AUTOSAR Dcm User Manual

1.1 Document Change History

Date (YYYY-MM-DD)	Ver.	Editor	Chap	Description (before change -> after revision)
2016-04-18	1.0.0	SG. Baek	All	• Initial Creation
2016-05-25	1.0.1	J. Jung	All	• Renewal
2016-05-30	1.0.2	J. Jung	6.1.1	• Dcm 1.1.0 Update
2016-07-15	1.0.3	J. Jung	6.1.4.4 10.2.1	• Dcm 1.1.1 Update
2016-10-16	1.0.4	J. Jung	5.4 7.3.1 10.2 and 10.3	• Dcm 1.1.2 Update
2016-10-28	1.0.5	J. Jung	5.4.2 6.1.1	• Dcm 1.2.0 Update
2016-11-16	1.0.6	J. Jung	6.1.4	• Dcm 1.2.1 Update
2016-12-07	1.0.7	J. Jung	5.3	• Dcm 1.2.2 Update
2017-01-10	1.0.8	J. Jung	5.3 6.1.2	• Dcm 1.3.0 Update

			8.2.1	
2017-02-17	1.0.9	J. Jung	5.3 5.4 7.3 10.4	• Dcm 1.3.1 Update
2017-03-28	1.0.10	J. Jung	6.1.3 10.5 10.6	• Dcm 1.4.0 Update
2017-04-14	1.0.11	J. Jung	5.3	• Dcm 1.4.1 Update
2017-04-14	1.0.12	J. Jung	5.3 3 10.1 10.2	• Dcm 1.5.0 Update (see 5.3 Change Log) - Added explanation of Seed and RandomSeed - Added Dcm_GetRandomSeed() and Dcm_GetPublicKey callout function and guides • Modified Seed-Key (L1) and Advanced Seed-Key (L9) Sample Code, and consolidated into Appendix 10.2 SecurityAccess Sample Code
2017-05-30	1.0.13	J. Jung	5.3	• Dcm 1.5.1 Update (see Change Log)
2017-06-08	1.0.14	J. Jung	5.3 5.4.2	• Dcm 1.5.2 Update (see Change Log) - Partially applied to AUTOSAR_SWS_DiagnosticCommunicationManager 4.2.2 DcmDslDiagRespMaxNumRespPend
2017-06-28	1.0.15	J. Jung	5.3	• Dcm 1.6.0 Update (see Change Log)
2017-10-31	1.0.16	YJ. Yun	5.3 10.2	• Dcm 1.7.0 Update (see Change Log)
2017-11-10	1.0.17	YJ. Yun	5.3	• Dcm 1.7.1 Update (see Change Log)

2017-11-23	1.0.18	YJ. Yun	11.1.3.1 11.2	• Added cautions to update Seed through Dcm_GetRandomSeed() when C-SAC is going to be applied with pseudo random • Modified the Security Access example
2018-03-28	1.0.19	YS. Jeon	1 5.3 6.1.1	• Dcm 1.7.3 Update (see Change Log)
2018-06-19	1.0.20	YS. Jeon	5.3 5.4.2 6.1.1 6.1.4.5	• Dcm 1.8.0 Update (see Change Log) - Support of ES95486-50 specifications - Indication Callback ASR 4.3.0 was applied - Change to DcmTimStrP2(Star)ServerAdjust Max value - Change to DcmDspSessionP2(Star)ServerMax Max value
2018-09-12	1.0.21	YS. Jeon	5.3 6.1.1	• Dcm 1.9.0 Update (see Change Log) - Support of **QZN04** specifications - Change to the priority of SID31 Subfunction NRC
2018-09-18	1.0.22	YS. Jeon	5.3	• Dcm 1.9.1 Update (see Change Log) - Created Dcm.exe file - Modified RTRT dynamic verification Dcm
2018-11-08	1.0.23	YS. Jeon	5.3 10.1.2.1.1 10.2.2.1.2.4	• Dcm 1.9.2 Update (see Change Log) - Modified RoutineControl stop DataIn - Added RemainUnlockCondition - Added explanation of Dcm_GetCertificationInfo - Applied HSM to F1KM
2018-11-19	1.0.24	YS. Jeon	5.3	• Dcm 1.9.3 Update (see Change Log) - Applied AutoEver library from Security Level 21 ETAS library
2019-02-19	1.0.25	YS. Jeon	5.3	• Dcm 1.9.4 Update (see Change Log)
2019-06-21	1.0.26	YS. Jeon	5.3 5.4.1 6.1.1	• Dcm 1.9.5 Update (see Change Log) • Added DcmRemainUnlockCondition to DcmGeneral • Specified the sub-functions which are not supported by READDTCINFORMATION SERVICE in Limitations

2019-07-29	1.0.27	YS. Jeon	5.3 5.4.1 6.1.1	• Dcm 2.0.0 Update (see Change Log) • Added DcmForcedEcuReset to DcmGeneral • Deleted RequestDownload, TransferData, RequestTransferExit from Limitations
2019-09-11	1.0.28	EK. Kim	5.3 6.1.5	• Dcm 2.1.0 Update (see Change Log) • Added DcmDspDataBlockIdRef support
2019-10-10	1.1.0	EK. Kim	5.3 6	• Dcm 2.1.0.0 Update (see Change Log) • Configuration item properties were changed
2019-12-11	1.1.1	YJ. Yun	5.3 6	• Dcm 2.1.1.0 Update (see Change Log) • Configuration item properties were changed
2019-12-16	1.1.2	EK. Kim	5.3 7.1.5 10.1.4	• Dcm 2.2.0.0 Update (see Change Log) • Added Dcm_NegativeResponseCodeType • Added Security Access 2.0 Guide
2020-04-06	1.1.3	EK. Kim	5.3 6.1.1, 6.1.5 7.3 10.1.4	• Dcm 2.3.0.0 Update (see Change Log) • Configuration item properties were added and changed • Interface was added • Updated Security Access 2.0 Guide
2020-04-13	2.3.1	EK. Kim	10.2.2.1	• Improved description of Advanced Seedkey Reference code (update of reference code)
2020-10-15	2.3.1.0	YJ. Yun	5.3	• Dcm 2.3.1.0 Update (see Change Log)
2020-11-04	2.3.2.0	EK. Kim	5.3	• Dcm 2.3.2.0 Update (see Change Log)
2021-01-13	2.3.3.0	EK. Kim	5.3	• Dcm 2.3.3.0 Update (see Change Log)
2021-01-28	2.3.4.0	EK. Kim	5.3	• Dcm 2.3.4.0 Update (see Change Log)
2021-03-17	2.3.5.0	EK. Kim	5.3	• Dcm 2.3.5.0 Update (see Change Log)
2021-03-31	2.3.6.0	EK. Kim	5.3	• Dcm 2.3.6.0 Update (see Change Log)
2021-04-19	2.3.7.0	EK. Kim	5.3	• Dcm 2.3.7.0 Update (see Change Log)
2021-04-30	2.3.8.0	SK. Park	5.3 6.1.5.9.2	• Dcm 2.3.8.0 Update (see Change Log) • Note was added
2021-06-02	2.3.8.1	SK. Park	7.3.1.2 10.2.2.1.2	• Modified GetSesCtrlType breakdown (see Change Log) • Added explanation and modified the name of the RandomGenerate function in AppDcm_GetSeed_L9
2021-06-11	2.3.9.0	SK. Park	5.3	• Dcm 2.3.9.0 Update (see Change Log)

			7.3.12	• Included DCM_E_PENDING in the return value description of Xxx_Start(), Xxx_Stop(), and Xxx_RequestResults() function
2021-06-17	2.3.2.1	JH Lim	5.3	• Dcm 2.3.2.1 Update (see Change Log)
2021-06-17	2.3.10.0	SK. Park	5.3	• Dcm 2.3.10.0 Update (see Change Log)
2021-07-05	2.3.11.0	YJ.Yun	5.3	• Dcm 2.3.11.0 Update (see Change Log)
2021-09-08	2.3.11.1	DK.NAM	10.1.5	• Guided constraints in diagnostic services depending on engine condition
2021-09-21	2.3.12.0	YJ.Yun	5.3	• Dcm 2.3.12.0 Update (see Change Log)
2021-10-01	2.3.13.0	YJ.Yun	5.3	• Dcm 2.3.13.0 Update (see Change Log)
2021-11-12	2.3.14.0	KH.Kim	5.3	• Dcm 2.3.14.0 Update (see Change Log)
2021-12-13	2.3.15.0	KH.Kim	11.2.2.1.2	• Added warning when using True Random Generator with Autoever HSM 2.0 • Dcm 2.3.15.0 Update (see Change Log) • Added explanation of 8.3.13.3 and 8.3.13.4 Input parameter OpStatus • Added explanation of AppDcm_GetRandomSeed function to the note
2021-12-31	2.3.16.0	LanhLT	6.3	• Dcm 2.3.16.0 Update (see Change Log)
2022-01-12	2.3.17.0	KH.Kim	6.3	• Dcm 2.3.17.0 Update(see Change Log)
2022-01-25	2.3.18.0	KH.Kim	6.3 11.1.2.1.1	• Dcm 2.3.18.0 Update(see Change Log) • Added explanation of Dcm_GetCerHolderReference API
2022-02-25	2.4.0.0	DK.Nam	6.3	• Dcm 2.4.0.0 Update(see Change Log)
2022-02-25	2.4.0.0	KH.Kim	6.3 11.1.2.1.1 9.2.1	• Dcm 2.4.0.0 Update(see Change Log) • Deleted Dcm_GetCerHolderReference API and added explanation of Dcm_GetCertificationInfo API • Refined Not Supported, Fixed and Changeable Items • Added error message for failure in meeting ES specifications
2022-05-27	2.5.0.0	LanhLT	6.3	• Dcm 2.5.0.0 Update(see Change Log)
2022-08-12	2.5.1.0	LanhLT	6.3	• Dcm 2.5.1.0 Update(see Change Log)

2022-09-22	2.5.3.0	LanhLT	6.3, 9.2.1	• Dcm 2.5.2.0 Update (see Change Log) • Add new error message
2022-09-30	2.6.0.0	LanhLT	6.3, 6.4.1, 7.1, 8.1, 8.3.1, 9.2.1, 10.1.1	• Add Authentication Service
2022-11-29	2.6.0.1	KH Kim	7.1.4.1, 6.3	• Dcm 2.6.0.1 Update (see Change Log) • Add the information for setting buffer size when using Rxswin.
2023-01-18	2.6.1.0	DanhTQ1	6.3 9.2.1 7.1.4.4	• Dcm 2.6.1.0 Update (see Change Log) • Add new error message • Add DcmDslProtocolRxConnectionId
2023-04-19	2.6.3.0	LanhLT	6.2, 6.3, 6.4.2, 8.3.14	• Dcm 2.6.3.0 Update (see Change Log) • Add user defined function for Authentication service (Dcm_Authentication_User_CallOut)
2023-04-24	2.6.4.0	KT Kim	6.3	• Dcm 2.6.4.0 update(see Change Log)
2023-05-26	2.6.5.0	KT Kim	6.3 7.1.4.3	• Dcm 2.6.5.0 update(see Change Log) • Change DcmDslDiagRespOnSecondDeclinedRequest configuration to unsupported
2023-06-29	2.6.5.0 _hotfix	GS Ryu	6.3 7.1.4.4.1 9.2.1	• Dcm 2.6.5.0_hotfix Update (see Change Log) • Delete DcmDslProtocolRxConnectionId parameter from DcmDslConnection • ConnecionID related message ERR053221, ERR053222 strikethrough processing
2023-07-31	**2.7.0.0**	**EK Kim**	**6.2 6.3 7.1.1 7.1.5.10 7.1.5.11 7.1.5.17 7.1.5.18 7.1.5.21 9.2.1**	• Dcm 2.7.0.0 Update(see Change Log) • Add DcmObdProtocolId Configuration • Change DcmDspPid 설정 변경 (N -> C) • Change DcmDspRequestControl 설정 변경 (N -> C) • Change DcmDspTestResultByObdmid 설정 변경 (N -> C) • Change DcmDspVehInfo 설정 변경 (N ->C) • Add DcmDspReadDTCInformationSupportedObdUds DtcSeparation
2023-08-28	**2.7.1.0**	**SY Kim**	**6.3**	• Dcm 2.7.1.0 Update(Refer Change Log)
2023-11-16	**2.8.0.0**	**SY Kim**	**6.3 7.1.5.20.2**	• Dcm 2.8.0.0 Update(Refer Change Log) • Update DcmDspAuthenticationConnectionES

2023-11-27	**2.9.0.0**	**GS Ryu**	**6.3** **7.1.1**	• Dcm 2.9.0.0 Update (Refer Change Log) • Change DcmFblUsedType related to Fbl parameter
2023-12-29	2.10.0.0	SY Kim	6.3 7.1.5.22 8.3.16 9.2.1	• Dcm 2.10.0.0 Update(Refer Change Log) • Add RequestFileTransfer Service
2024-01-04	2.10.1.0	GS Ryu	6.3 7.1.5.9 8.3.4	• Dcm 2.10.1.0 Update (Refer Change Log) • Insert phrases related to RequestDownload and RequestUpload (requires implementation in Callout when Range Check is required)
2024-01-31	2.11.0.0	DH Kwak	6.2, 6.3	• Dcm 2.11.0.0 Update (Refer Change Log)
2024-02-23	2.11.0.0_HF1	GS Ryu	6.3 7.1.5.14	• Dcm 2.11.0.0_HF1 Update (Refer Change Log) • DcmDspDidDataPos must additionally be entered with a battery of 8.
2024-04-23	2.11.1.0	JH Hong	6.3	• Dcm 2.11.1.0 Update (Refer Change Log)
2024-04-26	2.12.0.0	SY Kim	6.2, 6.3 7.1.1 7.1.5.14	• Dcm 2.12.0.0 Update (Refer Change Log) • Delete DcmSecureFlashSupport • Add Configuration for DcmRoutineInfo Role
2024-04-30	2.9.0.0_HF1	JH Lee	6.3	• Dcm 2.9.0.0_HF1 Update (Refer Change Log)
2024-06-28	2.12.0.0_HF1	JH Hong	6.3	• Dcm 2.12.0.0_HF1 Update (Refer Change Log)
2024-07-11	2.13.0.0	JH Lee	6.3	• Dcm 2.13.0.0 Update (Refer Change Log) • 8.3.4 Remove note • 10.1.1 Add Service ID (0x41)
2024-07-26	2.9.0.0_HF2	JH Lee	6.3	• Dcm 2.9.0.0_HF2 Update (Refer Change Log)
2024-08-30	2.13.1.0	JH Lee	6.3 7.1	• Dcm 2.13.1.0 Update (Refer Change Log) • Add item in DcmGeneral of 7.1 Configuration
2024-09-10	2.13.0.0_HF1	JH Hong	6.3	• Dcm 2.13.0.0_HF1 Update (Refer Change Log)
2024-10-11	2.14.0.0	HW Seo	6.3 6.4.2 7.1.5.15 7.1.5.15.1 8.3.7.1.4 8.3.7.1.5	• Dcm 2.14.0.0 Update (Refet Change Log) • Change DcmDslDiagRespMaxNumRespPend Type • Add configuration in DcmDspSecurity • Add configurations in DcmDspSecurityRow • Add Xxx_{Get/Set}SecurityAttemptCounter asynchronous operations in SecurityAccess_{SecurityLevel} • Add Generator error message

			9.2.1 11.2.3 11.2.4	• Add sample code for Xxx_{Get/Set}SecurityAttemptCounter

Table of Contents

2 Overview

This document provides considerations in configuration of diagnostic stack parameters or system design.

See reference documents for more details.

Note : This document is based on the AUTOSAR diagnostic stack, and HMC ES95486-00 and ES95486-02.

The following terms on configuration category mean:

- Changeable (C): Items that can be configured by user
- Fixed (F): Items that cannot be changed by user
- NotSupported (N): Items that are not used

3 Reference

Sl. No.	Title	Version
1.	AUTOSAR BSW Service API Guide.doc	1.0.0 or later
2.	AUTOSAR_SWS_DiagnosticCommunicationManager.pdf	4.2.0
3.	ES95486-00.pdf	1.9.0 or later
4.	ES95486-02.pdf	1.1.1 or later

4 Acronyms and abbreviations

Acronym:	*Description:*
N_OK	Not OK
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	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: • development, • manufacturing, and • service (in a garage). Example External Diagnostic Tools are: • a diagnostic tester, • an OBD scan tool. The External Diagnostic Tool is to be connected by a mechanic to gather information from “inside” the car.
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	A device/ECU which is connected to the vehicle communication network. The Internal Diagnostic Tool can be used for: advanced event tracking, advanced analysis, for service. 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.
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’.

