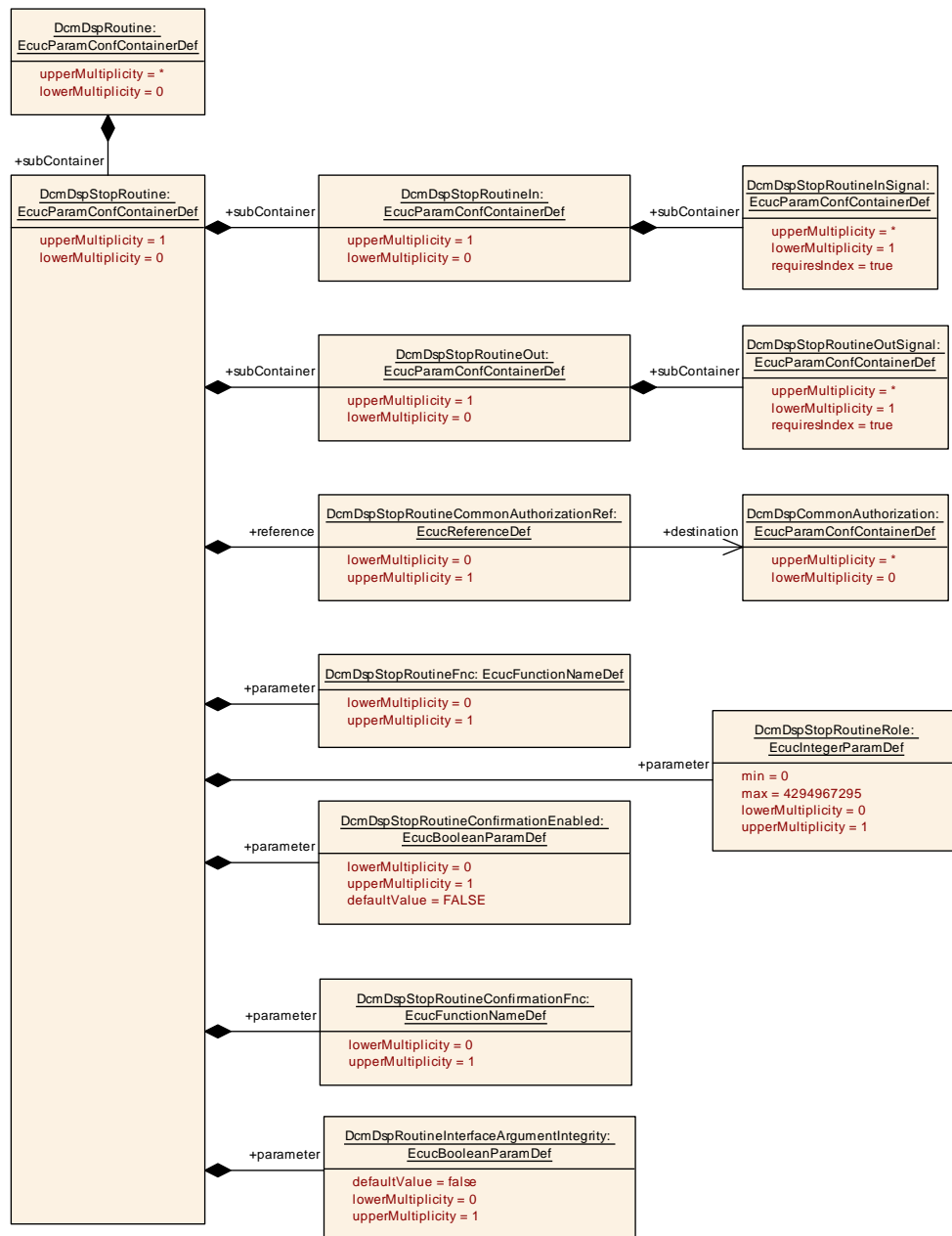


Figure 10.50: DcmDspStopRoutine configuration overview

10.2.5.18.13 DcmDspStopRoutineIn

SWS Item	[ECUC_Dcm_00832]
Container Name	DcmDspStopRoutineIn
Parent Container	DcmDspStopRoutine
Description	Provide description of input parameter of Stop subservice for RoutineControl service.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspStopRoutineInSignal	1..*	<p>Provide description of a routine signal used in RoutineControl service.</p> <p>The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataInN elements in the XXX_Stop function call.</p>

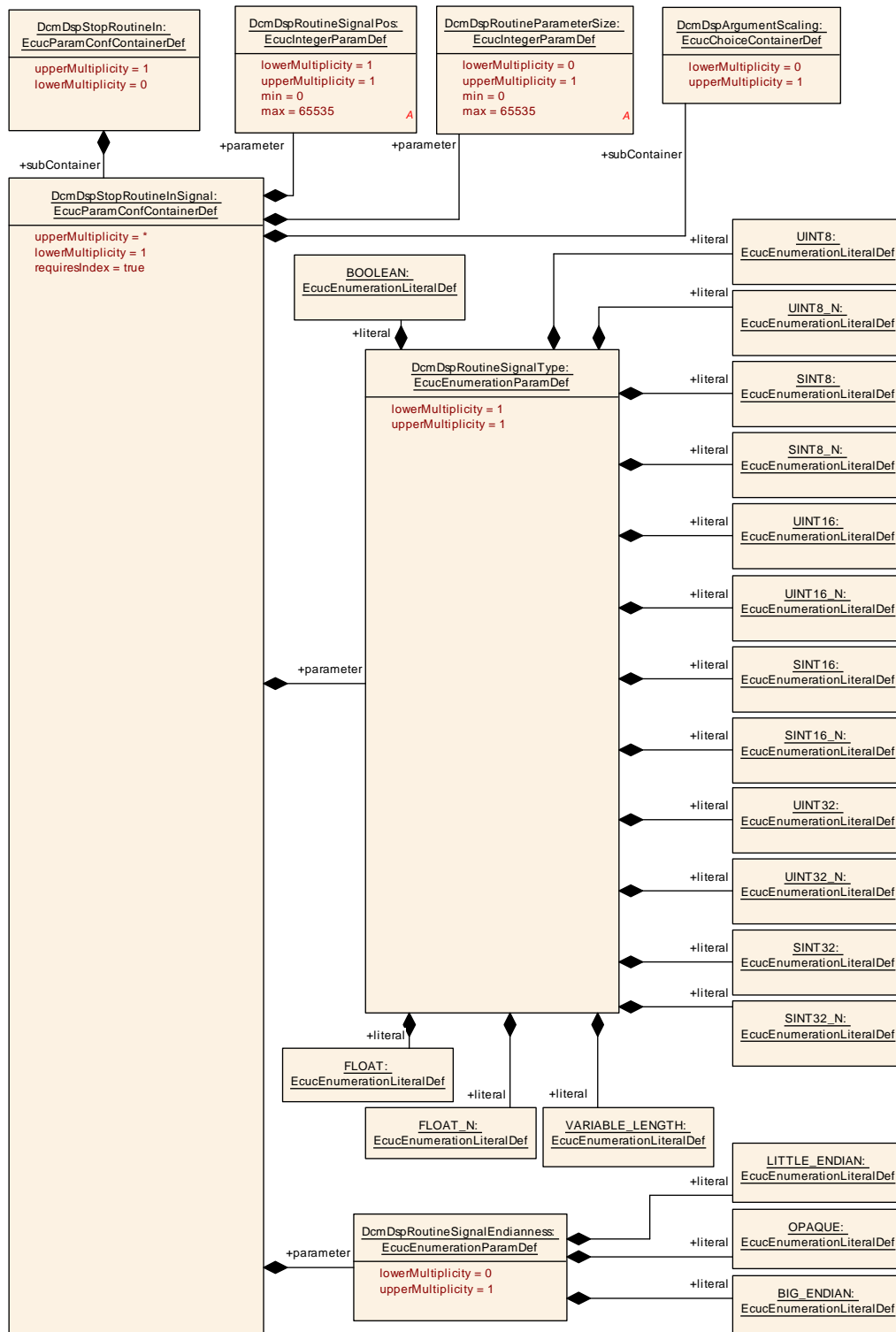


Figure 10.51: DcmDspRoutineStopIn configuration overview

10.2.5.18.14 DcmDspStopRoutineInSignal

SWS Item	[ECUC_Dcm_00839]
----------	------------------

Container Name	DcmDspStopRoutineInSignal
Parent Container	DcmDspStopRoutineIn
Description	<p>Provide description of a routine signal used in RoutineControl service.</p> <p>The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataInN elements in the XXX_Stop function call.</p> <p>Attributes: requiresIndex=true</p>
Configuration Parameters	

Name	DcmDspRoutineParameterSize [ECUC_Dcm_00841]		
Parent Container	DcmDspStopRoutineInSignal		
Description	Provide the size of a RoutineControl parameter in bytes		
Multiplicity	0..1		
Type	EcuIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalEndianness [ECUC_Dcm_01014]	
Parent Container	DcmDspStopRoutineInSignal	
Description	Defines the endianness of the data belonging to a Routine In Signal for Stop subfunction.	
Multiplicity	0..1	
Type	EcuEnumerationParamDef	
Range	BIG_ENDIAN	Most significant byte shall be stored at the lowest address.
	LITTLE_ENDIAN	Most significant byte shall be stored at the highest address
	OPAQUE	Opaque data endianness
Post-Build Variant Multiplicity	false	
Post-Build Variant Value	false	

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalPos [ECUC_Dcm_00840]		
Parent Container	DcmDspStopRoutineInSignal		
Description	Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalType [ECUC_Dcm_00882]	
Parent Container	DcmDspStopRoutineInSignal	
Description	Provide the type of the signal in the RoutineControl request/response.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BOOLEAN	Type of the signal is boolean.
	FLOAT	Type of the data is float.
	FLOAT_N	Type of the data is float array.
	SINT16	Type of the signal is sint16.
	SINT16_N	Type of the signal is sint16 array.
	SINT32	Type of the signal is sint32.
	SINT32_N	Type of the signal is sint32 array.
	SINT8	Type of the signal is sint8.
	SINT8_N	Type of the signal is sint8 array.
	UINT16	Type of the signal is uint16.
	UINT16_N	Type of the signal is uint16 array.
	UINT32	Type of the signal is uint32.
	UINT32_N	Type of the signal is uint32 array.
	UINT8	Type of the signal is uint8.
	UINT8_N	Type of the signal is uint8 array.