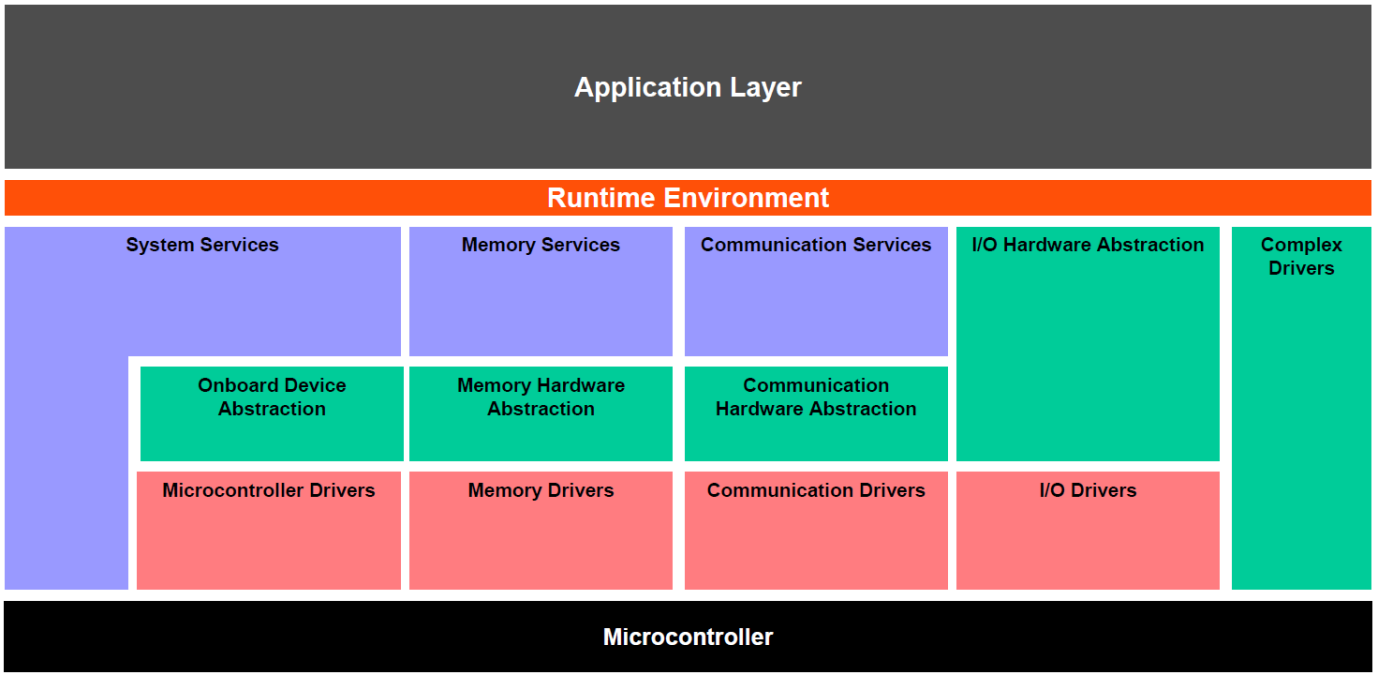
UDS Service	This refers to a UDS Service as defined in ISO14229-1
Callouts	Callouts are pieces of code that have to be added to the DCM during ECU integration. The content of most callouts is hand-written code, for some callouts the DCM configuration tool shall generate a default implementation that is manually edited by the integrator. Conceptually, these callouts belong to the ECU Firmware.

Abbreviation:	*Description:*
API	Application Programming Interface
BSW	Basic Software
CRC	Cyclic Redundancy Check
Dcm	Diagnostic Communication Manager
Dem	Diagnostic Event Manager
Det	Development Error Tracer
DID	Data Identifier
DTC	Diagnostic Trouble Code
ECU	Electronic Control Unit
EcuM	Electronic Control Unit Manager
ISO	International Standardization Organization
IUMPR	In Use Monitoring Performance Ratio
OBD	Onboard Diagnostics
OEM	Original Equipment Manufacturer (Automotive Manufacturer)
OS	Operating System
PID	Parameter Identification
RTE	Runtime Environment
SSCP	synchronous server call point
SW	Software
SW-C	Software Component
UDS	Unified Diagnostic Services
DDDID	Dynamically Defined Data IDentifier

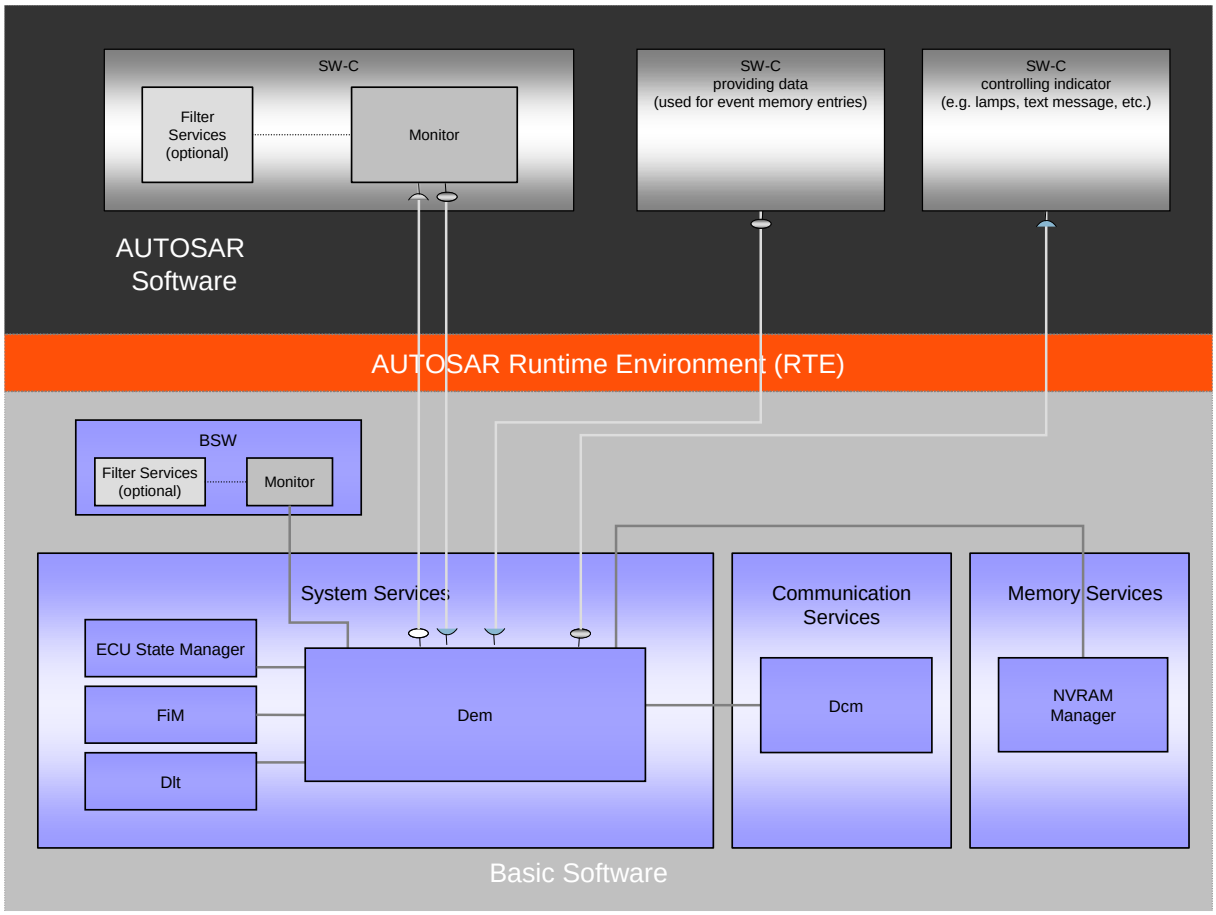
5 AUTOSAR System

5.1 Overview of Software Layers

The AUTOSAR platform is a layered architecture as illustrated in the below diagram. The AUTOSAR platform is categorized into Service Layer, ECU Abstraction Layer, Complex Device Drivers and Microcontroller Abstraction Layer.



5.2 AUTOSAR Dcm Module



6 Product Release Notes

6.1 Overview

This chapter provides the release information of Hyundai AutoEver Dcm Products, describing the features and limitations of the Dcm Release version.

6.2 Scope of the release

All content in this document applies only to the following Hyundai AutoEver Dcm modules.

Module	Autosar version	SWS version	Module version
Dcm	4.0.3	4.2.0	2.14.0

6.3 Change Log

6.3.1 Version 2.14.0.0

- Feature
  - Change DcmDslDiagRespMaxRespPend Type

Rationale	Change the type to allow the RespMaxNumRespend value to be set at least 255 times in order to keep the Pending response at least 20 minutes (255 times).
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

- Feature
  - Development of a feature to manage the Security Attempt Counter as a non-volatile value

Rationale	Security Attempt Counter shall be managed as a non-volatile value.
Impact on behavior	None
Impact on	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/

settings	DcmDspSecurityMaxAttemptCounterReadoutTime Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow/ DcmDspSecurityAttemptCounterEnabled Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow/ DcmDspSecurityGetAttemptCounterFnc Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow/ DcmDspSecuritySetAttemptCounterFnc
Required ASW actions	The operation of read/write the Security Attempt Counter to non-volatile memory must be performed in the application's Xxx_GetSecurityAttemptCounter()/Xxx_SetSecurityAttemptCounter().

➢ Bug

■ Fix the problem that the Security Attempt Counter does not exist by Security Level If DCM_STANDARD_SUPPORT = {DCM_ES95486_SUPPORT/DCM_ES95486_02_SUPPORT/ DCM_ES95486_50_SUPPORT/DCM_ISO14229_SUPPORT}.

Rationale	All Security Levels share a single Security Attempt Counter variable.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ Fix the point in time when the Security Attempt Counter is initialized to 0 from the start of the Security Delay Timer to after the Security Delay Timer expires.

Rationale	Initialize Security Attempt Counter to 0 when Security Delay Timer starts.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ Fix to respond with NRC 0x12 instead of NRC 0x31 when requesting Security Access RequestSeed for a Security Level that is set in DcmDslSubService and not set in DcmDspSecurityRow.

Rationale	NRC Error
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ Fix to respond with NRC 0x24 instead of responding positively when requesting Security Access SendKey after the Security Delay Timer expires.

Rationale	When starting Security Delay Timer, a variable that store the Security Access sequence is not initialized if DCM_STANDARD_SUPPORT = {DCM_ES95486_SUPPORT / DCM_ES95486_02_SUPPORT /DCM_ES95486_50_SUPPORT/DCM_ISO14229_SUPPORT}.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ Undesired deauthencation state transition is occured

Rationale	Among authentication timeout timer and P2,P3 timers, transtion condition is not considered at particular cases.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ Unexpected NRC(0x00) is occured with Authentication Service(29 01) using by invalid cert

Rationale	If the authentication service (29 01) was requested twice in a row with invalid cert, the NRC value was undefined.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

Undesired NRC error is occured that RequestFileTranfer, TransferData service is requested, after RequestDownload, TransferData is requeseted.

Rationale	Variable initialize condition is not considered between RequestFileTranfer, RequestDownload services.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.2 Version 2.13.0.0_HF1

➢ Bug

■ Undesired deauthencation state transition is occured

Rationale	Among authentication timeout timer and P2,P3 timers, transtion condition is not considered at particular cases.
Impact on behavior	None
Impact on settings	None

Required ASW actions	None

➢ Bug

■ Unexpected NRC(0x00) is occured with Authentication Service(29 01) using by invalid cert

Rationale	If the authentication service (29 01) was requested twice in a row with invalid cert, the NRC value was undefined.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

Undesired NRC error is occured that RequestFileTranfer, TransferData service is requested, after RequestDownload, TransferData is requeseted.

Rationale	Variable initialize condition is not considered between RequestFileTranfer, RequestDownload services.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.3 Version 2.13.1.0

➢ Bug

■ Improve generation time when there are more than 600 DIDs.

Rationale	Spend more than 30 minuites for generation of DID, with repeatly searching and sorting of signals which is referenced each DID.
Impact on behavior	None

Impact on settings	None
Required ASW actions	None

➢ Improvement

■ Fix to remain C-SAC(0x21) security level after jumping to boot loader.

Rationale	Set security level though the configuration index of C-SAC level is not stored at the first call to main after jumping to boot loader
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ Fix length check condition of variable data to include shorter data than maximum length.

Rationale	When 2E service is requested with the DID which is referencing variable data and shorter length than maximum, NRC 13 is occurred. It is because that the length check condition does not include shorter length than maximum.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ Fix to modify the generated value of DCM_CNR_USED_SHA1 according to its configuration.

Rationale	The name of generation rule is not matched with the name of configuration template
Impact on behavior	None
Impact on	None

settings	
Required ASW actions	None

➢ Bug

■ Fix to receive message of higher priority protocol during performing previous request.

Rationale	When receiving new request through higher priority protocol than previous protocol, the reception status of the incoming message is not completed. The next request through same or higher priority protocol is not received
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ Solve compile error of DID service in light platform caused from dependency of Nvm.

Rationale	Though the modifying of macro value which make Nvm header to be not included, the related functions or types are used. It makes compile error when using DID service in the environment of light platform without Nvm module
Impact on behavior	None
Impact on settings	Add item of DcmGeneral / NvmIntegrated
Required ASW actions	None

➢ Bug

■ Fix NOT_OK return of functional request after jumping to boot loader.

Rationale	After jumping to boot loader with no need of response, the flag of first request is set to FALSE, but the PduId of using protocol is not set. Therefor, because the PduId of following message is not matched with stored ID, return value of new request is NOT_OK
Impact on behavior	None

Impact on settings	None
Required ASW actions	None

➢ Bug

■ Improve race condition by adding delay between setting timer value and its flag of session timer.

Rationale	In multi-core environments and distributed tasks of Dcm, race condition of timer variables can cause sessions transition to default before timeout. Even the sequence of assigning value and flag is changed, those still have race condition in assembly level.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.4 Version 2.9.0.0_HF2

■ Improve race condition between timer value and its flag of session timer.

Rationale	In multi-core environments and distributed tasks of Dcm, race condition of timer variables can cause sessions transition to default before timeout. Even the sequence of assigning value and flag is changed, those still have race condition in assembly level.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.5 Version 2.13.0.0

➢ Bug

■ When service ID 29 is requested with invalid data length, NRC is not appropriate.

Rationale	Validation logic sequence with invalid data length is not correct
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ Routine Control (0x31) NRC logic sequence order is not correct.

Rationale	The NRC 13 as RoutineControl (0x31) request result is occured after RTE application layers.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ NRC 35(Invalid Key) occurs for third attempts of SecurityAccess(27) rather than NRC 36 (ExceededNumberOfAttempts), with re-initialization of extended session

Rationale	Attempt counter is reset in session transition to non-default session from non-default session
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Improvement

■ Improvement for no response for EcuReset(11 01) request after reprogramming.

Rationale	After reprogramming, the ApplUpdated flag is set to true, which makes Dcm start as WARM_START by the function Dcm_Internal_SetProgConditions(). Because the ResponseRequired flag is not 1, it has no response and changes session to programming session in Dcm_Internal_GblFirstCallToMain()
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Improvement

■ Add DET code in functions of message transmission to figure out point where error occurs.

Rationale	None
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Improvement

■ In condition ES95486-50, security level delay time, and NumAttDelay are not able to change from 180/3.

Rationale	It is designed to ensure that that parameter cannot be changed from 180/3, in condition ES95486-50
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Improvement

■ Excessive time is spent for Dcm generator when the hundreds of DIDs exist.

Rationale	Inefficient sorting logic while generating SW component
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.6 Version 2.12.0.0_HF1

- Bug
  - WDBI(WriteDataByIdentifier) NRC logic sequence order is not correct.

Rationale	The sequence of WDBI NRC 0x13 and 0x31 was reversed.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

- Bug
  - NRC 24 is not occured, when security access services ASK/OEUK are used mixed.

Rationale	Different security access level challenge does not cancelled former requested Seed.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.7 Version 2.12.0.0

- Bug
  - DCM_SECURE_FLASH_SUPPORT is always generated STD_OFF

Rationale	Logic error when generating DCM_SECURE_FLASH_SUPPORT
Impact on behavior	None
Impact on settings	DcmGeneral/DcmTransferSignatureNotWriteFlash
Required ASW actions	None

- Feature
  - Support ResumeFile(0x06) for RequestFileTransfer Service

Rationale	Support ResumeFile(0x06) for RequestFileTransfer Service refer to ISO spec
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

- Feature
  - Develop AuthenticationRole for RoutineControl

Rationale	Develop AuthenticationRole for RoutineControl
Impact on behavior	None
Impact on settings	DcmDspRoutineInfo/DcmDspRoutineRequestResOut/DcmDspRoutineRequestResOutSignal DcmDspRoutineInfo/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal DcmDspRoutineInfo/DcmDspRoutineStopOut/DcmDspRoutineStopOutSignal
Required ASW actions	None

6.3.8 Version 2.11.1.0

- Bug

■ Order reversals and omissions occur when passing the input output of routine control signals. (from 2.11.0.0._HF1)

Rationale	Due to a logic error in Endian (Little, Big) processing, when the RoutineControl Signal data is declared as 2byte type or more and the signal is delivered (Input, Output), order reversal and omission occur.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ Unexpected pending response is repeat, when Authentication service is requested(29 01)

Rationale	When the first pending response is sent, the following Authuntication function are performed regardless of whether TpTxconfirmation is complete.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ Unexpected NRC(0x00) is occured with Authentication Service(29 01)

Rationale	If the authentication service (29 01) was requested twice in a row, the NRC value was undefined.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ When Dcm_DcmTransferData is performed, undesired NRC response is occured.