Post-Build Variant Value	VARIABLE_LENGTH false	Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspArgument Scaling	0..1	This container contains the configuration (arguments) of an alternative Diagnosis Representation. Out if this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.

10.2.5.18.15 DcmDspStopRoutineOut

SWS Item	[ECUC_Dcm_00833]
Container Name	DcmDspStopRoutineOut
Parent Container	DcmDspStopRoutine
Description	Provide description of output parameter of Stop subservice for RoutineControl service.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspStopRoutineOut Signal	1..*	Provide description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_Stop function call.

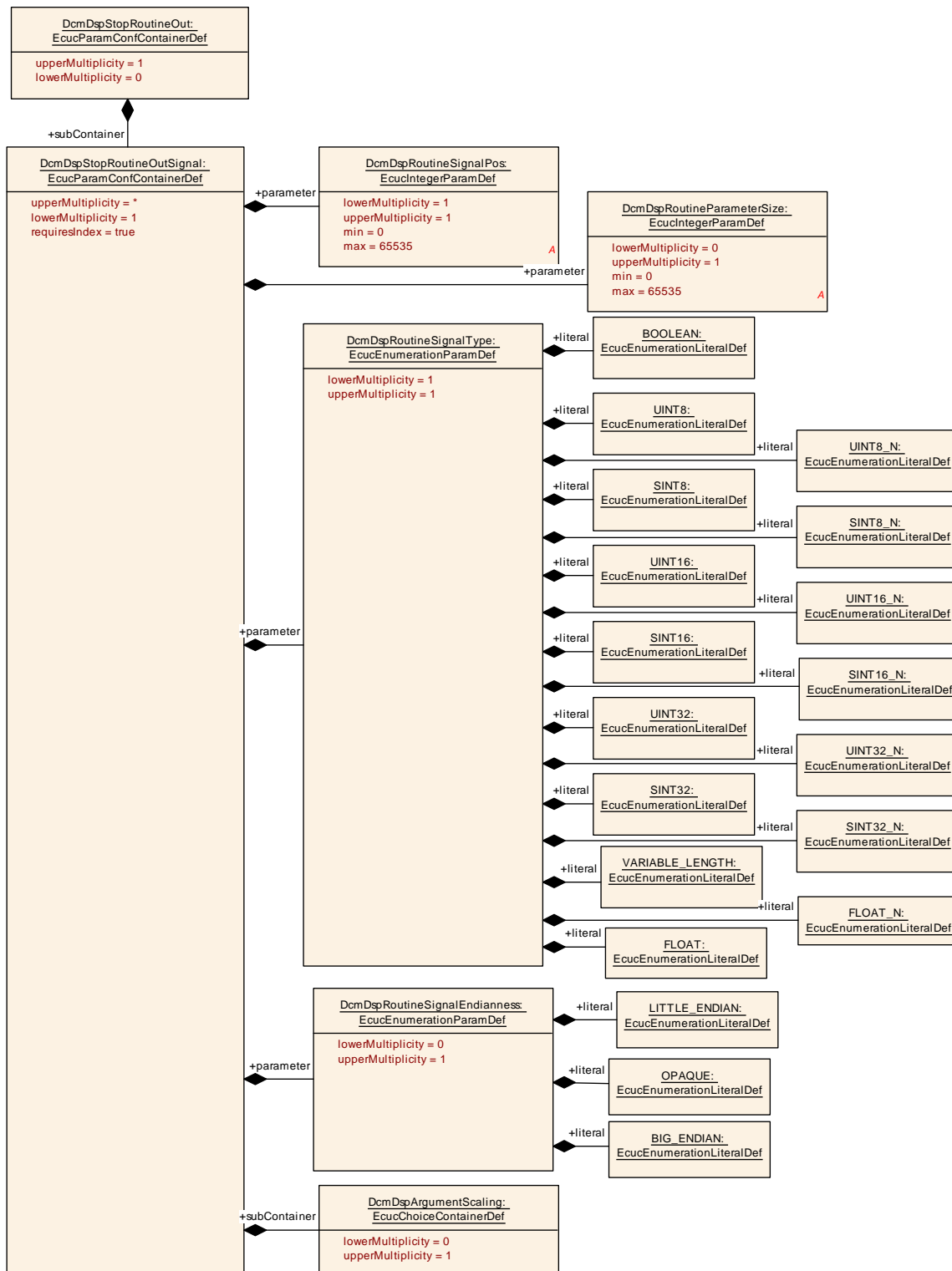


Figure 10.52: DcmDspStopRoutineOut configuration overview

10.2.5.18.16 DcmDspStopRoutineOutSignal

SWS Item	[ECUC_Dcm_00842]
Container Name	DcmDspStopRoutineOutSignal
Parent Container	DcmDspStopRoutineOut
Description	<p>Provide description of a routine signal used in RoutineControl service.</p> <p>The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_Stop function call.</p> <p>Attributes: requiresIndex=true</p>
Configuration Parameters	

Name	DcmDspRoutineParameterSize [ECUC_Dcm_00844]		
Parent Container	DcmDspStopRoutineOutSignal		
Description	Provide the size of a RoutineControl parameter in bytes		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalEndianness [ECUC_Dcm_01015]	
Parent Container	DcmDspStopRoutineOutSignal	
Description	Defines the endianness of the data belonging to a Routine Out Signal for Stop subfunction.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	BIG_ENDIAN	Most significant byte shall be stored at the lowest address.
	LITTLE_ENDIAN	Most significant byte shall be stored at the highest address
	OPAQUE	Opaque data endianness
Post-Build Variant Multiplicity	false	
Post-Build Variant Value	false	

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalPos [ECUC_Dcm_00843]		
Parent Container	DcmDspStopRoutineOutSignal		
Description	Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspRoutineSignalType [ECUC_Dcm_00883]	
Parent Container	DcmDspStopRoutineOutSignal	
Description	Provide the type of the signal in the RoutineControl request/response.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BOOLEAN	Type of the signal is boolean.
	FLOAT	Type of the data is float.
	FLOAT_N	Type of the data is float array.
	SINT16	Type of the signal is sint16.
	SINT16_N	Type of the signal is sint16 array.
	SINT32	Type of the signal is sint32.
	SINT32_N	Type of the signal is sint32 array.
	SINT8	Type of the signal is sint8.
	SINT8_N	Type of the signal is sint8 array.
	UINT16	Type of the signal is uint16.
	UINT16_N	Type of the signal is uint16 array.
	UINT32	Type of the signal is uint32.
	UINT32_N	Type of the signal is uint32 array.
	UINT8	Type of the signal is uint8.
	UINT8_N	Type of the signal is uint8 array.

Post-Build Variant Value	VARIABLE_LENGTH false	Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspArgument Scaling	0..1	This container contains the configuration (arguments) of an alternative Diagnosis Representation. Out if this the scaling between Diagnosis and ECU internal representation and vice versa can be calculated.

10.2.5.19 Session Security and Modes

10.2.5.19.1 DcmDspSecurity

SWS Item	[ECUC_Dcm_00764]
Container Name	DcmDspSecurity
Parent Container	DcmDsp
Description	This container contains the configuration (DSP parameter) for security level configuration (per security level) Description This container contains Rows of DcmDspSecurityRow
Configuration Parameters	

Name	DcmDspSecurityMaxAttemptCounterReadoutTime [ECUC_Dcm_01101]	
Parent Container	DcmDspSecurity	
Description	Delay, in seconds, from startup (measured from the first call of the Dcm_MainFunction()), allowed for all AttemptCounter values to be obtained from the Application. Must be a multiple of the DcmTaskTime. min: A value equal to the DcmTaskTime	
Multiplicity	1	
Type	EcucFloatParamDef	
Range]0 .. 65535[
Default Value		
Post-Build Variant Value	false	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspSecurityRow	0..31	Definition of a single Row of configuration for security level configuration (per security level) The name of this container is used to define the name of the R-Port through which the DCM accesses the interface SecurityAccess_{SecurityLevel}. The R-Port is named SecurityAccess_{SecurityLevel} where {SecurityLevel} is the name of the container DcmDspSecurityRow. If there is no reference, no check of security level shall be done.

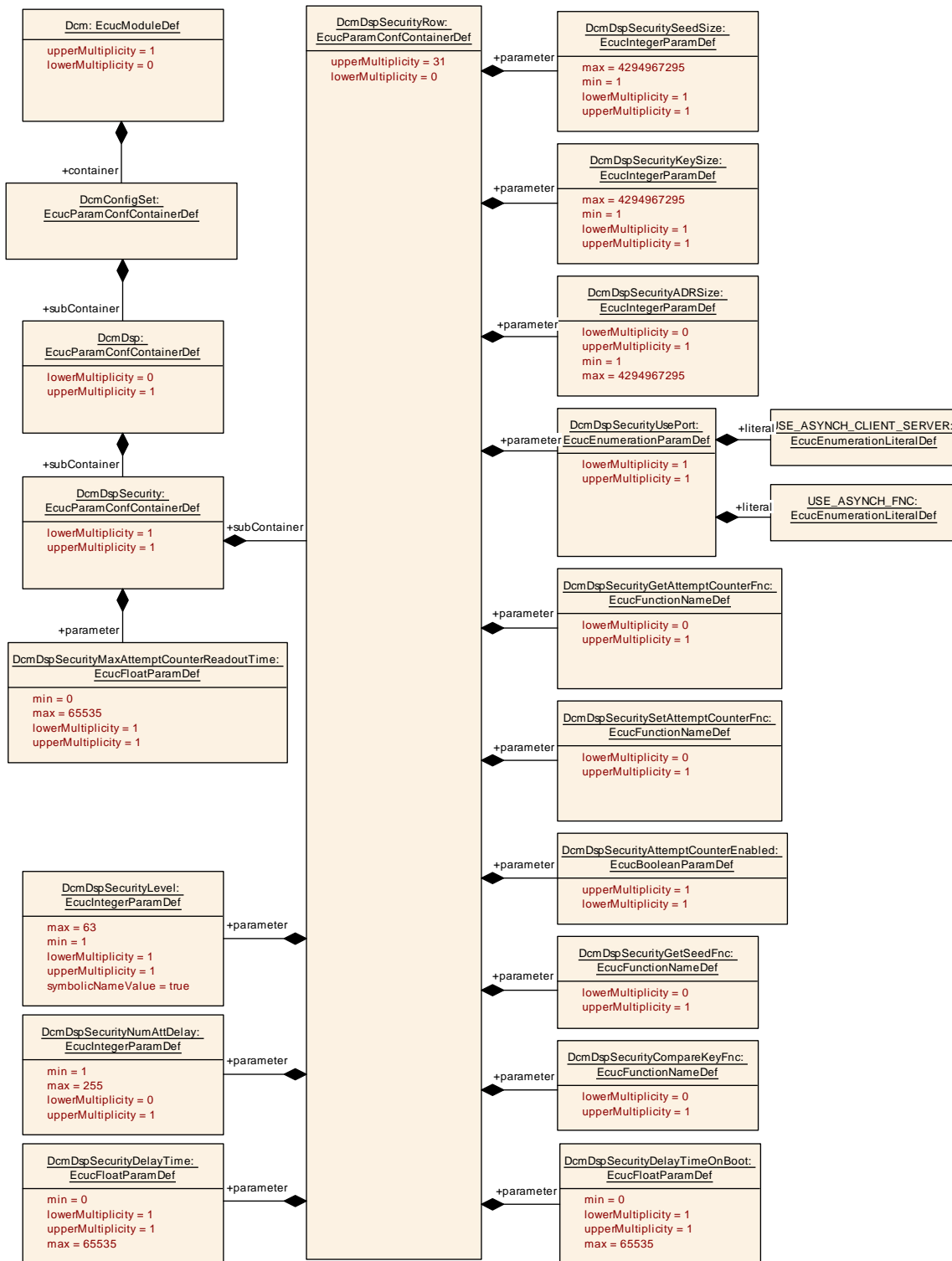


Figure 10.53: DcmDspSecurity configuration overview

10.2.5.19.2 DcmDspSecurityRow

SWS Item	[ECUC_Dcm_00759]
----------	------------------

Container Name	DcmDspSecurityRow
Parent Container	DcmDspSecurity
Description	Definition of a single Row of configuration for security level configuration (per security level) The name of this container is used to define the name of the R-Port through which the DCM accesses the interface SecurityAccess_{SecurityLevel}. The R-Port is named SecurityAccess_{SecurityLevel} where {SecurityLevel} is the name of the container DcmDspSecurityRow. If there is no reference, no check of security level shall be done.
Configuration Parameters	

Name	DcmDspSecurityADRSIZE [ECUC_Dcm_00725]		
Parent Container	DcmDspSecurityRow		
Description	Size in bytes of the AccessDataRecord used in GetSeed		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspSecurityAttemptCounterEnabled [ECUC_Dcm_01050]		
Parent Container	DcmDspSecurityRow		
Description	This configuration parameter controls the existence of the APIs to set / get the attempt counter values towards application (Xxx_SetSecurityAttemptCounter() / Xxx_GetSecurityAttemptCounter()). In case of enabled, the security attempt counter values are passed to application, whenever there is a change in the value. This allows storing the values in nonvolatile RAM and restoring them at ECU startup.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	

Scope / Dependency	scope: ECU
---------------------------	------------

Name	DcmDspSecurityCompareKeyFnc [ECUC_Dcm_00969]		
Parent Container	DcmDspSecurityRow		
Description	Function name to request the result of a key comparison. Parameter is only relevant if DcmDspSecurityUsePort=="USE_ ASYNCH_FNC". This parameter is related to the interface Xxx_CompareKey.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspSecurityDelayTime [ECUC_Dcm_00757]		
Parent Container	DcmDspSecurityRow		
Description	Delay time after failed security access in seconds. This is started after DcmDspSecurityNumAttDelay number of failed security accesses. min: A negative value is not allowed.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. 65535]		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspSecurityDelayTimeOnBoot [ECUC_Dcm_00726]		
Parent Container	DcmDspSecurityRow		
Description	Value of the delay timer in case of 'power on' in seconds. This delay indicates the time at ECU boot power-on time during which the Dcm does not accept a security access. min: A negative value is not allowed.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. 65535]		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspSecurityGetAttemptCounterFnc [ECUC_Dcm_01048]		
Parent Container	DcmDspSecurityRow		
Description	Function name to request the value of an attempt counter. This parameter is related to the interface Xxx_GetSecurityAttemptCounter.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspSecurityGetSeedFnc [ECUC_Dcm_00968]		
Parent Container	DcmDspSecurityRow		
Description	Callout function name used to request a seed. This parameter is related to the interface Xxx_GetSeed.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			

Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspSecurityKeySize [ECUC_Dcm_00760]		
Parent Container	DcmDspSecurityRow		
Description	size of the security key (in Bytes).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 4294967295		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspSecurityLevel [ECUC_Dcm_00754]		
Parent Container	DcmDspSecurityRow		
Description	<p>Value of Security level. The locked state cannot be configured explicitly.</p> <p>1,2,3...63: configuration dependent - Conversion formula to calculate SecurityLevel out of tester requested</p> <p>SecurityAccessType parameter: $SecurityLevel = (SecurityAccessType(requestSeed) + 1) / 2$</p> <p>Type: Dcm_SecLevelType</p>		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	1 .. 63		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	

Scope / Dependency	scope: local
---------------------------	--------------

Name	DcmDspSecurityNumAttDelay [ECUC_Dcm_00762]		
Parent Container	DcmDspSecurityRow		
Description	Number of failed security accesses after which the delay time is activated		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspSecuritySeedSize [ECUC_Dcm_00755]		
Parent Container	DcmDspSecurityRow		
Description	size of the security seed (in Bytes).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 4294967295		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspSecuritySetAttemptCounterFnc [ECUC_Dcm_01049]		
Parent Container	DcmDspSecurityRow		
Description	Function name to set the value of an attempt counter. This parameter is related to the interface Xxx_SetSecurityAttemptCounter.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspSecurityUsePort [ECUC_Dcm_00967]		
Parent Container	DcmDspSecurityRow		
Description	Defines which kind of interface shall be used for security access.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	USE_ASYNC_CLIENT_SERVER	The DCM will access the data using an R-Port requiring a asynchronous ClientServerInterface SecurityAccess_{SecurityLevel}.	
	USE_ASYNC_FNC	The DCM will access the data using the functions that are defined in the parameters DcmDspSecurityGetSeedFnc and DcmDspSecurityCompareKeyFnc as well as the functions defined in DcmDspSecurityGetAttemptCounterFnc and DcmDspSecuritySetAttemptCounterFnc, if enabled by the parameter DcmDspSecurityAttemptCounterEnabled. DCM_E_PENDING return is allowed and OpStatus is existing as IN parameter.	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.5.19.3 DcmDspSession

SWS Item	[ECUC_Dcm_00769]
Container Name	DcmDspSession
Parent Container	DcmDsp
Description	Parent container holding single rows to configure particular sessions
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspSessionRow	0..31	This container holds all parameters needed to configure a single session

SWS Item	[ECUC_Dcm_00764]
Container Name	DcmDspSecurity
Parent Container	DcmDsp
Description	This container contains the configuration (DSP parameter) for security level configuration (per security level) Description This container contains Rows of DcmDspSecurityRow
Configuration Parameters	

Name	DcmDspSecurityMaxAttemptCounterReadoutTime [ECUC_Dcm_01101]		
Parent Container	DcmDspSecurity		
Description	Delay, in seconds, from startup (measured from the first call of the Dcm_MainFunction()), allowed for all AttemptCounter values to be obtained from the Application. Must be a multiple of the DcmTaskTime. min: A value equal to the DcmTaskTime		
Multiplicity	1		
Type	EcucFloatParamDef		
Range]0 .. 65535[
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspSecurityRow	0..31	Definition of a single Row of configuration for security level configuration (per security level) The name of this container is used to define the name of the R-Port through which the DCM accesses the interface SecurityAccess_{SecurityLevel}. The R-Port is named SecurityAccess_{SecurityLevel} where {SecurityLevel} is the name of the container DcmDspSecurityRow. If there is no reference, no check of security level shall be done.