Rationale	If the data with 0xFF Block Sequence Counter is transfer twice continuosly, undesired NRC response is occured.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

- Bug
  - P2*serverMax is not correct value, when response of the programming session is performed

Rationale	The resolution value of P2*serverMax is not applied, when Programming session is requested from Application.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.9 Version 2.11.0.0_HF1

- Bug
  - Order reversals and omissions occur when passing the input output of routine control signals.

Rationale	Due to a logic error in Endian (Little, Big) processing, when the RoutineControl Signal data is declared as 2 byte type or more and the signal is delivered (Input, Output), order reversal and omission occur.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.10 Version 2.11.0.0

➢ Feature

■ CSAC SHA1 is supported through Dcm callout function when using Crypto R44

Rationale	On the R40 platform, the Csm provided logic to perform Seed Padding with 0 during the usage of the SHA1 algorithm. However, on the R44 platform, this functionality is no longer provided by CSM. Therefore, it is implemented through a Callout function in Dcm.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Improvement

■ Developed jump to bootloader logic when DiagnosticSessionControl($10 02) is requested. (FBL 3.0)

Rationale	In the FBL 3.0, the program is stayed at FBL when there's request for DiagnosticSessionControl($10 02). Therefore, developed to respond positive or non-response at WARM-START and operate the S3timer.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ NRC occurs when using TransferData service.

Rationale	RequestDataLength includes the blockSequenceCounter, so when comparing it with BlockLength, 1 should be subtracted.
Impact on behavior	None
Impact on settings	None
Required ASW	None

actions	

➢ Improvement

■ Improve to prevent the occurrence of uint64 errors when the RequestFileTransfer service is not used.

Rationale	The uint64 variable error will not occur when RequestFileTransfer service is not used since the uint64 variable type is only necessary during the utilization of this service.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.11 Version 2.10.1.0

➢ Bug

■ Modification of DID Range command length validity judgment logic

Rationale	Length check failed (0x13 NRC) when requesting to write data of DID Range through WriteDataByIdentifier (2E) service. Modification of DID Range setting value reference logic.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Improvement

■ Improved shortTermAdjustment(0x03) command length validity judgment logic

Rationale	An improvement request was reflected so that the minimum controlState length for the shortTermAdjustment(0x03) command of the InputOutputControl(2F) service is ignored rather than judged as a set value.
Impact on behavior	None

Impact on settings	None
Required ASW actions	None

➢ Task

■ Added information to the User Manual regarding Request Download/Upload.

Rationale	In the case of RequestDownload and RequestUpload, a statement that the user must implement it himself was added to the User Callout that is called when a range check for the memory address is required.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.12 Version 2.10.0.0

➢ Feature

■ Add RequestFileTransfer Service

Rationale	Because of updating ES specification, RequestFileTransfer Service is developed.
Impact on behavior	None
Impact on settings	DcmConfigSet/DcmDsp/DcmDspRequestFileTransfer
Required ASW actions	When Port is used, RequestFileTransfer PPort configuration must be added.

6.3.13 Version 2.9.0.0_HF1

■ Improve session transition occurring before timeout, caused by timer race condition

Rationale	In multi-core environments and distributed tasks of Dcm, race condition of timer variables can cause sessions transition to default before timeout. For request of a service with extended session conditions, it responds with NRC 7F.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.14 Version 2.9.0.0

- Bug
  - Improved Security Access behavior

Rationale	When Security Access is in progress, when an invalid key is determined in the key authentication stage after seed transmission, it is not in the key authentication stage and the state is not initialized to the initial seed transmission stage.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

- Bug
  - InputOutputControlByIdentifier (2F) Add status processing logic

Rationale	When InputOutputControlByIdentifier (2F) is operated in the Async method, Condition check and Read Data status are requested as Pending from the beginning. Init processing is not possible when calling the App function.
Impact on behavior	None
Impact on settings	None
Required ASW	None

actions	

➢ Bug

■ Improved status processing logic when a negative occurs after requesting RoutineControl RequestResults (31)(03)

Rationale	RoutineControl RequestResults (31)(03) Status not initialized when negative occurs after request
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ Added minimum length processing filter logic for Routine Control Signal

Rationale	When sending less than the set length of the routine signal, overflow occurs and reset occurs
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Improvement

■ Remove FBL Progconditions configuration dependency

Rationale	Connection of Progconditions shared between FBL and RTSW varies depending on FBL version. The user is required to be able to select it according to the FBL being used. A setting has been added to allow users to select dependencies for using Connection. (FBL type selection setting required)
Impact on behavior	None
Impact on	Dcm / DcmConfigSet / DcmGeneral / Fbl Used Type

settings	
Required ASW actions	None

6.3.15 Version 2.8.0.0

- Feature
  - Add Authentication NRC(5A/5D) Callout

Rationale	SWP provide Callout for Authentication NRC 5A/5C.
Impact on behavior	None
Impact on settings	DsmDspAuthentication/DcmDspAuthenticationConnectionES/ DsmDspAuthenticationSettingAccessRightsFailedFunc DsmDspAuthenticationDeauthenticationFailedFunc DsmDspAuthenticationUsePort
Required ASW actions	Add Callout Function when using NRC 5A/5D with Authentication Service.

- Improvement
  - Fix Authentication default role

Rationale	If the role is not set when using the Authentication Service, it is generated as 0x00 and a negative NRC 34 (authenticationrequired) response occurs. Fix generator to generate role as 0xFF.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

- Improvement
  - Separate CSAC/Authentication precompile option

Rationale	Separate Authentication Service and CSAC precompile option so that Authentication Service can works without CSAC configuration.
Impact on	None

behavior	
Impact on settings	None
Required ASW actions	None

➢ Improvement

■ Change name of option Secure Flash Support

Rationale	The option name of DcmSecureFlashSupport is not clear, so name is changed to DcmTransferSignatureNotWriteFlash
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ Fixed $19 $1A service response NRC 0x31 in situations where a normal response is required

Rationale	$19 $1A service should not check FunctionalGroupIdentifier and DTCFormatIdentifier, but there was a logic to check them.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ When DTCs should not be displayed in $19 $42 service, DTCs are displayed, so it is fixed.

Rationale	Wrong parameter input for Dem_SetDTCFilter function when receiving $19 $42 service
Impact on behavior	None
Impact on settings	None

Required ASW actions	None

6.3.16 Version 2.7.1.0

- Feature
  - Authentication Service (NRC 58)

Rationale	When using Authentication Service, if ProofOfOwnership is failed, NRC 58 (Ownership verification failed) must be responsed.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

- Task
  - Fix Minor Version value

Rationale	Dcm Minor Version value is invalid.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Improvement

■ Add new NRC for ES95486-02 rev. 19

Rationale	For ES95486-02 rev.19, add NRC for application.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Delete dependency between Dcm 2.7.0.0 and Dem 3.4.0.0

Rationale	Dcm 2.7.0.0 have dependency about Dem 3.4.0.0 for J1979-2. So delete dependency when J1979-2 is not used.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.17 Version 2.7.0.0

➢ Feature

■ Secure Access SHA2 Support

Rationale	When Secure Access is processing, signature and C&R verify can be worked with SHA2 algorithm.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Authentication Service (Vendor Specific)

Rationale	When using Authentication service, service must be support the certificate that required in ES specification.
Impact on	None

behavior	
Impact on settings	None
Required ASW actions	None

■ J1979-2 specification development

Rationale	Request for J1979-2 specification development
Impact on behavior	None
Impact on settings	/AUTRON/Dcm/DcmConfigSet/DcmGeneral/DcmObdProtocolId (refer 7.1.1) /AUTRON/Dcm/DcmConfigSet/DcmDsp/DcmDspReadDTCInformation/ DcmDspReadDTCInformationSupportedObdUdsDtcSeparation (refer 7.1.5.21)
Required ASW actions	None

■ J1979 specification development

Rationale	Request for J1979 specification development
Impact on behavior	None
Impact on settings	/AUTRON/Dcm/DcmConfigSet/DcmDsp/DcmDspPid (refer 7.1.5.10) /AUTRON/Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl(refer 7.1.5.11) /AUTRON/Dcm/DcmConfigSet/DcmDsp/DcmDspTestResultByObdmid(refer 7.1.5.17) /AUTRON/Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo (refer 7.1.5.18)
Required ASW actions	None

6.3.18 Version 2.6.5.0

➢ Bug

■ Change the diagnostic behavior of the high Priority protocol (Client B) to operate normally when the high Priority protocol (Client A) preempts the low Priority protocol (Client A)

Rationale	After receiving a diagnostic request from the High Priority Protocol (Client B), the request's reception status remains in progress and the next request

	cannot be processed
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Task

■ Change the DcmDslDiagRespOnSecondDeclinedRequest configuration to unsupported

Rationale	The DcmDslDiagRespOnSecondDeclinedRequest configuration is unsupported
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.19 Version 2.6.4.0

➢ Bug

■ Fixed Minimum length Check logic error in WriteDataByIdentifier service.

Rationale	Error in Minimum length Check logic in WriteDataByIdentifier service
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ When performing the read dtc service, the variable that checks the sequence of security access is modified so that it is not initialized.

Rationale	When performing the read dtc service, the variable that checks the sequence

	of security access is initialized.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Bug

■ Changed to set FBL-related flags after reset according to programming session

Rationale	FBL-related flag is not set after reset according to programming session, so it is not transitioned
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.20 Version 2.6.3.0

■ Feauture

■ Improvement logic for validate SID and subfunction with white list.

Rationale	Currently logic for validate service id and subfunction id is almost same. It should be apply new logic and new white list data structure for customize the validation in Dsd submodule.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.21 Version 2.6.1.0

■ Improvement

■ Modify static code and generator code to keep programing session after reset.

Rationale	Currently, Dcm doesn't keep programing session after reset, the session is changed to default session. That impacts to some services not available in default session.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Improvement

■ Create new pre-compile and add it for Dem API and Dem header include, so that they are compiled only when using Dem service

Rationale	Compilation errors occur when Dcm is used in SWP that does not use Dem.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Improvement

■ In static code, modify logic to flexibly get value of lengthFormatIdentifier and maxNumberOfBlockLength.

Rationale	Currently, Dcm is using hard code to update value of lengthFormatIdentifier and maxNumberOfBlockLength (when returns positive response of service 0x34 – RequestDownload)
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

➢ Improvement

■ improvement of missing '>' in generator pdf file.

Rationale	missing '>' in generator pdf file
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.22 Version 2.6.0.1

■ Task

■ Update User Manual

Rationale	Add the user manual English version. Add the information for setting buffer size when using RXSWIN.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.23 **Version 2.6.0.0**

■ Feature

■ Add Authentication Service

Rationale	Add Authentication Service for Dcm R40
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.24 Version 2.5.2.0

- Improvement
  - Fix UNECE

Rationale	Fix UNECE
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

- Improvement
  - Update the static code with Autosar Dcm version R4.3. When default session transition, Communication control state and Control DTC state must be reset.

Rationale	In Dcm R40, when Dcm transit to default session, communication control state and control DTC state is not reset because Autosar R4.0 spec there's no clear expression about it. So Dcm need to be applied Autosar Dcm version R4.3 to make sure the Communication control state and control DTC state must be reset when Dcm transit to default session.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.25 Version 2.5.1.0

- BUG
  - Update generator to add validation for DcmDslBufferSize base on specifications standard support.

Rationale	PduLengthType is only supported as uint16, DcmDslBufferSize can be set up to a maximum of 65535. In the case of ES specification support, the max value can be set up to 4095.
Impact on	None

behavior	
Impact on settings	None
Required ASW actions	None

■ Improvement

■ Fix UNECE

Rationale	Fix UNECE
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.26 Version 2.5.0.0

■ Feature

■ Change cause of occurrence in 0x22 service

Rationale	In service 0x22 ReadDataByIdentifier, change logic return of NRC 0x13 and NRC 0x31 from ES document version REV16 to version REV17/21.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Improvement

■ Verify return NRC 0x72, when CheckProgrammingDependancy failed.

Rationale	In previous version, there is no test case when CheckProgrammingDependancy failed. So add new test case to verify this

	condition.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Feature

■ Update source code of service 0x22 ReadDataByIdentifier to support Read Data Length function return pending.

Rationale	In previous version, Read Data Length function of service 0x22 only return E_OK. Currently Read Data Length function can be returned pending. So source code need support Read Data Length function return pending.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Improvement

■ Verify critical section pair for UT

Rationale	Improvement source code for detecting missing exit protection after enter protection.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Improvement

■ Improvement generator when same input, the code order of generated output files should be same.

Rationale	In previous version, the code order of generated output files is different when

	using the same input files. Improvement generator to make sure the generated output file should be same when same input file.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.27 Version 2.4.0.0

- Feature
  - The CAN/CAN FD RX Buffer is developed to be able to hold up to 4095. If the length exceeds the limit, Generate error will occur.

Rationale	<Requirement> : Stop receiving messages and send a FC with overflow status when the First frame data length exceeds 4095byte. <ES Specifications> : There is a TC triggering transmission of 'overflow' FC when FF_DL exceeding ES95486-12 4095byte is received : Define the max. value of ES95486-00,02,50 FF_DL as 4095byte <ISO Specifications> : If ISO 15765-2 FF_DL exceeds available buffer size, stop receiving messages and send 'overflow' FC
Impact on behavior	None
Impact on settings	Generate error will occur if the length of Dcm/DcmConfigSet/DcmDsl/DcmBuffer/UDS_RX CAN RX Buffer is set above 4095
Required ASW actions	None

- Bug

■ Wrong NRC response when Send Key precedes Request Seed during the run of Secure Access

Rationale	When send Key is run before Request seed, NRC(0X13 Invalid Length) is sent out due to internal logic error while the correct message is NRC 0x24(requestSequenceError.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Improvement

■ Enhanced explanation related to Dcm_GetCertificationInfo API and deleted unnecessary API

Rationale	Dcm_GetCertHolderReferece API was deleted as the Certifcate Holder Reference provided from Version 2.3.18.0 is available through the existing Dcm_GetCertificationInfo API.
Impact on behavior	None
Impact on settings	GetCertHolderReference operation was deleted from DcmServices
Required ASW actions	Need to change to Dcm_GetCertificationInfo API if GetCertHolderReference API was used Improved explanation of Dcm_GetCertificationInfo API in UM

■ Improvement

■ Fixed item was reclassified to Changeable on after the library codes are made open

Rationale	Users are allowed to change an item that used to be a fixed one after public disclosure of the code
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Improvement

■ Error message will be printed when a value other than mandatory ones in ES specifications is set up

Rationale	If ES 95486 is supported, when values other than access lock time 180 and num att delay 3 are set, the generator is supposed to create error
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.28 Version 2.3.18.0

■ Bug

■ If SecureAccess 2.0 is in use, the Certificate Holder Reference should be provided to the application through API.