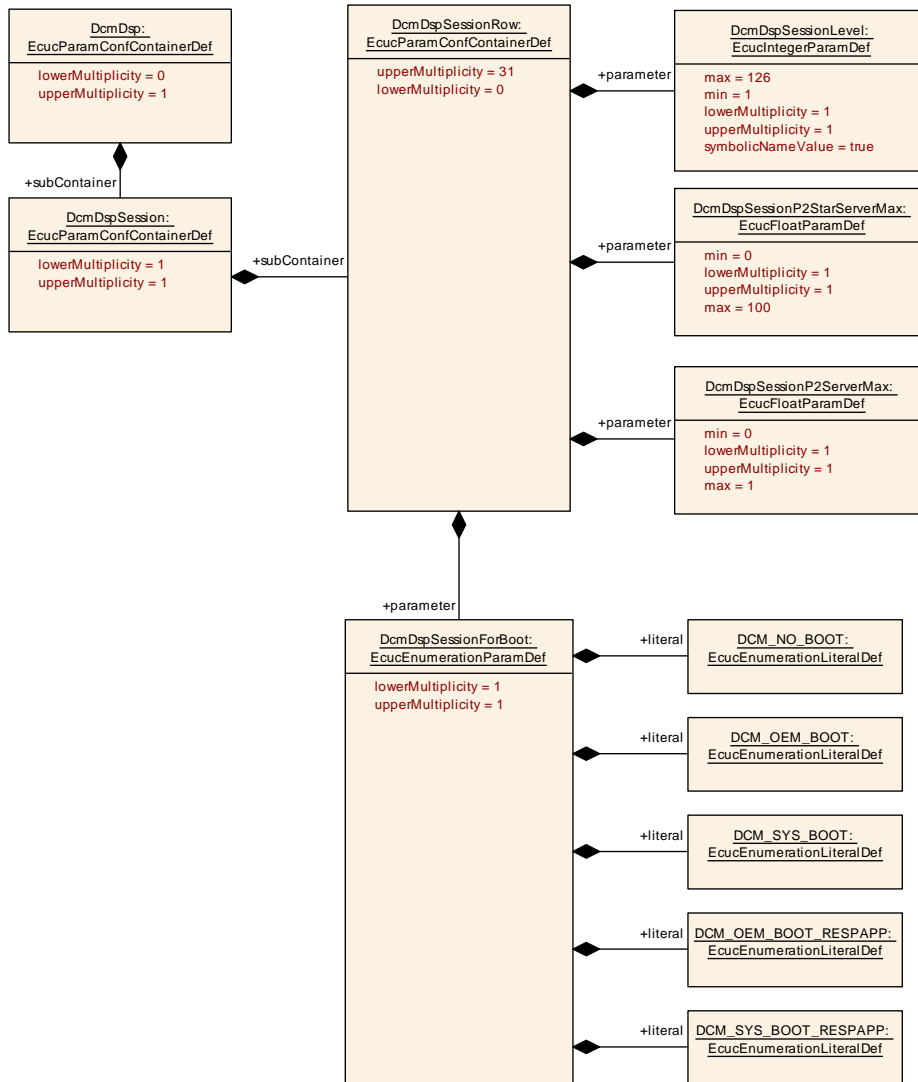


Figure 10.54: DcmDspSession configuration overview

10.2.5.19.4 DcmDspSessionRow

SWS Item	[ECUC_Dcm_00767]
Container Name	DcmDspSessionRow
Parent Container	DcmDspSession
Description	This container holds all parameters needed to configure a single session
Configuration Parameters	

Name	DcmDspSessionForBoot [ECUC_Dcm_00815]		
Parent Container	DcmDspSessionRow		
Description	This parameter defines whether this diagnostic session allows to jump to Bootloader (OEM Bootloader or System Supplier Bootloader) and determines, from which unit the final response will be sent. If this diagnostic session doesn't allow to jump to Bootloader the value DCM_NO_BOOT shall be chosen.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DCM_NO_BOOT	This diagnostic session doesn't allow to jump to Bootloader.	
	DCM_OEM_BOOT	This diagnostic session allows to jump to OEM Bootloader and bootloader sends final response.	
	DCM_OEM_BOOT_RESP APP	This diagnostic session allows to jump to OEM Bootloader and application sends final response.	
	DCM_SYS_BOOT	This diagnostic session allows to jump to System Supplier Bootloader and bootloader sends final response.	
	DCM_SYS_BOOT_RESP APP	This diagnostic session allows to jump to System Supplier Bootloader and application sends final response.	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspSessionLevel [ECUC_Dcm_00765]		
Parent Container	DcmDspSessionRow		
Description	subFunction value of the DiagnosticSession. 0, 127 and all values above 127 are reserved by ISO		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	1 .. 126		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspSessionP2ServerMax [ECUC_Dcm_00766]		
Parent Container	DcmDspSessionRow		
Description	This is the session value for P2ServerMax in seconds (per Session). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds. DCM configuration tools must convert this float value to the appropriate value format for the use in the software implementation of DCM. This value is reported to the tester within the response to the 'Session Control' service.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. 1]		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspSessionP2StarServerMax [ECUC_Dcm_00768]		
Parent Container	DcmDspSessionRow		
Description	This is the session value for P2*ServerMax in seconds (per Session). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds. DCM configuration tools must convert this float value to the appropriate value format for the use in the software implementation of DCM. This value is reported to the tester within the response to the 'Session Control' service.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. 100]		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.5.19.5 DcmModeCondition

SWS Item	[ECUC_Dcm_00928]
Container Name	DcmModeCondition
Parent Container	DcmProcessingConditions

Description	<p>This container contains the configuration of a mode condition or an environmental conditions which can be used as argument in DcmModeRules.</p> <p>One DcmModeCondition shall contain either one DcmSwcModeRef or one DcmBswModeRef or one DcmSwcSRDataElementRef.</p> <p>Please note that the Dcm acts as well as mode manager. Therefore the references DcmSwcModeRef or one DcmBswModeRef. might point to provided ModeDeclarationGroupPrototypes of the Dcm itself as well as to provided ModeDeclarationGroupPrototypes of other Bsw Modules or software components.</p> <p>In case of a configured DcmSwcModeRef or DcmBswModeRef only the DcmConditionType DCM_EQUALS or DCM_EQUALS_NOT are applicable.</p> <p>In case of DcmSwcSRDataElementRef all literals of DcmConditionType are possible.</p>
Configuration Parameters	

Name	DcmConditionType [ECUC_Dcm_00929]		
Parent Container	DcmModeCondition		
Description	This parameter specifies what kind of comparison that is made for the evaluation of the mode condition.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DCM_EQUALS		
	DCM_EQUALS_NOT		
	DCM_GREATER_OR_EQUAL		
	DCM_GREATER_THAN		
	DCM_LESS_OR_EQUAL		
	DCM_LESS_THAN		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmBswModeRef [ECUC_Dcm_00931]		
Parent Container	DcmModeCondition		
Description	<p>This parameter references a mode of a ModeDeclarationGroupPrototype provided by a Basic Software Module used for the condition.</p> <p>Please note that such ModeDeclarationGroupPrototype are owned by a Basic Software Module Description in the role providedModeGroup.</p>		
Multiplicity	0..1		
Type	Instance reference to MODE-DECLARATION context: MODE-DECLARATION-GROUP-PROTOTYPE		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmModeConditionCertificateCompareElementRef [ECUC_Dcm_01179]		
Parent Container	DcmModeCondition		
Description	Reference to a certificate data element that provides the compare value.		
Multiplicity	0..1		
Type	Symbolic name reference to KeyMCertificateElement		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmSwcModeRef [ECUC_Dcm_00930]		
Parent Container	DcmModeCondition		
Description	This parameter references a mode in a particular mode request port of a software component that is used for the condition.		
Multiplicity	0..1		
Type	Instance reference to MODE-DECLARATION context: ROOT-SW-COMPOSITION-PROTOTYPE SW-COMPONENT-PROTOTYPE P-PORT-PROTOTYPE MODE-DECLARATION-GROUP-PROTOTYPE		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmSwcSRDataElementRef [ECUC_Dcm_001037]		
Parent Container	DcmModeCondition		
Description	Reference to environmental conditions. It is possible to reference a S/R Receiver-Port to read physical values and compare (equal, greater, less,...) them with a configured value that is defined by DcmSwcDataElementValue.		
Multiplicity	0..1		
Type	Choice reference to [DcmDspExternalSRDataElement-Class,DcmDspPidService01ExternalSRDataElementClass]		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmSwcDataElement Value	1	This container contains the configuration of a compare value.

10.2.5.19.6 DcmSwcDataElementValue

SWS Item	[ECUC_Dcm_01123]
Container Name	DcmSwcDataElementValue
Parent Container	DcmModeCondition
Description	This container contains the configuration of a compare value.
Configuration Parameters	

Container Choices		
Container Name	Multiplicity	Scope / Dependency
DcmSwcDataElementArray	0..1	This container contains the configuration of a array compare value.
DcmSwcDataElementPrimitive	0..1	This container contains the configuration of a primitive compare value.