Rationale	The SWP of 5.4.2 SELF LOCK ACTIVATION REQUEST in ES95489-01 should provide the Certificate Holder Reference if SecureAccess 2.0 is in use.
Impact on behavior	None
Impact on settings	GetCertHolderReference operation was added to DcmServices
Required ASW actions	Need to use API referencing UM 11.1.2.1.1

■ Improvement

■ Eliminated AutoEver FBL dependency of the Secure Access

Rationale	Improvement was made to allow receipt of public key in Dcm when AutoEver FBL is not in use so the users can authenticate SecureAccess (before the improvement, they had to implement the public key in user callout by themselves).
Impact on behavior	None
Impact on	None

settings	
Required ASW actions	Dcm_GetPublicKey is not needed if Autoever FBL is not used when applying the Secure Access of Hyundai Automotive

6.3.29 Version 2.3.17.0

■ Improvement

■ Eliminated impact on Dcm Library (Library code open)

Rationale	Eliminated impact on Dcm Library (Library code open)
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Bug

■ Redressed the failure in S3 timer reload when multiple protocols are in use

Rationale	The S3 timer is not reloaded when multiple protocols are in use (failure in meeting specifications).
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.30 Version 2.3.16.0

■ Improvement

■ Improvement of coding convention for Cyber-Security

Rationale	Code improvements to comply with the UNECE Cyber Security regulations
Impact on	None

behavior	
Impact on settings	None
Required ASW actions	None

6.3.31 Version 2.3.15.0

■ Improvements

■ UM modification

Rationale	• Added warning when using True Random Generate with AutoEver HSM 2.0 (see Chapter 11.2.2.1.2) • Added explanation related to 8.3.13 OpStastus Pending • Added explanation of the AppDcm_GetRandomSeed function to the notes
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Improvements

■ Modified delivery of the local variable to parameters of the function in Dcm_DspRandomSeed() (to be able to support Security Access 1.0)

Rationale	If the setting o DCM_SECURITY_CRL_ENABLE is STD_OFF, compile error occurs (when Security Access 1.0 is in use)
Impact on behavior	Modified compile error
Impact on settings	None
Required ASW actions	None

■ Improvements

■ Fixed incorrect designation of Memory Section

Rationale	Redressed the missing or wrong (Const Data -> Variable Data section) designation of memory section for some variables in Dcm
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.32 Version 2.3.14.0

■ Improvements

■ Modified delivery of the local variable to parameters of the function in Dcm_DspRandomSeed()

Rationale	Local variable is delivered to parameters when the Csm_RandomSeedUpdate function is called within Dcm_DspRandomSeed()
Impact on behavior	Take an action not to cause damage to RAM if the CsmRandomSeedUpdate() function is run asynchronously. No impact as it is run synchronously in the current release
Impact on settings	None
Required ASW actions	None

■ Improvements

■ Changed Security Access lock time following revision of the ES specifications

Rationale	Improvement of ES specifications
Impact on behavior	180-second lock will be applied if login fails more than three times in Security access service, following the revision of the ES specifications
Impact on settings	None
Required ASW	None

actions	

■ Improvements

■ Addressed the abnormal Responsebyte in the event of start/stop/requestRoutineResults during Routine Control

Rationale	The variable length does not get initialized upon start of RoutineControl
Impact on behavior	When there is a sub function without signal in the event of Sub function Start/Stop/requestRoutineResults and another sub function is configured with variable length, if the sub function without signal is called and then the sub function without signal or a sub function whose signal type is fixed is called, responsebyte of such sub function is printed as same as the responsebyte of the signal configured as variable length Normal Case 1: Start in/out signal : fixed or variable length Stop in/out signal : fixed or variable length requestRoutineResults signal : fixed or variable length Normal Case 2: Start in/out signal : none or fixed Stop in/out signal : none or fixed requestRoutineResults signal : none or fixed ---------- Limitation: It is considered abnormal when both 1) and 2) conditions are met; 1) if there is a sub function without signal 2) if there is a sub function configured as variable length Abnormal Case 1: Start in or Start out signal: variable length Stop in/out signal : none requestRoutineResults signal : fixed

	Start -> Stop(abnormal response) -> requestRoutineResults (abnormal response) Abnormal Case 2: Start in/out signal : none Stop in/out signal : fixed requestRoutineResults signal : variable length Start -> Stop -> requestRoutineResults -> Start (abnormal response)-> Stop (abnormal response) Abnormal Case 3: Start in/out signal : fixed Stop in/out signal : variable length requestRoutineResults signal : none Start -> Stop -> requestRoutineResults -> Start (abnormal response)-> requestRoutineResults (abnormal response) Abnormal Case 4: Start in/out signal : none Stop in/out signal : none requestRoutineResults signal : variable length Start -> Stop -> requestRoutineResults -> Start (abnormal response)-> Stop (abnormal response)
Impact on settings	None
Required ASW actions	None

6.3.33 Version 2.3.13.0

■ Improvements

■ S3Server Timer functions in DefaultSession when it receives Functional TesterPresent

Rationale	Breach of ES95486-XX specifications. The S3Server Timer is supposed to function only in non-default session but it happened to function in default session when it receives Functional TesterPresent(3E 80)
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.34 Version 2.3.12.0

■ Improvements

■ Application of SecurityAccess 2.0 triggers memory violation with the local variable in CSM API elements

Rationale	When SecureAccess 2.0 is in use, the pointer delivered to CSM API is sent to the Dcm local variable (stack memory), being used as a determinant of the loop count in a for statement which serves to copy memory in CSM Task. At this time, Context Switching triggers change in stack memory values that the pointer is pointing at, making the number of loops increase, causing memory violation during memory copy
Impact on behavior	Controller error occurs if the memory used in other module (or application) is violated
Impact on settings	None
Required ASW actions	None

6.3.35 Version 2.3.11.1

■ Improvements

■ User Manual updated

Rationale	Engine condition(Ex. Engine Stop, IGN On)-driven limitations in the diagnostic service were added to the guide. Enhanced explanation of Callback by adding settings/sample codes
Impact on behavior	None
Impact on settings	None
Required ASW actions	Apply the Callback function depending on the environment of the controller

6.3.36 Version 2.3.11.0

- Improvements
  - Failure in processing requests received continuously at short intervals if the first request is Functional TesterPresent(suppressPosRspMsgIndicationBit = true)

Rationale	The first request after ECU reset is processed as a usual request, not allowing simultaneous processing of two requests
Impact on behavior	If Functional TesterPresent (suppressPosRspMsgIndicationBit = true) is received before or after receipt of a request, the request should be processed
Impact on settings	None
Required ASW actions	None

6.3.37 Version 2.3.10.0

- Improvements
  - In the event of Concurrent Testerpresent, MetaData processing fails

Rationale	If Concurrent Testerpresent takes place during Ethernet diagnosis, MetaData for the Testpresent is processed, without retaining the previous MetaData. This leads to failure in recognition of response to the request as the response to Testerpresent is sent out. Due to this, reprogramming may fail.

	※ Concurrent Testerpresent : When TesterPresent is requested to a functional address, SuppressPosRspMsgIndicationBit is true (subfunction: 0x80).
Impact on behavior	If Concurrent TesterPresent is received during processing of a request, ignore MetaData processing for the TesterPresent and send response to the request
Impact on settings	None
Required ASW actions	None

6.3.38 Version 2.3.9.0

- Improvements
  - If Xxx_RequestResults request is pending during RoutineControl, negative response is generated

Rationale	If RequestResults subfunction of Routine Control is used and User Callout returns DCM_E_PENDING, negative response is generated instead of pending response. If E_FORCE_RCRRP is returned, negative response is generated after one pending response. If User Callout returns E_OK, E_NOT_OK, or E_FORCE_RCRRP, however, it runs as usual.
Impact on behavior	Modify to generate pending response if User Callout returns DCM_E_PENDING or E_FORCE_RCRRP, as in the specifications
Impact on settings	None
Required ASW actions	None

6.3.39 Version 2.3.8.1

- Improvements
  - User Manual updated

Rationale	Deleted ‘Masked by 0x80’ stated in Parameter(Out) and Description of 7.3.1.2 GetSesCtrlType as it is wrong
Impact on behavior	None

Impact on settings	None
Required ASW actions	None

■ User Manual updated

Rationale	Required changes are missing in the guide to implement AppDcm_GetSeed_L9() related to the change in the random generation logic which is incorporated into Dcm Version 2.3.8.0
Impact on behavior	If HSM is applied, change the random generation logic to run as PRNG after the first TRNG
Impact on settings	None
Required ASW actions	Need to refer to AppDcm_GetSeed_L9() in 10.2.2.1.2.1 and 10.2.2.1.2.2 to change the random generation logic to run as PRNG after the first TRNG if HSM is applied.

6.3.40 Version 2.3.8.0

■ Improvements

■ If HSM is in use, change the SecureAccess CnR random generation from HSM_TRNG to HSM_PRNG

Rationale	If HSM is applied, change the random generation logic to run as PRNG after the first TRNG
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Added the Requested Download overflow check logic

Rationale	If Secure Flash Support was applied, the memory size value can overflow as the signature information is added when receiving Request Download Request from Client
Impact on behavior	None

Impact on settings	None
Required ASW actions	None

■ Modified the item in breach of MISRA C

Rationale	Modified the item in breach of MISRA C
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.41 Version 2.3.7.0

■ Improvements

■ Developed a re-send logic to enhance OTA background transmission

Rationale	Developed logic in line with the change in the specifications which mandates re-sending of UDS Request if execution controller generates negative or no response during background transmission
Impact on behavior	If an identical block sequence number is received during background transmission, process it according to the OTA specifications
Impact on settings	None
Required ASW actions	Need to change the Dcm_WrriteMemory logic in OTA Application Application can know about arrival of the same block sequence number because Dcm_WriteMemory is called delivering the same Memory Address as parameters when TransferData arrives with the same block sequence number. Logic should be run accordingly.

6.3.42 Version 2.3.6.0

■ Improvements

■ Modified the item in breach of MISRA C

Rationale	Modified the item in breach of MISRA C
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.43 Version 2.3.5.0

- Improvements
  - Synchronized Reentrant and Can be invoked Concurrently settings

Rationale	Change the Dcm_MainFunction Can be invoked Concurrently setting to false
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

  - Modified compile warning

Rationale	Modified compile warning
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

  - MISRA C Verification

Rationale	Verification of MISRA-C
Impact on behavior	None
Impact on settings	None

Required ASW actions	None

6.3.44 Version 2.3.4.0

■ Improvements

■ Fixed messaging of abnormal response when negative response was generated in the Silent Com state

Rationale	If diagnostic message is received in the Silent Com state, negative response is generated. This is because the negative response processing function was run while the mode is yet to switch to Full Com and Silent Com processing is yet to be done
Impact on behavior	Modified to switch to Full Com so as to generate negative response successfully if negative response is generated for a diagnostic message received in the Silent Com state
Impact on settings	None
Required ASW actions	None

■ Changed memory size of Dcm_ReadMemory

Rationale	In AUTOSAR specifications, the memory address and the memory size of ReadMemory are defined as uint32 but it can be 1 byte on the current platform
Impact on behavior	Changed to 4byte as in the AUTOSAS specifications
Impact on settings	None
Required ASW actions	None

■ Fixed generation of outofRange negative response when DID is 256 or above

Rationale	OutofRange negative response is generated when DID is 256 or above. The data type of DID is generated as uint8
Impact on behavior	Changed the data type of DID generated

Impact on settings	None
Required ASW actions	None

■ Modified MISRA C mandatory item

Rationale	Modified the item in breach of the MISRA C mandatory rule
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.45 Version 2.3.3.0

■ Improvements

■ MISRA C Verification

Rationale	Verification of MISRA-C
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.46 Version 2.3.2.1(Patch)

■ Improvements

■ In the event of Concurrent Testerpresent, MetaData processing fails

Rationale	If Concurrent Testerpresent takes place during Ethernet diagnosis, MetaData for the Testpresent is processed, without retaining the previous MetaData. This leads to failure in recognition of response to the request as the response to Testerpresent is sent out. Due to this, reprogramming may fail.

	※ Concurrent Testerpresent : When TesterPresent is requested to a functional address, SuppressPosRspMsgIndicationBit is true (subfunction: 0x80).
Impact on behavior	If Concurrent TesterPresent is received during processing of a request, ignore MetaData processing for the TesterPresent and send response to the request
Impact on settings	None
Required ASW actions	None

6.3.47 Version 2.3.2.0

■ Improvements

■ Modified the stopping of S3 timer if the result of TpTxConfirmation is NOT_OK

Rationale	If notification result of Dcm_TpTxConfirmation is NOT_OK, the S3 timer stops running, causing transition to a default session
Impact on behavior	If multi-frame fails to receive FC, notification result of Dcm_TpTxConfirmation becomes NOT_OK, reigniting the S3 timer.
Impact on settings	None
Required ASW actions	None

6.3.48 Version 2.3.1.0

■ Improvements

■ Failure in receiving Concurrent TesterPresent

Rationale	Failure in processing Concurrent TesterPresent request which is received after transmission of 0x78 Pending Response
Impact on behavior	In the event of Concurrent TesterPresent error, transition from a non-default session to a default session takes place cancelling the request under processing
Impact on settings	None
Required ASW	None

actions	

6.3.49 Version 2.3.0.0

- New features
  - Development of Dcm_GetVin()

Rationale	It was developed upon the request for Dcm_GetVin which is a mandatory interface of the DoIP module. 4.1 spec of the relevant feature was applied in advance to implement the API
Impact on behavior	None
Impact on settings	Added setting DcmVinRef [/AUTOSAR/Dcm/DcmConfigSet/DcmGeneral/DcmVinRef]
Required ASW actions	None

  - Deleted the ECU Forced Reset feature

Rationale	It was requested to delete the Ecu Reset processing feature during OTA reprogramming
Impact on behavior	None
Impact on settings	Deactivated existing setting Dcm/DcmConfigSet/DcmGeneral/DcmForcedEcuReset
Required ASW actions	None

- Improvements
  - Dcm_GaaReadMemRngConfig symbol error occurs during OTA-related configuration

Rationale	Fixed the compile error which is generated if ReadDatabyAddress is not configured when TransferData service(0x36) is in use
Impact on behavior	None
Impact on settings	None
Required ASW	None

actions	

■ Modified the logic related to the Secure Access 2.0 features

Rationale	Modified the settings related to the Secure Access 2.0 features 1. Modified the logic that identifies Secure Access 2.0. Changed the basis of determination from request msg length to ADR size in settings 2. Modified the logic to get date. Currently, the date to determine validity of certification is read from Dcm after application saves the date information in Nvm => Modified this to read from callback function by connecting port
Impact on behavior	None
Impact on settings	Added setting DcmDspCallbackPresentDate [/AUTOSAR/Dcm/DcmConfigSet/DcmDsp/DcmDspCallbackPresentDate]
Required ASW actions	None

■ Modified the logic to validate public key identifier for CRL issuer among Secure Access 2.0 features

Rationale	Modification of exponent in public key during validation of public key identifier
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Modified to allow receipt of signature block while using OTA

Rationale	Modified to allow receipt of signature block in addition to the memory size requested during request download when Secure Flash Support is set to True
Impact on behavior	None
Impact on settings	Added setting DcmSecureFlashSupport [/AUTOSAR/Dcm/DcmConfigSet/DcmSecureFlashSupport]

Required ASW actions	None

6.3.50 Version 2.2.0.0

- New features
  - Developed a Security Access 2.0 feature

Rationale	To support run of a Security Access 2.0 feature
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

- Improvements
  - N/A

6.3.51 Version 2.1.1.0

- New features
  - N/A
- Improvements
  - Changed configuration item properties to make code publicly available

Rationale	Changed configuration item properties to make code publicly available
Impact on behavior	None
Impact on settings	Property of DcmRespondAllRequest was changed to FIXED
Required ASW actions	None

6.3.52 Version 2.1.0.0

- New features

- N/A

■ Improvements

- Changed configuration item properties to make code publicly available

Rationale	Changed configuration item properties to make code publicly available
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.53 Version 2.1.0

■ New features

- Developed support for DcmDspDataBlockIdRef setting

Rationale	Add a feature to allow DCM to access NVM block directly for Data Read/Write by using DcmDspDataBlockRef if Read/WriteDataByIdentifier(0x22/0x2E) is in service
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

- Added Meta Data processing for the new EthDiag feature

Rationale	Developed processing of Source Address and Target Address, the information required for the new EthDiag feature during Ethernet communication, through Metadata
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Improvements

■ N/A

6.3.54 Version 2.0.0

■ New features

■ Processed prioritization of the ECU reset service in accordance with the revised OTA specifications

Rationale	Reset should be made available in the event of a fatal server failure during OTA reprogramming
Impact on behavior	Servers can be reset upon the ECUReset(0x11 0x81) request if DcmForcedEcuReset check box is ticked on no matter what service is being processed
Impact on settings	1. Added new setting - Dcm/DcmConfigSet/DcmGeneral/DcmForcedEcuReset
Required ASW actions	None

■ Improvements

■ The problem of failed preemption of a lower priority protocol during processing of protocol preemption requests with different priorities and the failure in operation of preemption timer

Rationale	1. When several protocols are given different priorities and run for preemption, those with lower priority are left out of preemption. 2. The preemption timer does not work when a request with a higher priority is being processed during preemption of high priority protocols
Impact on behavior	Modify to allow successful preemption among protocols and optimize the condition for message processing
Impact on settings	None
Required ASW actions	None

■ The failure in automatic update of memory address when TransferData(0x36) service is requested continuously

Rationale	When TransferData service is requested following processing of RequestDownload, MemoryAddress should increase too, in line with

	blockSequenceCounter but the same MemoryAddress is called
Impact on behavior	Modified to update the MemoryAddress to be used next in line with blockSequenceCounter
Impact on settings	None
Required ASW actions	None

■ The failure in restart of S3Timer in the event of error during receipt of multi-frame request message

Rationale	The failure in restart of S3Timer in the event of error in Consecutive Frame during receipt of multi-frame request message
Impact on behavior	Modified to restart S3Timer in the event of error in Consecutive Frame during receipt of multi-frame request message
Impact on settings	None
Required ASW actions	None

6.3.55 Version 1.9.5

■ New features

■ N/A

■ Improvements

■ The failure in initialization of OPSTATUS values in DID after processing of DID Max Pending

Rationale	After processing of Max Pending, the OPSTATUS value of individual DID does not change to DCM_OPSTATUS_INITIAL
Impact on behavior	After 5 MAX PENDING, respond with $10 NRC When $2F DID is requested shortly after that, positive response is wrongfully generated without running the service
Impact on settings	None
Required ASW actions	None

- There is no issue in terms of feature but sorting was done to prevent change in the ucPduIdStatusMask values

Rationale	As ucPduIdStatusMask was not sorted for every generation, values were changed.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

- For the continued Seed Request in the QZN04 specifications, modification was done to return the same Seed Value as Positive Response

Rationale	For continued Seed Request, change should be made to return the same Seed Value together with positive response. When Error Count is 2, NRC 36 should be returned
Impact on behavior	Send positive response with the same Seed Value for continued Seed Request. If the error count is 2, respond with NRC 36.
Impact on settings	None
Required ASW actions	None

6.3.56 Version 1.9.4

- New features
  - N/A

- Improvements
  - Compile error when DcmDspStartRoutineFnc is in use

Rationale	Compile error occurs when using Cdd as the functional prototype that was

	input as DcmDspStartRoutineFnc is not found. Note : No error if Routine Control is used through RTE.
Impact on behavior	Change to allow use of DcmDspStartRoutineFnc
Impact on settings	None
Required ASW actions	None

■ When requesting Routine control Request result, if the App responds after pending, all additional data is printed as 0

Rationale	When Routine control(SID 31) Request result was requested, immediate response without pending is reflected on User Response buffer update. Yet if the App responds after pending, it is not updated in the User Response buffer and returns 0x00 as a response
Impact on behavior	Change to generate normal response even after Request result Pending is requested
Impact on settings	None
Required ASW actions	None

■ In RoutineControl, VARIABLE_LENGTH option does not work in Stop and Request result sub-function excluding Start

Rationale	VARIABLE_LENGTH was not reflected on Dcm_GaaRoutineSignalOutData during Generate
Impact on behavior	Changed to incorporate VARIABLE_LENGTH
Impact on settings	None
Required ASW actions	None

■ Failure in creation of Signal Type UINT16 and UINT32 of the sub-service request result in

RoutineControl

Rationale	The data of Dcm_GaaRoutineSignalOutData16 and Dcm_GaaRoutineSignalOutData32 was not incorporated during Generate
Impact on behavior	Change to incorporate the data of Dcm_GaaRoutineSignalOutData16 and Dcm_GaaRoutineSignalOutData32
Impact on settings	None
Required ASW actions	None

6.3.57 Version 1.9.3

- New features
  - N/A

- Improvements
  - Applied AutoEver library in Security Level 21 ETAS library

Rationale	Need to apply AutoEver library in Security Level 21 ETAS library
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.58 Version 1.9.2

- New features
  - N/A

- Improvements
  - Modified RoutineControl stop sub-function DataIn

Rationale	In SID 31 Routine Control, Datain1 value was not sent properly to AppDcm_Stop_CalibrationActuator
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Added RemainUnlockCondition

Rationale	To retain the Security Level during transition from Extended Session -> Security Access -> Extended Session after CGW Unlock, change to specification to retain Security Level amid change in sessions was requested. To meet the specifications, it is needed to develop the changes to Secure Access
Impact on behavior	If Remain Unlock Condition is set, the security level remains during transition of Extended Session -> Security Access -> Extended Session
Impact on settings	Support Remain Unlock Condition setting in DcmGeneral
Required ASW actions	None

■ Applied F1KM HSM

Rationale	Added HSM, change in Dcm and guides for F1KM
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.59 Version 1.9.1

■ New features

■ N/A

■ Improvements

■ Generate error due to broken Dcm.exe file

Rationale	1. Generate error occurs due to broken Dcm.exe file. It should be newly created and added. 2. Compile error occurs due to inclusion of INCLUDE, a temporary code for Dcm.template. It should be deleted and released
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Modified RTRT dynamic verification Dcm

Rationale	Additional change is needed due to RTRT dynamic verification (make change and eliminate duplicate define to meet the DataType AUTOSAS specifications)
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.60 Version 1.9.0

■ New features

■ N/A

■ Improvements

■ Change to the priority of SID31 Subfunction NRC

Rationale	Need to change the sequence of NRC check to be able to check only subfunctions which are not SID 0x31 in line with the ES95486 specifications
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Support for the QZN04 specifications

Rationale	Need to make additional change in line with the review of the QZN04 Diag specifications 1) Security Access : Apply QZN04 specifications for fail counting 2)ComCommunication: Apply the QZN04 specifications to subfunc 01(enableRxAndDisableTx) and 02(disableRxAndEnableTx) behaviors 3) ECUReset: Apply the QZN04 specifications to be able to use subfunc 01, 02 and 03 only
Impact on behavior	Run diagnostic service in accordance with the QZN04 specifications when setting up DCM_QZN04_SUPPORT
Impact on settings	DCM_QZN04_SUPPORT in the standard support of DcmGeneral
Required ASW actions	None

6.3.61 Version 1.8.0

■ New features

■ N/A

■ Improvements

■ Change the DcmTimStrP2(Star)ServerAdjust PDF Max value based on the latest specifications

Rationale	Change the DcmTimStrP2(Star)ServerAdjust PDF and DcmDspSessionP2(Star)ServerMax Max values based on the ASR4.3.0 specifications
Impact on behavior	None
Impact on settings	Setup of the DcmTimStrP2(Star)ServerAdjust and DcmDspSessionP2(Star)ServerMax above the max values is not possible. Min ~ Max value of DcmTimStrP2ServerAdjust 의 Min ~ Max value - AUTOSAR 4.3.0 : 0 ~ 1 Min - Max values of DcmTimStrP2StarServerAdjust - AUTOSAR 4.3.0 : 0 ~ 5 Min - Max values of DcmDspSessionP2ServerMax - AUTOSAR 4.3.0 : 0 ~ 1 Min - Max values of DcmDspSessionP2StarServerMax - AUTOSAR 4.3.0 : 0 ~ 100
Required ASW actions	None

■ Support of ES95486-50 specifications

Rationale	Need to modify to support the ES95486-50 specifications
Impact on behavior	Run diagnostic service in accordance with the ES95486_50 when setting up DCM_ES95486_50_SUPPORT
Impact on settings	DCM_ES95486_50_SUPPORT in the standard support of DcmGeneral
Required ASW actions	None

■ Indication Callback ASR 4.3.0 was applied

Rationale	Need to apply the calling location of manufacturer indication callback to ASR 4.3.0 in advance
Impact on behavior	Follow the below sequence for verification when receiving Request message: 1. Verification of Manufacturer permission (Call of the manufacturer interface indication operation) 2. Verification of the SID

	3. Verification of the Diagnostic Session 4. Verification of the Service Security Access levels 5. Verification of the Supplier permission (Call of the Supplier interface indication operation) 6. Verification of the Mode rules for service IDs
Impact on settings	None
Required ASW actions	None

6.3.62 Version 1.7.3

- New features
  - N/A

- Improvements
  - Support of the ES95486-02 specifications

Rationale	Need to modify to support the ES95486-02 specifications
Impact on behavior	Run diagnostic service in accordance with the ES95486_02 when setting up DCM_ES95486_02_SUPPORT
Impact on settings	DCM_ES95486_02_SUPPORT in the standard support of DcmGeneral
Required ASW actions	None

6.3.63 Version 1.7.2

- New features
  - N/A

- Improvements
  - Change in SecurityAccess examples

Rationale	Change in examples during new release of CSM
Impact on behavior	None

Impact on settings	None
Required ASW actions	Need to modify application referring to sample codes when using SecurityAccess

- Apply the limitation of RH850 F1K ICUS(HSM)

Rationale	Asynchronous method is used to create Seed through HSM
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.64 Version 1.7.1

- New features
  - N/A

- Improvements
  - Improved Compile Warning

Rationale	Improved Compile Warning
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.65 Version 1.7.0

- New features
  - N/A

■ Improvements

■ Applied the new CSM security module

Rationale	With introduction of the new CSM, sample codes in L1 and L9 were changed and CSM API was changed for L21. This led to changes in Dcm internal codes. *The new CSM security module It includes existing CAL module features and the HSM feature that generates true random (see the CSM manual for MCU that HSM supports).
Impact on behavior	None
Impact on settings	None
Required ASW actions	See Appendix 10.2 Sample Codes

■ Support SAEJ1939-73 DTC format for the ReadDTCInformation service

Rationale	Change to support DTCFormatIdentifier 0x4 for response messages such as sub service reportNumberOfDTCByStatusMask in ReadDTCInformation
Impact on behavior	None
Impact on settings	DemGeneral / DemTypeOfDTCSupported = DEM_DTC_TRANSLATION_J2012DA_FORMAT_04
Required ASW actions	Need to modify the DemTypeOfDTCSupported setting

6.3.66 Version 1.6.0

■ New features

■ N/A

■ Improvements

■ For Dcm SW-Component, RTE Warning [WRN 103: There is no execution context information of RunnableEntity] was improved

Rationale	Warning was generated as Synchronous Server Call Point of Dcm R Port

	connected to Runnable of Dcm SW-Component (SSCP hereinafter) was not found in Rte settings. As the actual behavior is calling of Runnable in the form of Rte_Call_~ within Dcm MainFunction, there was no negative impact on the behavior. Avoid Rte Warning by mapping SSCP of Runnable in SW-C to Dcm MainFunction Task in Bsw.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.67 Version 1.5.2

- New features
  - N/A

- Improvements
  - Partially applied to AUTOSAR_SWS_DiagnosticCommunicationManager 4.2.2 DcmDslDiagRespMaxNumDiagResp

Rationale	Applied the higher version of AUTOSAR_SWS_DiagnosticCommunicationManager (AUTOSAR Dcm hereinafter) 4.2.2 to DcmDslDiagRespMaxNumDiagResp settings for the interest of Fail Safety
Impact on behavior	If DcmDslDiagRespMaxNumDiagResp is set to 0xFF, - AUTOSAR Dcm 4.0.3 (old): No Limit. Infinite Response Pending - AUTOSAR Dcm 4.2.2 (current): 255 times of Response Pending followed by General Reject (NRC10)
Impact on settings	None
Required ASW actions	None

6.3.68 Version 1.5.1

- New features
  - N/A

- Improvements
  - Additional creation of Security Level macro constant in Rte_Dcm_Type.h

Rationale	The macro constant for Security Level was changed as below. - DCM_SEC_LEV_LOCKED (0x00) : Existing - DCM_SEC_LEV_ALL (0xFF) : Existing - DCM_SEC_LEV_L1 (0x01) : Existing - DCM_SEC_LEV_L9 (0x09) : Added since Dcm 1.5.1 - DCM_SEC_LEV_L21 (0x21) : Added since Dcm 1.5.1
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.69 Version 1.5.0

- New features
  - N/A

- Improvements
  - Fixed the configuration error of creation of Client-Server Interface Operation in wrong sequence (see analytics of SWP change)

Rationale	Generator logic error - The sequence of keys for Dcm settings to be saved as values in hash in the generator logic (1, 2, 3, ..., 9, 10, 11, ...) is partially different from the

	sequence of keys that fetch hash values (1, 10, 11, ..., 19, 2, 20,...)
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Update of Seed-Key(L1) and Advanced Seed-Key(L9) Sample Code

Rationale	Updated the sample code for Cal module update and greater code stability
Impact on behavior	None
Impact on settings	None
Required ASW actions	The sample code in the Appendix to this document was applied to the algorithm of Seed-Key(L1) and Advanced Seed-Key(L9)

■ Deletion of the DcmAutronFblSecureLibUsed parameter set up to use the C-SAC (L21) algorithm and provision of the Dcm_GetPublicKey() Callout function for users to be able to provide PublicKey when implementing Fbl by themselves

Rationale	When using AutoEver Fbl in a controller using C-SAC, the public key provided by AutoEver Fbl must be used unconditionally, so the DcmAutronFblUsed setting includes the meaning of the DcmAutronFblSecureLibUsed setting. - DcmAutronFblUsed (=true) : Use AutoEver Fbl - DcmAutronFblSecureLibUsed (=*true*) : Use PublicKey in AutoEver Fbl to authenticate C-SAC when using AutoEver Fbl
Impact on behavior	Compatible with integration_Dcm version 1.0.9 or higher
Impact on settings	Delete DcmAutronFblSecureLibUsed setting
Required ASW actions	If the controller to which the C-SAC (L21) algorithm is applied does not use AutoEver Fbl and implements Fbl itself (DcmAutronFblUsed = false), PublicKey must be provided to the platform through the Dcm_GetPublicKey() Callout function

■ Dcm_GetRandomSeed() Callout function was provided so that users can update RandomSeed when using C-SAC (L21) algorithm

Rationale	When using the C-SAC (L21) algorithm, the RandomSeed can be updated through the Callout function. To this end, the randomness of the seed was increased
Impact on behavior	- Upon the first request for SecurityAccess (27 41), the logic to update RandomSeed will be selectively applied to the Callout function - Compatible with integration_Dcm version 1.0.9 or higher
Impact on settings	None
Required ASW actions	Applied RandomSeed update by referencing Appendix during implementation of C-SAC algorithm

■ Change Dcm_GetCertificationInfo() to meet the AUTOSAR standard (in the case of C-SAC-applied controller, review is mandatory)

Rationale	Update Dcm_GetCertificationInfo() that has been used in the form of CDD in the C-SAC-applied controller up until Dcm version 1.5.0 with the followings so as to meet the AUTOSAR standard - Rte Port Interface supported - Return Value was changed to Std_ReturnType - Change the name of transfer factor type: Dcm_Certification_InfoType → Dcm_CertInfoType
Impact on behavior	None
Impact on settings	None
Required ASW actions	When implementing the C-SAC algorithm, make sure to review and apply the changes below, for codes that use Dcm_GetCertificationInfo() of Dcm 1.5.0 or earlier version - Use of the Rte Port Interface method is recommended - Use the changed Return Value - Applied the changed name of transfer factor

6.3.70 Version 1.4.1

- New features
  - N/A

- Improvements
  - Fixed the wrong creation of InputOutputControlByIdentifier Operation

Rationale	Some operation creation errors of InputOutputControlByIdentifier service occur due to incorrect Hash Sorting of configuration file within generation logic
Impact on behavior	Improvement of Generator
Impact on settings	None
Required ASW actions	None

6.3.71 Version 1.4.0

- New features
  - N/A

- Improvements
  - Changed StopDiagnosticSession (SID20) negative response General Reject (NRC10) to be handled in Application area when DcmDspSessionForBoot of ProgrammingSession is DCM_NO_BOOT

Rationale	See ES95486-00. NRC10 (General Reject), the negative response of StopDiagnosticSession (20 hex), is created on the condition of "Reprogramming routine is not completed yet". As this cannot be interpreted as “Unconditional for ProgrammingSession", application has to make decision to implement the General Reject negative response
Impact on behavior	None

Impact on settings	None
Required ASW actions	Implement NRC10 (General Reject), the negative response of the StopDiagnosticSession by referencing Appendix

■ Fixed the configuration error of creation of Client-Server Interface Operation in wrong sequence

Rationale	Generator logic error. The sequence of pushing configuration parameters into internal stack of generator (1, 2, 3, ..., 9, 10, 11, ...) was different from the sequence of accessing stack values to create generation code (1, 10, 11, ..., 19, 2, 20,...), leading to the problem
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ After Response Pending (0x78), suppressPosRspMsgIndicationBit will not be considered and Positive/Negative Response will be processed

Rationale	The following statement will be applied to ES95486-00 specifications. "When requestCottectlyReceived-ResponsePending (NRC = 78 hex) response code is used, the server shall always send a final response (positive or negative) independent of the suppressPosRspMsgIndicationBit value."
Impact on behavior	Prerequisite: AUTOEVER Fbl in use, DcmSendRespPendOnTransToBoot true In the above condition, when a Programming Session transition request is made with the suppressPosRspMsgIndiciationBit true condition, Response Pending (NRC 78) is sent and the positive response is processed
Impact on settings	None
Required ASW actions	None

■ Change DcmDsdSidTabSessionLevelRef, DcmDsdSidTabSecurityLevelRef, DcmDsdSubServiceSecurityLevelRef, and DcmDsdSubServiceSessionLevelRef settings to be user configurable

Rationale	Users can set and use services and sub-services according to the controller specifications, reducing risks such as re-deployment of platform
Impact on behavior	None
Impact on settings	Configuration considering SRS is not needed for the followings DcmDsdSidTabSessionLevelRef DcmDsdSidTabSecurityLevelRef DcmDsdSubServiceSecurityLevelRef DcmDsdSubServiceSessionLevelRef
Required ASW actions	Set up the configurable items above to meet the ES95486-00 and controller specifications (see Appendix)

6.3.72 Version 1.3.1

■ New features

■ N/A

■ Improvements

■ Process ConditionsNotCorrect negative response (NRC22) when diagnostic service cannot be processed due to 'Critical normal mode'

Rationale	If Supported Negative Response Codes of ES95486-00 diagnostics service includes [Use when the server is in a critical normal mode activity and ~~] the Application should process the negative response. Relevant to SID28 CommunicationControl, SID29 EnableNormalMessageTransmission, and SID 85 ControlDTCSetting services
Impact on behavior	None
Impact on settings	None

Required ASW actions	(See Appendix) If the current controller cannot process SID28 CommunicationControl, SID29 EnableNormalMessageTransmission, and SID 85 ControlDTCSetting services, return NRC22 (*ErrorCode = 0x22) together with negative response (E_NOT_OK) within ServiceRequestSupplierNotification_Indication().