10.2.5.19.7 DcmSwcDataElementPrimitive

SWS Item	[ECUC_Dcm_01124]
Container Name	DcmSwcDataElementPrimitive
Parent Container	DcmSwcDataElementValue
Description	This container contains the configuration of a primitive compare value.
Configuration Parameters	

Name	DcmSwcDataElementPrimitiveValue [ECUC_Dcm_01126]		
Parent Container	DcmSwcDataElementPrimitive		
Description	Primitive compare value.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 ..		
	18446744073709551615		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

No Included Containers

10.2.5.19.8 DcmSwcDataElementArray

SWS Item	[ECUC_Dcm_01125]
Container Name	DcmSwcDataElementArray

Parent Container	DcmSwcDataElementValue
Description	This container contains the configuration of a array compare value.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmSwcDataElementArrayElement	0..*	This container contains the configuration of a array element compare value.

10.2.5.19.9 DcmSwcDataElementArrayElement

SWS Item	[ECUC_Dcm_01129]
Container Name	DcmSwcDataElementArrayElement
Parent Container	DcmSwcDataElementArray
Description	This container contains the configuration of a array element compare value.
Configuration Parameters	

Name	DcmSwcDataElementArrayElementIndex [ECUC_Dcm_01127]		
Parent Container	DcmSwcDataElementArrayElement		
Description	Index to an element of the compare value array.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

Name	DcmSwcDataElementArrayElementValue [ECUC_Dcm_01128]		
Parent Container	DcmSwcDataElementArrayElement		
Description	Value of an array element compare value.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default Value			
Post-Build Variant Value	false		

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

No Included Containers

10.2.5.19.10 DcmModeRule

SWS Item	[ECUC_Dcm_00925]
Container Name	DcmModeRule
Parent Container	DcmProcessingConditions
Description	<p>This container contains the configuration of a mode rule which represents a logical expression with DcmModeConditions or other DcmModeRules as arguments.</p> <p>All arguments are processed with the operator defined by DcmLogicalOperator, for instance: Argument_A AND Argument_B AND Argument_C</p>
Configuration Parameters	

Name	DcmLogicalOperator [ECUC_Dcm_00926]		
Parent Container	DcmModeRule		
Description	This parameter specifies the logical operator to be used in the logical expression. If the expression only consists of a single condition this parameter shall not be used.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	DCM_AND		
Post-Build Variant Multiplicity	DCM_OR		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmModeRuleNrcValue [ECUC_Dcm_00949]		
Parent Container	DcmModeRule		
Description	Optional parameter which defines the NRC to be sent in case the mode rule condition is not valid.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmArgumentRef [ECUC_Dcm_00927]		
Parent Container	DcmModeRule		
Description	This is a choice reference either to a mode condition or a an other mode rule serving as sub-expression. Attributes: requiresIndex=true		
Multiplicity	1..*		
Type	Choice reference to [DcmModeCondition,DcmModeRule]		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

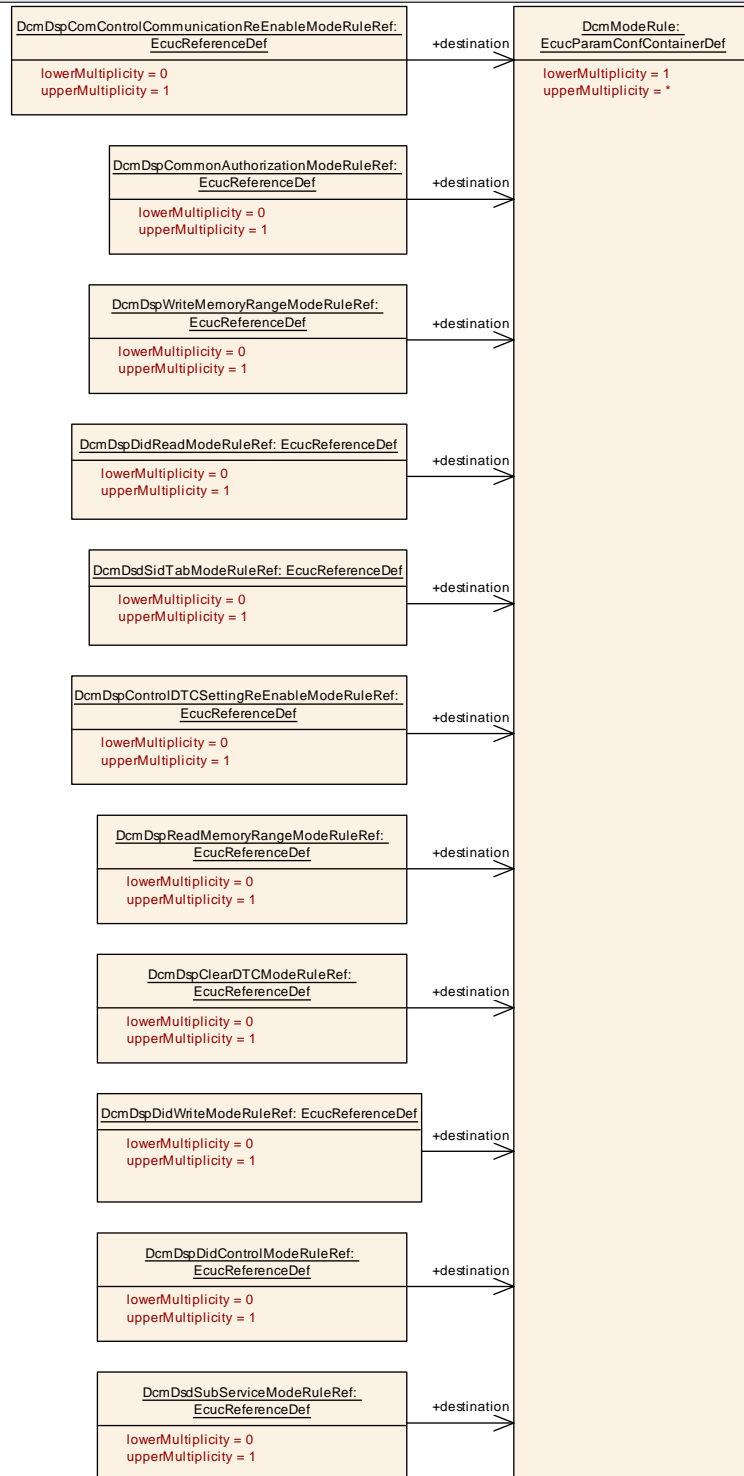


Figure 10.55: DcmModeRuleUsage configuration overview

10.2.5.20 DcmDspVehInfo

SWS Item	[ECUC_Dcm_00630]
Container Name	DcmDspVehInfo
Parent Container	DcmDsp
Description	This container contains the configuration (parameters) for one single VehicleInfoType of service \$09
Configuration Parameters	

Name	DcmDspVehInfoInfoType [ECUC_Dcm_00631]		
Parent Container	DcmDspVehInfo		
Description	value of InfoType. Within each DcmConfigSet all DcmDspVehInfoInfoType values shall be unique.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspVehInfoNODIProvResp [ECUC_Dcm_01051]		
Parent Container	DcmDspVehInfo		
Description	Indicate the Dcm, which side is responsible to fill the number of data items (NODI), Dcm or the provider of the InfoType data. In case the responsibility is on provider side, only one DcmDspVehInfoData container is allowed. <ul style="list-style-type: none"> • true: Provider is responsible for providing the number of data items parameter • false or not existing: Dcm is responsible for providing the number of data items parameter 		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DcmDspVehInfoData	1..*	Data Item of an InfoType; ShortName is post-fix of the port interface name.

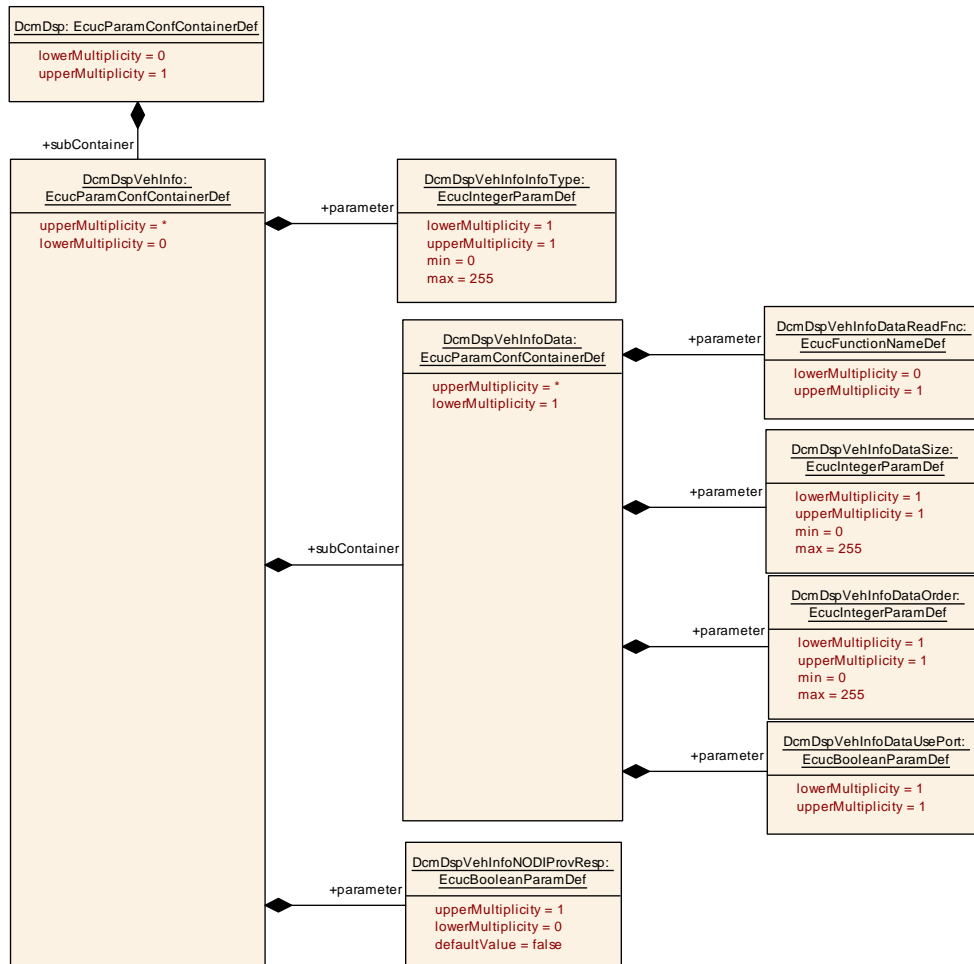


Figure 10.56: DcmDspVehInfo configuration overview

10.2.5.21 DcmDspVehInfoData

SWS Item	[ECUC_Dcm_00888]
Container Name	DcmDspVehInfoData
Parent Container	DcmDspVehInfo
Description	Data Item of an InfoType; ShortName is post-fix of the port interface name.
Configuration Parameters	

Name	DcmDspVehInfoDataOrder [ECUC_Dcm_00891]		
Parent Container	DcmDspVehInfoData		
Description	Defines the order of the data item in the InfoType; values: 0..255; first data item having the order number 0; the next 1 and so on. The configuration of order needs to be unique per InfoType.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspVehInfoDataReadFnc [ECUC_Dcm_00889]		
Parent Container	DcmDspVehInfoData		
Description	Callout function name for reading InfoType data item. Only required in case parameter 'DcmDspVehInfoDataUsePort' is set to 'false'		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspVehInfoDataSize [ECUC_Dcm_00890]		
Parent Container	DcmDspVehInfoData		
Description	Size in bytes of the InfoType data item.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Value	false		

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspVehInfoDataUsePort [ECUC_Dcm_00727]		
Parent Container	DcmDspVehInfoData		
Description	<p>When this parameter is set to true the DCM will access the Data using an R-Port requiring a PortInterface IInfotypeServices_{VehInfoData}. The R-Port is named InfotypeServices_{VehInfoData} where {VEHINFODATA} is the name of the container DcmDspVehInfoData. In that case, the DcmDspVehInfoDataReadFnc is ignored and the RTE APIs are used.</p> <p>When this parameter is set to false, the DCM calls the function defined in DcmDspVehInfoDataReadFnc.</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

10.2.5.22 DcmDspPeriodicTransmission

SWS Item	[ECUC_Dcm_00957]
Container Name	DcmDspPeriodicTransmission
Parent Container	DcmDsp
Description	This container contains the configuration (parameters) for Periodic Transmission Scheduler.
Configuration Parameters	

Name	DcmDspMaxPeriodicDidScheduler [ECUC_Dcm_00962]	
Parent Container	DcmDspPeriodicTransmission	
Description	Defines the maximum number of periodicDataIdentifiers that can be scheduled concurrently.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	1 .. 255	

Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspPeriodicTransmissionFastRate [ECUC_Dcm_00960]		
Parent Container	DcmDspPeriodicTransmission		
Description	<p>This parameter give the transmission rate of the requested periodicDataIdentifiers to be used if the parameter transmissionMode given in the ReadDataByPeriodicID request is equal to 0x03 ("sendAtFastRate"). This parameter value in seconds have to be configured as a multiple of DcmTaskTime.</p> <p>min: A negative value and zero is not allowed.</p>		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[1E-4 .. 1]		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspPeriodicTransmissionMaxPeriodicFastTransmissions [ECUC_Dcm_01182]		
Parent Container	DcmDspPeriodicTransmission		
Description	Defines the maximum number of periodic connections used for fast transmission.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspPeriodicTransmissionMaxPeriodicMediumTransmissions [ECUC_Dcm_01181]		
Parent Container	DcmDspPeriodicTransmission		
Description	Defines the maximum number of periodic connections used for medium transmission.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspPeriodicTransmissionMaxPeriodicSlowTransmissions [ECUC_Dcm_01183]		
Parent Container	DcmDspPeriodicTransmission		
Description	Defines the maximum number of periodic connections used for slow transmission.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspPeriodicTransmissionMediumRate [ECUC_Dcm_00959]		
Parent Container	DcmDspPeriodicTransmission		
Description	<p>This parameter give the transmission rate of the requested periodicDataIdentifiers to be used if the parameter transmissionMode given in the ReadDataByPeriodicID request is equal to 0x02 ("sendAtMediumRate"). This parameter value in seconds have to be configured as a multiple of DcmTaskTime.</p> <p>min: A negative value and zero is not allowed.</p>		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[1E-4 .. 1]		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspPeriodicTransmissionSchedulerType [ECUC_Dcm_01180]		
Parent Container	DcmDspPeriodicTransmission		
Description	Defines the used scheduler type according to ISO 14229-1:2018 for read data by periodic identifier.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	SCHEDULER_TYPE1		Selects the scheduler type#1 of ISO 14229-1:2018
	SCHEDULER_TYPE2		Selects the scheduler type#2 of ISO 14229-1:2018
Default Value	SCHEDULER_TYPE1		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDspPeriodicTransmissionSlowRate [ECUC_Dcm_00958]		
Parent Container	DcmDspPeriodicTransmission		
Description	<p>This parameter give the transmission rate of the requested periodicDataIdentifiers to be used if the parameter transmissionMode given in the ReadDataByPeriodicID request is equal to 0x01 ("sendAtSlowRate"). This parameter value in seconds have to be configured as a multiple of DcmTaskTime.</p> <p>min: A negative value and zero is not allowed.</p>		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[1E-4 .. 1]		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

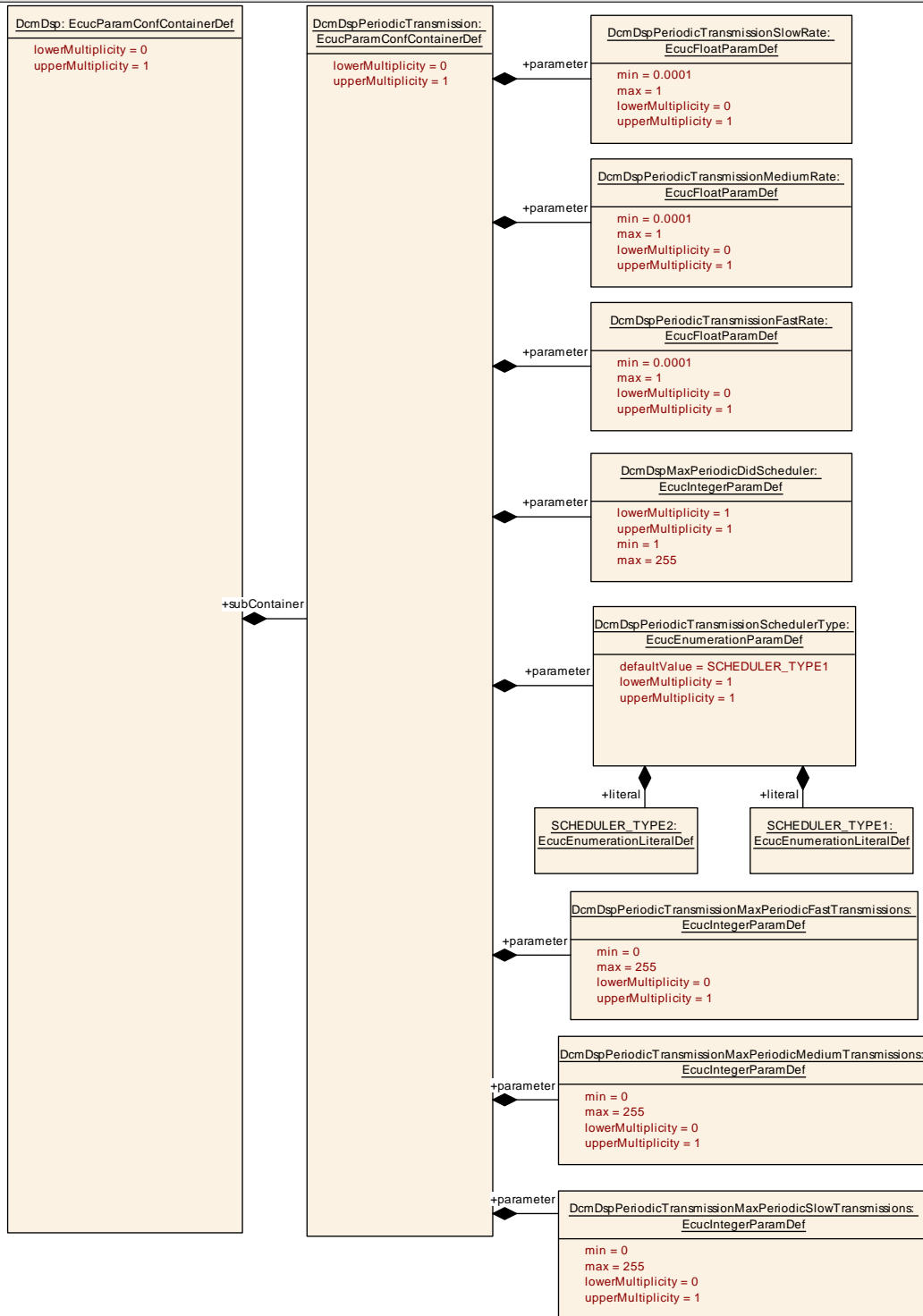


Figure 10.57: DcmDspPeriodicTransmission configuration overview

10.2.5.23 DcmDspClearDTC

SWS Item	[ECUC_Dcm_01064]
-----------------	------------------

Container Name	DcmDspClearDTC
Parent Container	DcmDsp
Description	This container contains the configuration for the Clear DTC service.
Configuration Parameters	