■ Update of the User Defined Service Function Guide

Rationale	As the number of use cases in which companies implement their own services increases recently, the need to reinforce the User Defined Service Function guide provided in Chapter 7.3.1.15 of the Dcm User Manual increases
Impact on behavior	None
Impact on settings	None
Required ASW actions	If users register and use a User Defined Service Function on their own other than the diagnostic service provided by the platform, it is necessary to check whether the cautions in the guide are followed

■ DLT not supported

Rationale	Reflected AUTOSAR Diagnostic Communication Manager 4.2.2 specifications - DLT not supported
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Fixed the Generator error related to RoutineControl Service

Rationale	Regarding RoutineControl, compile error during configuration was improved as below. 1) Compile error occurs if UINT8 is not found among DcmDspRoutineSignalType, the configuration parameters of DcmDspRoutineStartOutSignal, DcmDspRoutineStopOutSignal, and

	DcmDspRoutineRequestResOutSignal container 2) Fixed the compile error that occurred when using both containers with DcmDspRoutineStopInSignal when the identifier of the DcmDspRoutine container with DcmDspRoutineFixedLength set to false is smaller than the identifier value of the DcmDspRoutine container with DcmDspRoutineFixedLength set to true
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

■ Callout Function 'Dcm_GetProgConditions()' and 'Dcm_SetProgConditions()' Description was added

Rationale	Need to improve the manual with related details as inaccurate use of Callout Function may cause a problem
Impact on behavior	None
Impact on settings	None
Required ASW actions	Callout Functions Dcm_GetProgConditions() and Dcm_SetProgConditions() cannot be changed if AutoEver FbI is in use

■ Initialize the PROG_CONDITIONS area after reprogramming

Rationale	There is no logic to initialize ApplUpdated flag in RTSW after reprogramming
Impact on behavior	Changed the logic to initialize ApplUpdated flag while initializing the PROG_CONDITIONS area during the Dcm_Init() phase after reprogramming
Impact on settings	None
Required ASW actions	None

■ If Positive Response is processed after Response Pending, Dcm_ConfirmationStatusType of ServiceRequestSupplierNotification_Confirmation() will be DCM_RES_POS_OK.

Rationale	Even if Positive Response is successfully processed after Response Pending, Dcm_ConfirmationStatusType of ServiceRequestSupplierNotification_Confirmation() is still DCM_RES_NEG_OK.
Impact on behavior	If Positive Response is processed after the first Response Pending, Dcm_ConfirmationStatusType of ServiceRequestSupplierNotification_Confirmation() will have the following difference. - Previous Dcm: DCM_RES_NEG_OK (based on the first negative response) - Current Dcm: DCM_RES_POS_OK (based on the last positive response)
Impact on settings	None
Required ASW actions	If Dcm_ConfirmationStatusType of ServiceRequestSupplierNotification_Confirmation() was used to create logic in earlier version of Dcm, it is needed to review the logic referring to the impact on behavior.

6.3.73 Version 1.3.0

- New features
  - DcmPageBufferCfg

Rationale	The PagedBuffer feature enables Response even when Tx Response Length is larger than Tx Buffer Size, preventing memory waste which is caused by unnecessarily large sizing of Tx Buffer
Impact on behavior	1. PagedBufferEnabled : If it is true - If Tx Response Length > Tx Buffer Size, positive response 2. PagedBufferEnabled : If it is false - If Tx Response Length > Tx Buffer Size, negative response (NRC14, DCM_E_RESPONSETOOLONG)
Impact on settings	1. Add a new container - Dcm/DcmConfigSet/DcmPageBufferCfg 2. Add Paged Buffer to SRS
Required ASW	None

actions	

- Improvements
  - Optimization of Memmap Section

Rationale	- Unnecessary Memmap.h Inclusion at compile time slows down work
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.74 Version 1.2.2

- New features
  - N/A

- Improvements
  - Apply HAC Random Generate if CSAC is in use

Rationale	As a requirement of HMC, the Random Generate logic of the CSAC algorithm has been changed from AutoEver Random Generate to HAC Random Generate with enhanced randomness
Impact on behavior	There is a change in Random Generate logic but it does not affect CSAC behaviors.
Impact on settings	It has dependency with CryptoLib 1.0.4, Cal 1.0.8, FBL_core 1.7.1, and integration_Fbl 1.9.0 or later versions
Required ASW actions	None

  - Change in Dcm_CallOuts.h to support Cypress

Rationale	Compilation error occurred in CYPRESS controller because Dcm_McuDepProgConditionsType structure was defined only by HWRESOURCE_INFINEON, HWRESOURCE_FREESCALE and HWRESOURCE_RENESAS

Impact on behavior	Make change so that the Dcm_McuDepProgConditionsType structure is also defined in the CYPRESS controller
Impact on settings	None
Required ASW actions	None

6.3.75 Version 1.2.1

- New features
  - N/A

- Improvements
  - Change in DcmDslBufferSize in the User Manual

Rationale	Change to the User Manual in line with the SRS update
Impact on behavior	None
Impact on settings	Deleted information related to DcmDslBufferSize in the SRS - If CSAC is in use: Set up Tx Buffer Size 255 and Rx Buffer Size 620 - If CSAC is not in use: Set up Tx and Rx Buffer Size 255
Required ASW actions	None

  - Change in DcmDslBufferSize in the User Manual

Rationale	As Std_Types.h got simplified, the E_REQUEST_NOT_ACCEPTED macro which is used only in the Dcm module became part of the Dcm module
Impact on behavior	Regardless of Software Component Generation, E_REQUEST_NOT_ACCEPTED will be created without exception
Impact on settings	None
Required ASW actions	None

  - In RoutineControl Service Application Callback, OpStatus did not change from DCM_INITIAL to

DCM_PENDING and this error was fixed.

Rationale	DCM_PENDING processing logic error
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.76 Version 1.2.0

- New features
  - N/A

- Improvements
  - Change to negative response to EnableNormalMessageTransmission Service

Rationale	The change scheduled for ES95486-00 V1.9.0 or later was applied in advance
Impact on behavior	Before Dcm 1.2.0: - NRC22 (DCM_E_CONDITIONSNOTCORRECT) was supported After Dcm 1.2.0: - NRC12 (DCM_E_SUBFUNCTIONNOTSUPPORTED) is supported - NRC13 (DCM_E_INCORRECTMESSAGELENGTHORINVALIDFORMAT) is supported - NRC22 (DCM_E_CONDITIONSNOTCORRECT) is supported
Impact on settings	None
Required ASW actions	None

  - Deleted DcmSeedInvalidationForNewRequest setting

Rationale	No use case. No ground for the setting in the specifications
Impact on behavior	Work the same as the existing DcmSeedInvalidationForNewRequest setting false
Impact on	Delete

settings	Dcm/DcmConfigSet/DcmGeneral/DcmSeedInvalidationForNewRequest
Required ASW actions	None

■ In RoutineControl Service Application Callback, sequence error occurred when Pending Response was processed and this error was fixed.

Rationale	If App Callback Function that processes StartRoutine and StopRoutine of RoutineControl Service processes Response Pending, the next Request Sequence Flag is not properly applied.
Impact on behavior	None
Impact on settings	None
Required ASW actions	None

6.3.77 Version 1.1.2

■ New features

■ N/A

■ Improvements

■ Detailed user manual related to diagnosis and security

- Seed-Key (L1), Adv. Separate and update Seed-Key (L9) Sample Code
- Notify that UserCode is required when using SecurityAccess service for Deviation
- Change to Interface Description: Xxx_CompareKey

Rationale	User Guide needs to be provided for application of HSAC
Impact on behavior	None
Impact on settings	None
Required ASW actions	Appendix 10.2 and 10.3 was referenced when the algorithm of Seed-Key(L1) and Advanced Seed-Key(L9) was applied for coding

6.3.78 Version 1.1.1

- New features
  - N/A

- Improvements
  - Upon PDF update, Configuration was changed, too (from Category F to Category C under DcmDslMainConnection)

Rationale	After CAN Import/Harmonize, the Fixed Parameter is released for the part that requires manual setting for USD_ON_CAN.
Impact on behavior	None
Impact on settings	Users can change DcmDslMainConnection container and lower settings
Required ASW actions	None

  - 10.2 Advanced SeedKey Algorithm (HSAC) Sample Code Update

Rationale	User Guide needs to be provided for application of HSAC
Impact on behavior	None
Impact on settings	None
Required ASW actions	Appendix was referenced when the algorithm of Advanced Seed-Key(L9) was applied for coding

6.3.79 Version 1.1.0

- New features
  - AUTRON_AUTOSAR_Dcm_ECU_Configuration_PDF.arxml Update

Rationale	When using the CSAC algorithm, public key management is required for certificate verification

Impact on behavior	1. Whether Fbl is used or not (DcmAutronFblUsed) - true : Use AutoEver FBL - false : Use the FBL of its own 2. Whether Fbl-provided Public Key is used or not (DcmAutronFblSecureLibUsed) - true : Use the public key included in AutoEver FBL - false : Use the public key of its own
Impact on settings	Add DcmAutronFblUsed and DcmAutronFblSecureLibUsed settings
Required ASW actions	Specify use of the CSAC algorithm (L21) in SRS

- Improvements
  - When calling Runnable, initialize the Negative Response Code to be 0x00 (DCM_E_POSITIVERESPONSE)

Rationale	When designing user Runnable, it is advantageous for designing as the negative response code is consistently received from the platform
Impact on behavior	ErrorCode becomes initialized to 0x00 if Runnable is called through RTE Port Interface
Impact on settings	None
Required ASW actions	Design ASW considering this

6.4 Module Release Notes

6.4.1 Limitations

- Pre-Compile is supported
- OBD protocol is not supported
- ResponseOnEvent Service is not supported
- LinkControl Service is not supported
- RequestUpload is not supported
- Regarding DID signal type, only uint8 is supported and bit type is not supported
- Regarding DID and RID Signal constraints, follow AUTOSAR spec 4.2.1.

  As definition of Data position, length, and type is insufficient, the limitations in AUTOSAR 4.2.1 were

applied in advance.

- Only Client-Server interface is supported for DID and RID.
- Service related to Dynamically Defined Data Identifier and Periodic Data Identifier is not supported
- ReadScalingDataByIdentifier (SID24) service is not supported
- DLT not supported
- The following sub-functions in READDTCINFORMATION SERVICE are not supported
  i. reportDTCSnapshotRecordByRecordNumber(0x05)
  ii. reportMirrorMemoryDTCByStatusMask (0x0F)
  iii. reportMirrorMemoryDTCExtendedDataRecordByDTCNumber(0x10)
  iv. reportNumberOfMirrorMemoryDTCByStatusMask (0x11)
  v. reportNumberOfEmissionsRelatedOBDDTCByStatusMask (0x12)
  vi. reportEmissionsRelatedOBDDTCByStatusMask (0x13)
  vii. reportDTCWithPermanentStatus (0X15)
  - Authentication (0x29) Service
    i. Regarding DcmDspAuthenticationUsePort, only USE_SYNCH_FNC is supported

6.4.2 Deviations

- The following services are added or changed under HMC ES95486-00E V1.8.0 Specification
  (1) Services added
  - EnableNormalMsgTransmission
  - StopDiagnosticSession
  (2) Services changed
  - Dcm_DcmDiagnosticSessionControl
  - Dcm_DcmStopDiagnosticSession
  - Dcm_DcmSecurityAccess
  - Dcm_DcmCommunicationControl
  - Dcm_DcmEnableNormalMsgTransmission
- Depending on the type of security algorithm used when using the SecurityAccess(0x27) service (Seed-Key and Advanced Seed-Key), a user code using the Sample Code described in Appendix 10.2 and 10.3 must be added.
- According to HMC ES95486-00E V1.8.0 Specification, EnableNormalMessageTransmission service is modified to control Normal Message and NM Message at the same time.
- Modify DcmDspMaxDidToRead Parameter to have a value of 1-65535 according to AUTOSAR_SWS_DiagnosticCommunicationManager_4.2.2 Specification

- Change the negative response to EnableNormalMessageTransmission Service so as to meet ES95486-00E V1.9.0 and later
- Provide the BswM_Dcm_ApplicationUpdated() feature according to AUTOSAR_SWS_DiagnosticCommunicationManager_4.2.2 Specification
- Apply the Max value of AUTOSAR_SWS_DiagnosticCommunicationManager 4.3.0 Specification version

  Min ~ Max value of DcmTimStrP2ServerAdjust 의 Min ~ Max value

  - AUTOSAR 4.0.3 : 0~1000
  - AUTOSAR 4.3.0 : 0~1

  Min - Max values of DcmTimStrP2StarServerAdjust

  - AUTOSAR 4.0.3 : 0~5000
  - AUTOSAR 4.3.0 : 0~5

  Min - Max values of DcmDspSessionP2ServerMax

  - AUTOSAR 4.0.3 : 0~1000
  - AUTOSAR 4.3.0 : 0~1

  Min - Max values of DcmDspSessionP2StarServerMax

  - AUTOSAR 4.0.3 : 0~100000
  - AUTOSAR 4.3.0 : 0~100
- Apply Verification call sequence according to AUTOSAR_SWS_DiagnosticCommunicationManager_4.3.0 Specification
  i. Verification of Manufacturer permission (Call of the manufacturer interface indication operation)
  ii. Verification of the SID
  iii. Verification of the Diagnostic Session
  iv. Verification of the Service Security Access levels
  v. Verification of the Supplier permission (Call of the Supplier interface indication operation)
  vi. Verification of the Mode rules for service IDs
- According to AUTOSAR_SWS_DiagnosticCommunicationManager_4.4.0 Specification, Authentication Service is implemented. When using authentication service based on AUTOSAR, Crypto Stack should be used R4.4.0
- According to HMC ES95489-01(revision 7) 5.3.4.6, Authentication (0x29) Service's Sub-functions are not supported and if user need to use, User will implement,.
  i. verifyCerificateBidirectional (0x02)
  ii. transmitCertificate (0x04)

iii. requestChallengeForAuthentication (0x05)

iv. verifyProofOfOwnershipUnidirectional (0x06)

v. verifyProofOfOwnershipBidirectional (0x07)

- SWP provide callout function for Authentication (0x29) Service's below NRCs. NRC logics need to implement in Application side.
  i. Challenge calculation failed (0x59)
  ii. Setting Access Rights failed (0x5A)
  iii. DeAuthentication failed (0x5D)
  iv. CRLintegrityFailed (0xF0)
  v. CRLvalidityPeriodFailed (0xF1)
  vi. RoleandRightofCertificateDenied (0xF2)
- The AUTOSAR_SWS_DiagnosticCommunicationManager_4.0.3 specification indicates that the DCM_OEM(SYS)_BOOT_RESPAPP is not supported when jumping to the bootloader. Therefore, the option must be selected using SendRespPendOnTransToBoot.
  i. When SendRespPendOnTransToBoot is set to True: 0x78 is sent, followed by jumping to the boot on FBL, resulting in a positive response.
  ii. When SendRespPendOnTransToBoot is set to False : A positive response is sent, followed by jumping to the boot on FBL.

7 Configuration Guide

(1) Unless otherwise indicated, parameters are based on AUTOSAR specifications. If the parameters are added by HYUNDAI AUTOEVER

it is marked as (Hyundai AutoEver specific).

(2) If there is a default value while not supported (N), the set values must not be changed.

7.1 General

7.1.1 DcmGeneral

Parameter Name	Value	Category
DcmDevErrorDetect	User Defined	C
DcmRespondAllRequest	TRUE	F
DcmRequestManufacturerNotificationEnabled	User Defined	C
DcmRequestSupplierNotificationEnabled	User Defined	C
Standard Support(1)	User Defined	C
DcmTaskTime	0.01	F
DcmVersionInfoApi	User Defined	C
DcmFblUsedType(2)	User Defined	C
DcmRemainUnlockCondition(3)	User Defined	C
DcmForcedEcuReset		N
DcmVinRef	User Defined	C
DcmObdProtocolId(5)	User Defined	C
DemIntegrated	User Defined	C
NvmIntegrated(6)	User Defined	C

(1) Standard Support

UDS behaviors are based on specifications configured in Standard Support.

- DCM_ISO14229_SUPPORT: Support of the ISO14229 specifications
- DCM_ES95486_SUPPORT: ES95486-00 specification is supported

- DCM_ES96590_SUPPORT: ES96590 specification is supported
- DCM_ES95486_02_SUPPORT: Support of the ES95486-02 specifications
- DCM_ES95486_50_SUPPORT: Support of ES95486-50 specifications
- DCM_QZN04_SUPPORT: Support for the QZN04 specifications

(2) DcmFblUsedType

When deciding on the type of FBL to use,

- DCM_AUTOEVER_FBL_UNUSED : Disable Autoever FBL
- DCM_AUTOEVER_FBL_USED_RXCONNECTION_ID_UNUSED :

Autoever FBL is used and Rx Connection ID is not used in FBL. (Need to check FBL version)

- DCM_AUTOEVER_FBL_USED_RXCONNECTION_ID_USED :

Autoever FBL is used and Rx Connection ID is used in FBL. (Need to check FBL version)

(3) DcmRemainUnlockCondition

If Remain Unlock Condition is set,

the security level remains during transition of Extended Session -> Security Access -> Extended Session

(5) DcmObdProtocolId

OBD operation protocol configuration

- DCM_PROTOCOLID_OBD_NONE : OBD Not Used
- DCM_PROTOCOLID_J1979_2_OBD_ON_UDS : J1979-2 OBD on Uds protocol used
- DCM_PROTOCOLID_J1979_OBD2 : J1979 OBD2 protocol used

(6) NvmIntegrated

For the project which does not include Nvm module like as light pltaform, NvmIntegrated is set to FALSE.

7.1.2 DcmPageBufferCfg

Parameter Name	Value	Category
DcmPagedBufferEnabled	User Defined	C
DcmPagedBufferTimeout	User Defined	C

Note If you use a paged buffer, the value of parameter 'DcmPagedBufferTimeout' should set to more than twice the configured value for parameter 'DcmTaskTime'.

7.1.3 DcmDsd

7.1.3.1 DcmDsdService

Parameter Name	Value	Category
DcmDsdSidTabId	User Defined	C

7.1.3.1.1 DcmDsdServiceTable

Parameter Name	Value	Category
DcmDsdSidTabSecurityLevelRef	User Defined	C
DcmDsdSidTabServiceId(1)	User Defined or From SRS(1)	F or C
DcmDsdSidTabSessionLevelRef	User Defined	C
DcmDsdSidTabSubfuncAvail	User Defined or From SRS	F or C
DcmDsdSidTabFnc(2)	User Defined(2)	C
DcmDsdSidTabModeRuleRef		N
DcmDsdServiceRole(3)		C

(1) DcmDsdSidTabServiceId

****For some of diagnostic services, arbitrary change is not allowed as their operation is related to other modules.**

(2) DcmDsdSidTabFnc

Insert function symbol if user-defined service is in use.

(3) DcmDsdServiceRole

If user use Authenticaion Service, this parameter must be configured.

Default value: 0

DcmDsdSubService

Parameter Name	Value	Category
DcmDsdSubServiceId(1)	User Defined or From SRS(1)	F or C
DcmDsdSubServiceSecurityLevelRef	User Defined	C

Parameter Name	Value	Category
DcmDsdSubServiceSessionLevelRef	User Defined	C
DcmDsdSubServiceFnc	User Defined(2)	C
DcmDsdSubServiceModeRuleRef		N
DcmDsdSubServiceRole(3)	User Defined(3)	C

(1) DcmDsdSubServiceId

****For some of diagnostic services, arbitrary change is not allowed as their operation is related to other modules.**

(2) DcmDsdSubServiceFnc

Insert function symbol if user-defined service is in use.

(3) DcmDsdSubServiceRole

If user use Authenticaion Service, this parameter must be configured.

Default value: 0

7.1.4 DcmDsl

7.1.4.1 DcmDslBuffer

Parameter Name	Value	Category
DcmDslBufferSize(1)	User Defined	C

(1) DcmDslBufferSize:

Size of the diagnostic buffer in bytes.

For a linear buffer the size shall be as large as the longest diagnostic message (request or response).

For a paged buffer the size has impacts on the application performance..

Note This value is set to 255 by default except for using CSAC algorithm (Security Level L21).

If CSAC is in use, it is needed to set as rx buffer size 620.

If SecurityAccess 2.0 is in use, it is needed to add rx buffer size 501byte.

If OTA is in use, it is needed to set rx buffer size 1026.

If RXSWIN is in use, Application need to set the value according to their needs.

7.1.4.2 DcmDslCallbackDCMRequestService

Container Name	Value	Category
DcmDslCallbackDCMRequestService(1)	User Defined	C

(1) DcmDslCallbackDCMRequestService:

The name of this container is used to define the name of the R-Port through which the DCM access the interface CallbackDCMRequestServices.

The R-Port is named CallbackDCMRequestServices_<SWC> where _<SWC> is the name of the container DcmDslCallbackDCMRequestService

7.1.4.3 DcmDslDiagResp

Parameter Name	Value	Category
DcmDslDiagRespOnSecondDeclinedRequest		N
DcmDslDiagRespMaxNumRespPend	User Defined	C

7.1.4.4 DcmDslProtocolRow

Parameter Name	Value	Category
DcmDslProtocolID	User Defined	C
DcmDslProtocolEndiannessConvEnabled		N
DcmDslProtocolIsParallelExecutab		N
DcmDslProtocolPreemptTimeout	User Defined	C
DcmDslProtocolPriority	User Defined	C
DcmTimStrP2ServerAdjust	User Defined	C
DcmTimStrP2StarServerAdjust	User Defined	C
DcmDslProtocolRxBufferID	User Defined	C
DcmDslProtocolSIDTable	User Defined	C
DcmDslProtocolTxBufferID	User Defined	C
DcmDslProtocolSessionRef	User Defined	C
DcmDslProtocolTransType	TYPE2	F

Parameter Name	Value	Category
DcmSendRespPendOnTransToBoot	User Defined	C

7.1.4.4.1 DcmDslConnection

Sub Container Name	Value	Category
DcmDslMainConnection	User Defined	C
DcmDslPeriodicTransmission		N
DcmDslResponseOnEvent		N

DcmDslMainConnection

Parameter Name	Value	Category
DcmDslProtocolRxTesterSourceAddr	User Defined	C
DcmDslPeriodicTransmissionConRef		N
DcmDslROEConnectionRef		N
DcmDslProtocolAuthenticaionConnectionId *	User Defined	C

Note: DcmDslProtocolAuthenticaionConnectionId is only available when Authentication Service is used

7.1.4.4.1.1 DcmDslProtocolRx

Parameter Name	Value	Category
DcmDslProtocolRxAddrType	User Defined (DCM_FUNCTIONAL_TYPE or DCM_PHYSICAL_TYPE)	C
DcmDslProtocolRxChannelId *	User Defined (ComM Channel Id of DcmDslProtocolRxComMChannelRef)	C
DcmDslProtocolRxPduId	User Defined (0 or 1)	C
DcmDslProtocolRxPduRef	User Defined (DcmRxPduId reference for reception of requests)	C

Parameter Name	Value	Category
DcmDslProtocolRxComMChannelRef *	User Defined (Reference to the ComMChannel on which the DcmDslProtocolRxPdu is received)	C

Note : ComMChannelId of DcmDslProtocolRxChannelId, DcmDslProtocolRxComMChannelRef should be same.

7.1.4.4.1.1.2 DcmDslProtocolTx

Parameter Name	Value	Category
DcmDslProtocolTxPduRef	User Defined (DcmTxPduId reference for transmission of responses)	C
DcmDslTxConfirmationPduId	User Defined (Pud id of DcmDslProtocolRxPduId)	C

DcmDslPeriodicTransmission

Sub Container Name	Value	Category
DcmDslPeriodicConnection		N

7.1.4.4.2 DcmDslPeriodicConnection

Parameter Name	Value	Category
DcmDslPeriodicTxConfirmationPduId		N
DcmDslPeriodicTxPduRef		N

7.1.4.4.3 DcmDslResponseOnEvent

Parameter Name	Value	Category
DcmDslROETxPduRef		N
DcmDslRoeTxConfirmationPduId		N

7.1.4.5 DcmDslServiceRequestManufacturerNotification

Parameter Name	Value	Category
ShortName	User Defined	C

7.1.4.6 DcmDslServiceRequestSupplierNotification

Parameter Name	Value	Category
ShortName	User Defined	C

7.1.5 DcmDsp

Parameter Name	Value	Category
DcmDspPowerDownTime(1)	User Defined	C
DcmDspMaxDidToRead(2)	User Defined	C
DcmDspMaxPeriodicDidScheduler		N
DcmDspMaxPeriodicDidToRead		N
DcmDspDDDIDcheckPerSourceDID		N

(1) DcmDspPowerDownTime:

This parameter indicates to the client the minimum time of the stand-by sequence the server will remain in the power-down sequence. The resolution of this parameter is one second per count.

The following values are valid:

00 - FE hex: 0 - 254 s powerDownTime;

FF hex: indicates a failure or time not available.

In case the parameter DcmDspPowerDownTime is present, the DCM shall set the powerDownTime in positive response to sub-service enableRapidPowerShutDown of ECUReset service with value set in DcmDspPowerDownTime

(2) DcmDspMaxDidToRead:

Indicates the maximum allowed DIDs in a single "ReadDataByIdentifier" request.

The buffer size should be considered.

The following values are valid : 1 - 65535

7.1.5.1 DcmDspComControl

Sub Container Name	Value	Category
DcmDspComControlAllChannel	-	C
DcmDspComControlSpecificChannel	-	C
DcmDspComControlSetting		N

7.1.5.1.1 DcmDspComControlAllChannel

Parameter Name	Value	Category
DcmDspAllComMChannelRef	Reference to ComM channel.	C

7.1.5.1.2 DcmDspComControlSpecificChannel

Parameter Name	Value	Category
DcmDspSubnetNumber(1)	1	C
DcmDspSpecificComMChannelRef	Reference to ComM channel.	C

(1) DcmDspSubnetNumber : ES95486-00 Only

7.1.5.1.3 DcmDspComControlSetting

Parameter Name	Value	Category
DcmDspComControlCommunicationReEnableModeRuleRef		N

7.1.5.2 DcmDspData

Parameter Name	Value	Category
DcmDspDataConditionCheckReadFnc(1)	User Defined	C
DcmDspDataEcuSignal		N
DcmDspDataFreezeCurrentStateFnc(2)	User Defined	C
DcmDspDataGetScalingInfoFnc		N

Parameter Name	Value	Category
DcmDspDataReadDataLengthFnc(3)	User Defined	C
DcmDspDataReadFnc(4)	User Defined	C
DcmDspDataResetToDefaultFnc(5)	User Defined	C
DcmDspDataReturnControlToEcuFnc(6)	User Defined	C
DcmDspDataShortTermAdjustmentFnc(7)	User Defined	C
DcmDspDataWriteFnc(8)	User Defined	C
DcmDspDataReadEcuSignal(9)	User Defined	C
DcmDspDataSize(10)	User Defined	C
DcmDspDataType(11)	User Defined	C
DcmDspDataUsePort(12)	User Defined	C
DcmDspDataInfoRef(13)	User Defined	C
DcmDspDataBlockIdRef(14)	User Defined	C

(1)DcmDspDataConditionCheckReadFnc:

Function name to demand application if the conditions (e.g. System state) to read the DID are correct. (ConditionCheckRead-function).

Only relevant if DcmDspDataUsePort=="*USE_DATA_SYNCH_FNC* or DcmDspDataUsePort==*USE_DATA_ASYNCH_FNC*".

This parameter is related to the interface Xxx_ConditionCheckRead.

(2)DcmDspDataFreezeCurrentStateFnc:

Function name to request to application to freeze the current state of an IOControl. (FreezeCurrentState-function).

Only relevant if DcmDspDataUsePort=="*USE_DATA_SYNCH_FNC* or DcmDspDataUsePort==*USE_DATA_ASYNCH_FNC*" and SIGNAL and UDS Service *InputOutputControlByIdentifier* is configured.

This parameter is related to the interface Xxx_FreezeCurrentState.

(3)DcmDspDataReadDataLengthFnc:

Function name to request from application the data length of a DID. (ReadDataLength-function).

Only relevant if DcmDspDataUsePort=="*USE_DATA_SYNCH_FNC* or DcmDspDataUsePort==*USE_DATA_ASYNCH_FNC*" and *DcmDspDataFixedLength == TRUE*.

This parameter is related to the interface Xxx_ReadDataLength.

(4)DcmDspDataReadFnc:

Function name to request from application the data value of a DID. (ReadData-function).

Only relevant if DcmDspDataUsePort=="*USE_DATA_SYNCH_FNC* or DcmDspDataUsePort==*USE_DATA_ASYNCH_FNC*".

This parameter is related to the interface Xxx_ReadData.

(5)DcmDspDataResetToDefaultFnc:

Function name to request to application to reset an IOControl to default value. (ResetToDefault-function).

Only relevant if DcmDspDataUsePort=="*USE_DATA_SYNCH_FNC* or DcmDspDataUsePort==*USE_DATA_ASYNCH_FNC*" and SIGNAL and UDS Service *InputOutputControlByIdentifier* is configured.

This parameter is related to the interface Xxx_ResetToDefault.

(6)DcmDspDataReturnControlToEcuFnc:

Only relevant if DcmDspDataUsePort=="*USE_DATA_SYNCH_FNC* or DcmDspDataUsePort==*USE_DATA_ASYNCH_FNC*" and SIGNAL and UDS Service *InputOutputControlByIdentifier* is configured.

(7)DcmDspDataShortTermAdjustmentFnc:

Function name to request to application to return control to ECU of an IOControl. (ReturnControlToECU-

function).

Only relevant if DcmDspDataUsePort=="*USE_DATA_SYNCH_FNC* or DcmDspDataUsePort==*USE_DATA_ASYNCH_FNC*" and SIGNAL and UDS Service *InputOutputControlByIdentifier* is configured.

This parameter is related to the interface Xxx_ReturnControlToECU.

(8)DcmDspDataWriteFnc:

Function name to request application to write the data value of a DID. (WriteData-function).

Only relevant if DcmDspDataUsePort=="*USE_DATA_SYNCH_FNC* or DcmDspDataUsePort==*USE_DATA_ASYNCH_FNC*".

This parameter is related to the interface Xxx_WriteData.

(9)DcmDspDataReadEcuSignal:

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 and UDS Service *InputOutputControlByIdentifier* is configured.

(10)DcmDspDataSize:

Length of data in bits associated to the Data. If Data has variable datalength, that corresponds to the maximum datalength.

Note: bit type not supported. In case of byte, bit type means data range between 1 and 7.

(11)DcmDspDataType:

Provide the data type of Data belonging to a DID.

SINT16

SINT32

SINT8

UINT16

UINT32

UINT8

Note: UINT8 support only

(12)DcmDspDataUsePort:

Define which interface shall be used to access the data.

USE_BLOCK_ID

USE_DATA_ASYNCH_CLIENT_SERVER

USE_DATA_ASYNCH_FNC

USE_DATA_SENDER_RECEIVER

USE_DATA_SYNCH_CLIENT_SERVER

USE_DATA_SYNCH_FNC

USE_ECU_SIGNAL

Note: USE_DATA_SENDER_RECEIVER and USE_ECU_SIGNAL are not supported.

(13)DcmDspDataInfoRef:

Reference to DcmDspDataInfo

(14)DcmDspDataBlockIdRef:

NRAM blockId to access the data. Reference to [NvMBlockDescriptor]

Only relevant if DcmDspDataUsePort==USE_BLOCK_ID.

〈Data array type overview〉

	STATIC, VARIABLE
	[8-8*N]
DcmDspDataSize	(size MOD 8) == 0
DcmDspDidDataPos	(size MOD 8) == 0
DcmDspDataType	UINT8
Port	C/S FNC(C/S)
resulting ImpType	DataArrayUint8_{Data}

*C/S,FNC: Client Server Interface

7.1.5.3 DcmDspDataInfo

Parameter Name	Value	Category
DcmDspDataFixedLength(1)	User Defined	C
DcmDspDataScalingInfoSize(2)	User Defined	C

(1) DcmDspDataFixedLength:

Indicates if the datalength of the Data is fixed true = datalength of the Data is fixed false = datalength of the Data is variable

(2) DcmDspDataScalingInfoSize:

If Scaling information service is available for this Data, it provides the size of the scaling information.

7.1.5.4 DcmDspDid

Parameter Name	Value	Category
DcmDspDidIdentifier(1)	User Defined	C
DcmDspDidUsed(2)	User Defined	C
DcmDspDidInfoRef(3)	User Defined	C
DcmDspDidRef(4)	User Defined	C
DcmDspDidPreConfigured(AUTOEVER specific)		N
DcmDspDidRoeQueueEnabled		N

(1) DcmDspDidIdentifier:

2 byte Identifier of the DID. All DcmDspDidIdentifier values shall be unique.

(2) DcmDspDidUsed:

Allow to activate or deactivate the usage of a DID, for multi purpose ECUs

true = DID available

false = DID not available

(3) DcmDspDidInfoRef

Reference to DcmDspDidInfo containing information on this DID.