Name	DcmDspClearDTCCheckFnc [ECUC_Dcm_01066]		
Parent Container	DcmDspClearDTC		
Description	<p>Callback function for condition check, manufacturer / supplier specific checks on the groupOfDTC, which is requested to clear.</p> <p>This parameter is related to the interface : Xxx_ClearDTCCheckFnc.</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmDspClearDTCModeRuleRef [ECUC_Dcm_01065]		
Parent Container	DcmDspClearDTC		
Description	<p>Reference to DcmModeRule</p> <p>Mode rule which controls to clear the DTCs. If there is no reference, no check of the mode rule shall be done.</p>		
Multiplicity	0..1		
Type	Reference to DcmModeRule		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	

Scope / Dependency	scope: ECU
---------------------------	------------

No Included Containers

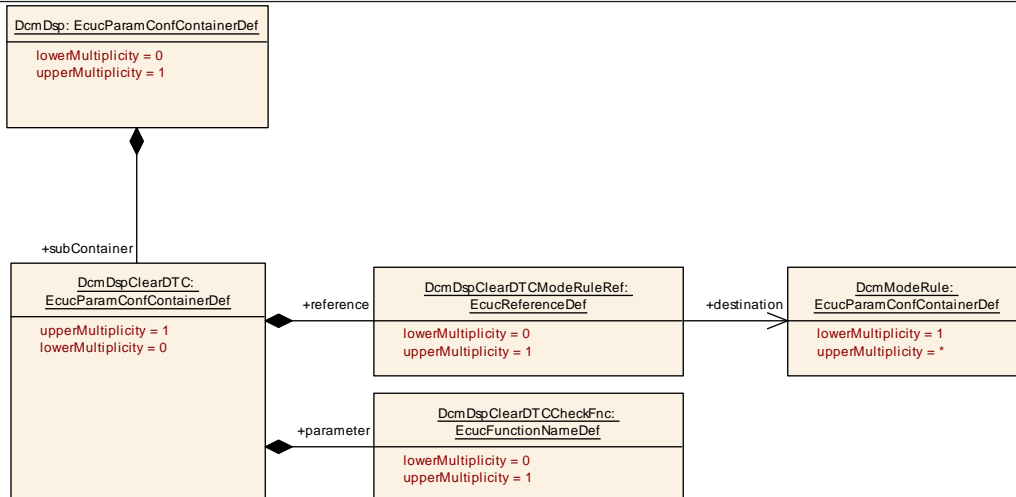


Figure 10.58: DcmDspClearDTC configuration overview

10.2.6 DcmGeneral

SWS Item	[ECUC_Dcm_00822]
Container Name	DcmGeneral
Parent Container	Dcm
Description	Contains general configuration parameters valid for the entire Dcm module.
Configuration Parameters	

Name	DcmDDDIDStorage [ECUC_Dcm_00971]		
Parent Container	DcmGeneral		
Description	This configuration switch defines, whether DDDID definition is stored non-volatile or not. true: DDDID are stored non-volatile false: DDDID are only maintained volatile		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	

Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmDevErrorDetect [ECUC_Dcm_00823]		
Parent Container	DcmGeneral		
Description	Switches the development error detection and notification on or off. <ul style="list-style-type: none"> • true: detection and notification is enabled. • false: detection and notification is disabled. 		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmHeaderFileInclusion [ECUC_Dcm_01019]		
Parent Container	DcmGeneral		
Description	Name of the header file(s) to be included by the Dcm module containing the used C-callback declarations.		
Multiplicity	0..*		
Type	EcucStringParamDef		
Default Value			
Regular Expression	[a-zA-Z0-9_]([a-zA-Z0-9\._])*		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmRespondAllRequest [ECUC_Dcm_00600]		
Parent Container	DcmGeneral		
Description	If set to FALSE the Dcm will not respond to diagnostic request that contains a service ID which is in the range from 0x40 to 0x7F or in the range from 0xC0 to 0xFF (Response IDs).		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	DcmTaskTime [ECUC_Dcm_00820]		
Parent Container	DcmGeneral		
Description	<p>Allow to configure the time for the periodic cyclic task. Please note: This configuration value shall be equal to the value in the RTE module.</p> <p>The AUTOSAR configuration standard is to use SI units, so this parameter is defined as float value in seconds. Dcm configuration tools must convert this float value to the appropriate value format for the use in the software implementation of Dcm.</p> <p>min: A negative value and zero is not allowed.</p> <p>upperMultiplicity: Exactly one TaskTime must be specified per configuration.</p> <p>lowerMultiplicity: Exactly one TaskTime must be specified per configuration.</p>		
Multiplicity	1		
Type	EcucFloatParamDef		
Range]0 .. INF[
Default Value			
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD
	Link time	X	VARIANT-LINK-TIME
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmVersionInfoApi [ECUC_Dcm_00821]		
Parent Container	DcmGeneral		
Description	Preprocessor switch to enable or disable the output Version info of the functionality.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	DcmVinRef [ECUC_Dcm_00984]		
Parent Container	DcmGeneral		
Description	Reference to the Did containing the VIN Information. This parameter is needed for function Dcm_GetVin		
Multiplicity	0..1		
Type	Reference to DcmDspDid		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

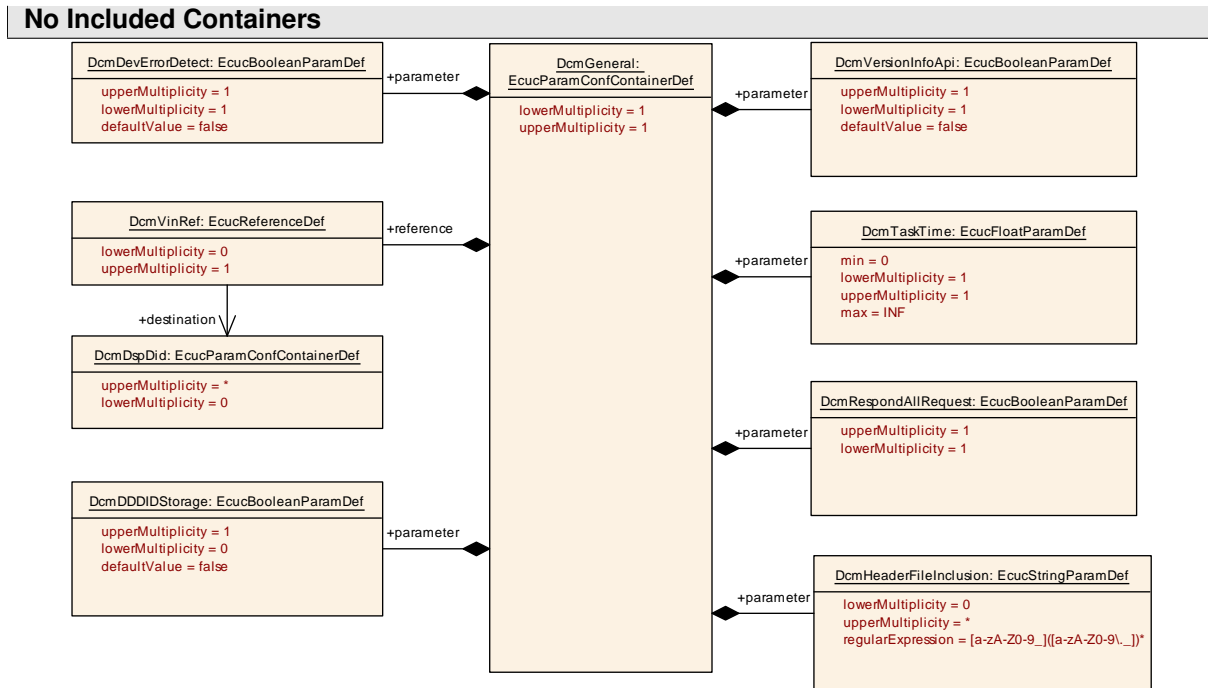


Figure 10.59: DcmGeneral configuration overview

10.3 Protocol Configuration Example

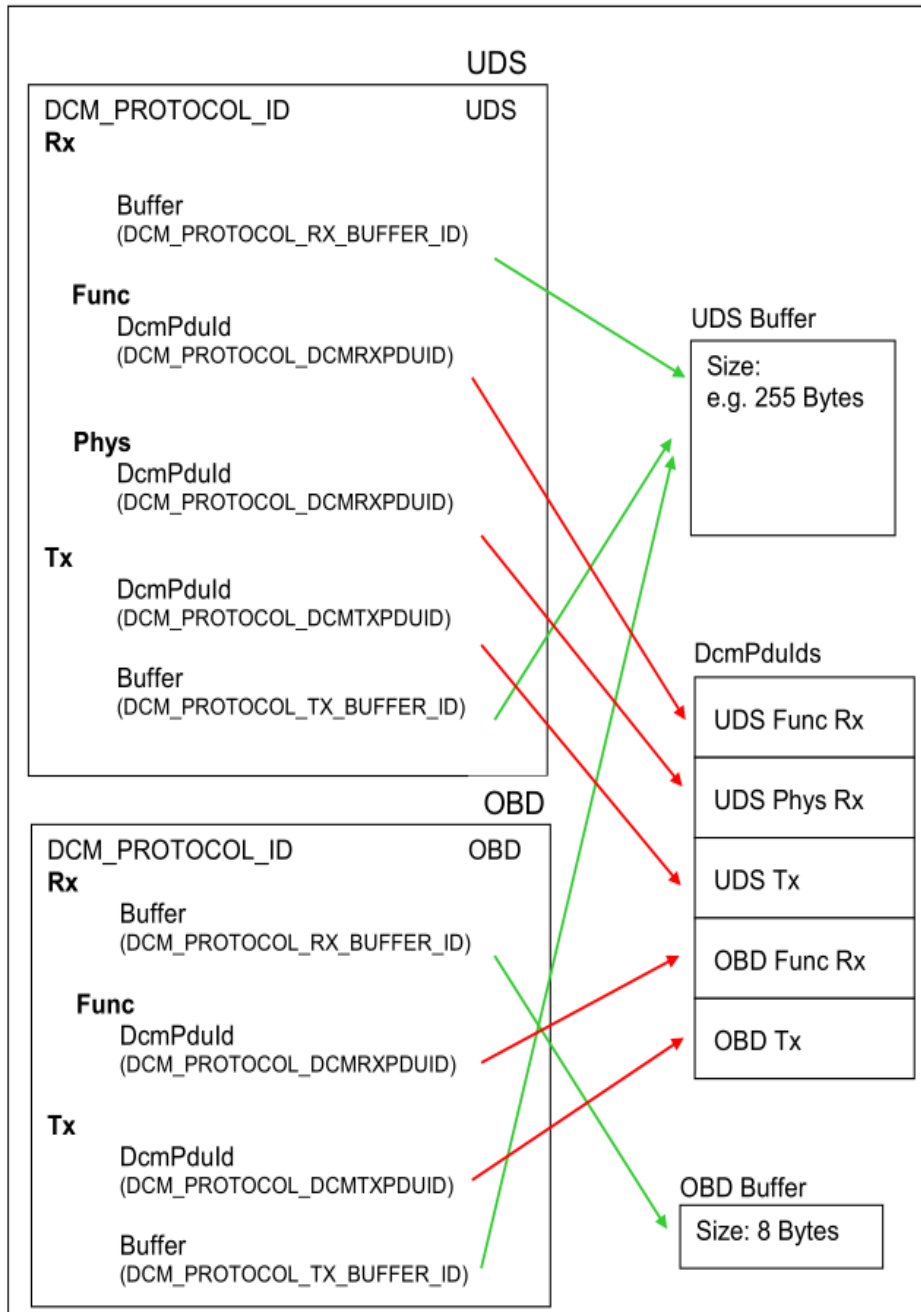


Figure 10.60: Examples of protocol configuration with focus on buffer / DcmPduld settings

Above example shows protocol configuration at the use cases examples [OBD](#) and [UDS](#) (used for customer enhanced diagnosis). It is assumed that for [UDS](#) communication, there are functional and physical requests. There will be separate `DcmPduRxDs` for functional and physical reception.

Concerning buffer configuration it is proposed to use a separate buffer for the functional requests. This in correspondence to support the keep alive logic with functional addressed TesterPresent commands.

It is also proposed to use a separate receive buffer for the OBD commands. This in reference to support the protocol switch functionality.

It is allowed to share for both protocols the transmit buffer. Please note: The `DcmDslProtocolRx` has two possible configurations:

- functional
- physical

The physical shall have a 1:1 (or 1:0) dependency to the `DcmDslMainConnection`. (which means: `DcmDslProtocolRxPduRef` in combination `DCM_PROTOCOL_RX_ADDR_TYP = physical` can exist only once per "Module") The functional shall have a 1:n dependency to the `DcmDslMainConnection`. (which means: `DcmDslProtocolRxPduRef` in combination `DCM_PROTOCOL_RX_ADDR_TYP = functional` can exist several times per "Module") The `DcmDslProtocolTx` shall exist only once per "Module"

10.4 Published Information

For details refer to the chapter 10.3 "Published Information" in SWS_BSWGeneral [7].

A Not applicable requirements

[SWS_Dcm_NA_00999] [These requirements are not applicable to this specification.] ([SRS_BSW_00005](#), [SRS_BSW_00006](#), [SRS_BSW_00007](#), [SRS_BSW_00009](#),

*SRS_BSW_00010, SRS_BSW_00158, SRS_BSW_00159, SRS_BSW_00160, SRS_ -
BSW_00161, SRS_BSW_00162, SRS_BSW_00164, SRS_BSW_00167, SRS_ -
BSW_00168, SRS_BSW_00170, SRS_BSW_00171, SRS_BSW_00172, SRS_ -
BSW_00300, SRS_BSW_00301, SRS_BSW_00304, SRS_BSW_00305, SRS_ -
BSW_00306, SRS_BSW_00307, SRS_BSW_00308, SRS_BSW_00309, SRS_ -
BSW_00310, SRS_BSW_00312, SRS_BSW_00314, SRS_BSW_00318, SRS_ -
BSW_00321, SRS_BSW_00323, SRS_BSW_00325, SRS_BSW_00327, SRS_ -
BSW_00328, SRS_BSW_00330, SRS_BSW_00331, SRS_BSW_00333, SRS_ -
BSW_00334, SRS_BSW_00335, SRS_BSW_00336, SRS_BSW_00339, SRS_ -
BSW_00341, SRS_BSW_00342, SRS_BSW_00343, SRS_BSW_00344, SRS_ -
BSW_00345, SRS_BSW_00346, SRS_BSW_00347, SRS_BSW_00350, SRS_ -
BSW_00351, SRS_BSW_00353, SRS_BSW_00357, SRS_BSW_00358, SRS_ -
BSW_00359, SRS_BSW_00360, SRS_BSW_00361, SRS_BSW_00371, SRS_ -
BSW_00374, SRS_BSW_00375, SRS_BSW_00377, SRS_BSW_00378, SRS_ -
BSW_00379, SRS_BSW_00380, SRS_BSW_00383, SRS_BSW_00384, SRS_ -
BSW_00385, SRS_BSW_00386, SRS_BSW_00388, SRS_BSW_00389, SRS_ -
BSW_00390, SRS_BSW_00392, SRS_BSW_00393, SRS_BSW_00394, SRS_ -
BSW_00395, SRS_BSW_00396, SRS_BSW_00397, SRS_BSW_00398, SRS_ -
BSW_00399, SRS_BSW_00400, SRS_BSW_00401, SRS_BSW_00402, SRS_ -
BSW_00403, SRS_BSW_00404, SRS_BSW_00405, SRS_BSW_00406, SRS_ -
BSW_00408, SRS_BSW_00409, SRS_BSW_00410, SRS_BSW_00411, SRS_ -
BSW_00413, SRS_BSW_00414, SRS_BSW_00415, SRS_BSW_00416, SRS_ -
BSW_00417, SRS_BSW_00419, SRS_BSW_00422, SRS_BSW_00423, SRS_ -
BSW_00425, SRS_BSW_00426, SRS_BSW_00427, SRS_BSW_00428, SRS_ -
BSW_00429, SRS_BSW_00432, SRS_BSW_00433, SRS_BSW_00437, SRS_ -
BSW_00439, SRS_BSW_00440, SRS_BSW_00441, SRS_BSW_00447, SRS_ -
BSW_00448, SRS_BSW_00449, SRS_BSW_00450, SRS_BSW_00451, SRS_ -
BSW_00452, SRS_BSW_00453, SRS_BSW_00454, SRS_BSW_00456, SRS_ -
BSW_00457, SRS_BSW_00458, SRS_BSW_00459, SRS_BSW_00460, SRS_ -
BSW_00461, SRS_BSW_00462, SRS_BSW_00463, SRS_BSW_00464, SRS_ -
BSW_00465, SRS_BSW_00466, SRS_BSW_00467, SRS_BSW_00469, SRS_ -
BSW_00470, SRS_BSW_00471, SRS_BSW_00472, SRS_BSW_00473, SRS_ -
BSW_00477, SRS_BSW_00478, SRS_BSW_00479, SRS_BSW_00480, SRS_ -
BSW_00481, RS_Diag_04002, RS_Diag_04007, RS_Diag_04019, RS_Diag_04024,
RS_Diag_04031, RS_Diag_04032, RS_Diag_04057, RS_Diag_04059, RS_Diag_ -
04063, RS_Diag_04064, RS_Diag_04068, RS_Diag_04071, RS_Diag_04077, RS_ -
Diag_04085, RS_Diag_04086, RS_Diag_04087, RS_Diag_04089, RS_Diag_04090,
RS_Diag_04091, RS_Diag_04093, RS_Diag_04097, RS_Diag_04100, RS_Diag_ -
04101, RS_Diag_04105, RS_Diag_04107, RS_Diag_04109, RS_Diag_04110, RS_ -
Diag_04111, RS_Diag_04112, RS_Diag_04113, RS_Diag_04117, RS_Diag_04118,
RS_Diag_04119, RS_Diag_04120, RS_Diag_04121, RS_Diag_04123, RS_Diag_ -
04124, RS_Diag_04125, RS_Diag_04126, RS_Diag_04127, RS_Diag_04129, RS_ -
Diag_04131, RS_Diag_04133, RS_Diag_04135, RS_Diag_04136, RS_Diag_04137,
RS_Diag_04139, RS_Diag_04140, RS_Diag_04143, RS_Diag_04144, RS_Diag_ -
04145, RS_Diag_04146, RS_Diag_04148, RS_Diag_04150, RS_Diag_04151)*