(4) DcmDspDidRef

Reference to DcmDspDid in case this DID refer to one or several other DID's

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

7.1.5.4.1 DcmDspDidSignal

Parameter Name	Value	Category
DcmDspDidDataPos(1)	User Defined	C
DcmDspDidDataRef(2)	User Defined	C
DcmDspDidSignalEndianness		N

(1) DcmDspDidDataPos:

Defines the position of the data defined by DcmDspDidDataRef reference to DcmDspData container in the DID. The position is defined in bits.

(2) DcmDspDidDataRef:

Reference to 1 DcmDspData container relevant for this DID.

7.1.5.5 DcmDspPeriodicTransmission

Parameter Name	Value	Category
DcmDspPeriodicTransmissionSlowRate		N
DcmDspPeriodicTransmissionMediumRate		N
DcmDspPeriodicTransmissionFastRate		N

7.1.5.6 DcmDspDidRange

Parameter Name	Value	Category
DcmDspDidRangeHasGaps(1)	User Defined	C

Parameter Name	Value	Category
DcmDspDidRangeIdentifierLowerLimit(2)	User Defined	C
DcmDspDidRangeIdentifierUpperLimit(3)	User Defined	C
DcmDspDidRangeIsDidAvailableFnc(4)	User Defined	C
DcmDspDidRangeMaxDataLength(5)	User Defined	C
DcmDspDidRangeReadDidFnc(6)	User Defined	C
DcmDspDidRangeUsePort(7)	User Defined	C
DcmDspDidRangeWriteDidFnc(8)	User Defined	C
DcmDspDidRangeInfoRef(9)	User Defined	C

(1) DcmDspDidRangeHasGaps:

Parameter specifying if there are gaps in the DID range (parameter set to TRUE) or not (parameter set to FALSE)

(2) DcmDspDidRangeIdentifierLowerLimit

Lower limit of DID range

(3) DcmDspDidRangeIdentifierUpperLimit:

Upper limit of DID range.

(4) DcmDspDidRangeIsDidAvailableFnc:

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.

(5) DcmDspDidRangeMaxDataLength:

Maximum data length in bytes.

(6) DcmDspDidRangeReadDidFnc:

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.

(7) DcmDspDidRangeUsePort:

When the parameter DcmDspDidRangeUsePort is set to true the DCM will access the Data using an R-Port requiring a PortInteface DataServices_DIDRange. In that case, DcmDspDidRangeIsDidAvailableFnc, DcmDspDidRangeReadDidFnc and DcmDspDidRangeWriteDidFnc are ignored and the RTE APIs are used.

Note: When the parameter DcmDspDidRangeUsePort is false, the DCM calls the functions defined in DcmDspDidRangeIsDidAvailableFnc, DcmDspDidRangeReadDidFnc and DcmDspDidRangeWriteDidFnc.

(8) DcmDspDidRangeWriteDidFnc:

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.

(9) DcmDspDidRangeInfoRef:

Reference to DcmDspDidInfo containing information on this DID Range.

7.1.5.7 DcmDspExtRoe

Not supported

7.1.5.8 DcmDspDidInfo

Parameter Name	Value	Category
DcmDspDidDynamicallyDefined(1)	User Defined	C

(1) DcmDspDidDynamicallyDefined:

Indicates if this DID can be dynamically defined true = DID can be dynamically defined false = DID can not be dynamically defined

7.1.5.8.1 DcmDspDidAccess

Sub Container(s)	Value	Category
DcmDspDidControl(1)	User Defined	C
DcmDspDidRead(2)	User Defined	C

Sub Container(s)	Value	Category
DcmDspDidWrite(3)	User Defined	C

(1) DcmDspDidControl:

This container contains the configuration (parameters) of the DID control.

(2) DcmDspDidRead:

This container contains the configuration (parameters) of the DID read

(3) DcmDspDidWrite:

This container contains the configuration (parameters) of the DID write.

7.1.5.8.2 DcmDspDidControl

This container contains the configuration of the InputOutputControlByIdentifier service.

Parameter Name	Value	Category
DcmDspDidFreezeCurrentState(1)	User Defined	C
DcmDspDidResetToDefault(2)	User Defined	C
DcmDspDidReturnControlToEcu(3)	User Defined	C
DcmDspDidShortTermAdjustment(4)	User Defined	C
DcmDspDidControlSecurityLevelRef(5)	User Defined	C
DcmDspDidControlSessionRef(6)	User Defined	C
DcmDspDidControlModeRuleRef		N

(1) DcmDspDidFreezeCurrentState:

This indicates the presence of "FreezeCurrentState".

(2) DcmDspDidResetToDefault:

This indicates the presence of "ResetToDefault".

(3) DcmDspDidReturnControlToEcu:

This indicates the presence of "ReturnControlToEcu"

(4) DcmDspDidShortTermAdjustment:

This indicates the presence of "ShortTermAdjustment".

(5) DcmDspDidControlSecurityLevelRef:

Reference to DcmDspSecurityRow

Security levels allowed to control this DID. If there is no reference, no check of security level shall be done.

(6) DcmDspDidControlSessionRef:

Reference to DcmDspSessionRow

Sessions allowed to control this DID. If there is no reference, no check of session level shall be done.

7.1.5.8.3 DcmDspDidRead

Parameter Name	Value	Category
DcmDspDidReadSecurityLevelRef[(1)]	User Defined	C
DcmDspDidReadSessionRef[(2)]	User Defined	C
DcmDspDDDIDMaxElements[(3)]		N
DcmDspDidReadModeRuleRef		N

(1) DcmDspDidReadSecurityLevelRef:

Reference to DcmDspSecurityRow

Security levels allowed to read this DID. If there is no reference, no check of security level shall be done.

(2) DcmDspDidReadSessionRef:

Reference to DcmDspSessionRow

Sessions allowed to read this DID. If there is no reference, no check of session level shall be done.

(3) DcmDspDDDIDMaxElements:

Maximum number of source elements of a DDDID(Dynamically Defined Data IDentifier)

7.1.5.8.4 DcmDspDidWrite

Parameter Name	Value	Category
DcmDspDidWriteSecurityLevelRef[(1)]	User Defined	C

Parameter Name	Value	Category
DcmDspDidWriteSessionRef(2)	User Defined	C
DcmDspDidWriteModeRuleRef		N

(1) DcmDspDidWriteSecurityLevelRef:

Reference to DcmDspSecurityRow

Security levels allowed to write this DID. If there is no reference, no check of security level shall be done.

(2) DcmDspDidWriteSessionRef:

Reference to DcmDspSessionRow

Sessions allowed to write this DID. If there is no reference, no check of session level shall be done.

7.1.5.9 DcmDspMemory

Parameter Name	Value	Category
DcmDspUseMemoryId		N

Note: The memory id is not supported(HMC ES95486-00)

Sub Container(s)	Value	Category
DcmDspAddressAndLengthFormatIdentifier(1)	User Defined	C
DcmDspMemoryIdInfo(2)	User Defined	C

(1) DcmDspAddressAndLengthFormatIdentifier:

This container contains the configuration of the supported AddressAndLengthFormatIdentifiers for memory access.

(2) DcmDspMemoryIdInfo:

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, RequestDownload, RequestUpload. (In the case of RequestDownload and RequestUpload, the user must directly implement the User Callout that is called when a range check for the memory address is required.)

7.1.5.9.1 DcmDspMemoryIdInfo

Parameter Name	Value	Category
DcmDspMemoryIdValue		N

Note: The memory id is not supported(HMC ES95486-00)

Sub Container(s)	Value	Category
DcmDspReadMemoryRangeInfo(1)	User Defined	C
DcmDspWriteMemoryRangeInfo(2)	User Defined	C

(1) DcmDspReadMemoryRangeInfo:

Provides the range of memory address allowed for reading.

(2) DcmDspWriteMemoryRangeInfo:

Provides the range of memory address allowed for writting.

DcmDspReadMemoryRangeInfo

Parameter Name	Value	Category
DcmDspReadMemoryRangeHigh(1)	User Defined	C
DcmDspReadMemoryRangeLow(2)	User Defined	C
DcmDspReadMemoryRangeSecurityLevelRef(3)	User Defined	C
DcmDspReadMemoryRangeModeRuleRef		N

(1) DcmDspReadMemoryRangeHigh:

High memory address of a range allowed for reading

(2) DcmDspReadMemoryRangeLow:

Low memory address of a range allowed for reading

(3) DcmDspReadMemoryRangeSecurityLevelRef:

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.

DcmDspWriteMemoryRangeInfo

Parameter Name	Value	Category
DcmDspWriteMemoryRangeHigh(1)	User Defined	C
DcmDspWriteMemoryRangeLow(2)	User Defined	C
DcmDspWriteMemoryRangeSecurityLevelRef(3)	User Defined	C
DcmDspWriteMemoryRangeModeRuleRef		N

(1) DcmDspWriteMemoryRangeHigh:

High memory address of a range allowed for writing.

(2) DcmDspWriteMemoryRangeLow:

Low memory address of a range allowed for writing

(3) DcmDspWriteMemoryRangeSecurityLevelRef:

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.

7.1.5.9.2 DcmDspAddressAndLengthFormatIdentifier

Parameter Name	Value	Category
DcmDspSupportedAddressAndLengthFormatIdentifier(1)	User Defined	C

(1) DcmDspSupportedAddressAndLengthFormatIdentifier:

This parameter defines the supported AddressAndLengthFormatIdentifier of the request message.

Note: The high nibble of Address and length Format Identifier (Number of bytes for Memory size) shall not exceed 4.

7.1.5.10 DcmDspPid

Parameter Name	Value	Category
DcmDspPidIdentifier	-	N
DcmDspPidSize	-	N
DcmDspPidUsed	-	N
DcmDspPidService	-	N

Sub Container(s)	Value	Category
DcmDspPidData	-	N
DcmDspPidSupportInfo	-	N

7.1.5.10.1 DcmDspPidData

Parameter Name	Value	Category
DcmDspPidDataPos	User Defined	C
DcmDspPidSignalEndianness	User Defined	C
DcmDspPidDataType	User Defined	C
DcmDspPidDataSize	User Defined	C

Sub Container(s)	Value	Category
DcmDspPidDataSupportInfo	User Defined	C
DcmDspPidService01	User Defined	C
DcmDspPidService02	User Defined	C

7.1.5.10.2 DcmDspPidDataSupportInfo

Parameter Name	Value	Category
DcmDspPidDataSupportInfoBit	User Defined	C
DcmDspPidDataSupportInfoRef	User Defined	C

7.1.5.10.3 DcmDspPidService01

Parameter Name	Value	Category
DcmDspPidDataReadFnc	User Defined	C
DcmDspPidDataUsePort	User Defined	C

7.1.5.10.4 DcmDspPidService02

Parameter Name	Value	Category
DcmDspPidDataDemRef	User Defined	C

7.1.5.10.5 DcmDspPidSupportInfo

Parameter Name	Value	Category
DcmDspPidSupportInfoLen	User Defined	C
DcmDspPidSupportInfoPos	User Defined	C

7.1.5.11 DcmDspRequestControl

Parameter Name	Value	Category
DcmDspRequestControlInBufferSize	User Defined	C
DcmDspRequestControlOutBufferSize	User Defined	C
DcmDspRequestControlTestId	User Defined	C
DcmDspRequestControlFnc	User Defined	C

7.1.5.12 DcmDspRoe

Parameter Name	Value	Category
DcmDspRoeBufSize	-	N
DcmDspRoeInitOnDSC	-	N
DcmDspRoeInterMessageTime	-	N
DcmDspRoeMaxNumberOfRetry	-	N
DcmDspRoeMaxEventLength	-	N
DcmDspRoeQueueEnabled	-	N
DcmDspRoeStopFnc	-	N
DcmDspRoeInitFnc	-	N
DcmDspRoeMaxQueueLength	-	N
DcmDspRoeBlockIdRef	-	N

7.1.5.13 DcmDspRoutine

Parameter Name	Value	Category
DcmDspRequestResultsRoutineFnc(1)	User Defined	C
DcmDspRoutineFixedLength(2)	User Defined	C
DcmDspRoutineIdentifier(3)	User Defined	C
DcmDspRoutineUsePort(4)	User Defined	C
DcmDspRequestResultsRoutineSupported(5)	User Defined	C
DcmDspRoutineUsed(6)	User Defined	C
DcmDspStopRoutineSupported(7)	User Defined	C
DcmDspStartRoutineFnc(8)	User Defined	C
DcmDspStopRoutineFnc(9)	User Defined	C
DcmDspRoutineInfoRef(10)	User Defined	C

(1) DcmDspRequestResultsRoutineFnc:

Function name for request to application the results of a routine. (Routine_RequestResults-function)

This parameter is related to the interface Xxx_RequestResults.

(2) DcmDspRoutineFixedLength:

Indicates if the datalength of the optional record in the RoutineControl request and response is fixed.

true = datalength of the optional record is fixed

false = datalength of the optional record is variable.

Note: In case DcmDspRoutineFixedLength is set to FALSE, the DcmDspRoutineSignalLength for the last signal is the maximum length (in bits) of the optional record.

(3) DcmDspRoutineIdentifier:

2 bytes Identifier of the RID.

All DcmDspRoutineIdentifier values shall be unique.

(4) DcmDspRoutineUsePort

If this parameter is set to true, the DCM uses a port requiring a PortInterface

RoutineServices_<ROUTINENAME>. 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.

Note: If this is false, the DCM expects to find the names of the functions to be used in DcmDspStartRoutineFnc, DcmDspStopRoutineFnc or DcmDspRequestResultsRoutineFnc.

(5) DcmDspRequestResultsRoutineSupported:

Indicates if the optional requestRoutineResults in the RoutineControl is supported.

true = requestRoutineResults is supported.

false = requestRoutineResults is not supported.

(6) DcmDspRoutineUsed:

Allow to activate or deactivate the usage of a Routine, for multi purpose.

true = Routine available ECUs.

false = Routine not available.

(7) DcmDspStopRoutineSupported:

Indicates if the optional stopRoutine in the RoutineControl is supported.

true = stopRoutine is supported.

false = stopRoutine is not supported.

(8) DcmDspStartRoutineFnc:

Function name for request to application to start a routine. (Routine_Start-function)

This parameter is related to the interface Xxx_Start.

(9) DcmDspStopRoutineFnc:

Function name for request to application to stop a routine. (Routine_Stop-function)

This parameter is related to the interface Xxx_Stop.

(10) DcmDspRoutineInfoRef:

Reference to DcmDspRoutineInfo containing information on this routine.

7.1.5.14 DcmDspRoutineInfo

Sub Container(s)	Value	Category
DcmDspRoutineAuthorization(1)	User Defined	C
DcmDspRoutineRequestResOut(2)	User Defined	C
DcmDspRoutineStopIn(3)	User Defined	C
DcmDspRoutineStopOut(4)	User Defined	C
DcmDspStartRoutineIn(5)	User Defined	C
DcmDspStartRoutineOut(6)	User Defined	C

(1) DcmDspRoutineAuthorization:

This container contains the configuration (parameters) for the Routine Authorization. (Security, Session)

(2) DcmDspRoutineRequestResOut:

Provide description of output parameter of RequestResult subservice for RoutineControl service.

(3) DcmDspRoutineStopIn:

Provide description of input parameter of Stop subservice for RoutineControl service.

(4) DcmDspRoutineStopOut:

Provide description of output parameter of Stop subservice for RoutineControl service.

(5) DcmDspStartRoutineIn:

Provide description of input parameter of Start subservice for RoutineControl service.

(6) DcmDspStartRoutineOut:

Provide description of output parameter of Start subservice for RoutineControl service.

7.1.5.14.1 DcmDspRoutineAuthorization

Parameter Name	Value	Category
DcmDspRoutineSecurityLevelRef(1)	User Defined	C
DcmDspRoutineSessionRef(2)	User Defined	C
DcmDspRoutineModeRuleRef		N

(1) DcmDspRoutineSecurityLevelRef:

Reference to DcmDspSecurityRow Security levels allowed to control this RID. If there is no reference, no check of security level shall be done.

(2) DcmDspRoutineSessionRef:

Reference to DcmDspSessionRow Sessions allowed to control this RID. If there is no reference, no check of session level shall be done.

7.1.5.14.2 DcmDspRoutineRequestResOut

Provide description of output parameter of RequestResult subservice for RoutineControl service

Sub Container(s)	Value	Category
DcmDspRoutineRequestResOutSignal(1)	User Defined	C

(1) DcmDspRoutineRequestResOutSignal:

Provide description of a routine signal used in RoutineControl service.

DcmDspRoutineRequestResOutSignal

Parameter Name	Value	Category
DcmDspRoutineSignalLength(1)	User Defined	C
DcmDspRoutineSignalPos(2)	User Defined	C
DcmDspRoutineSignalType(3)	User Defined	C
DcmDspRequestRoutineResultsRole	User Defined	C
DcmDspRoutineRequestEndianness		N

(1) DcmDspRoutineSignalLength:

Provide the length in bits of the signal in the RoutineControl request/response

(2) DcmDspRoutineSignalPos:

Provide the position of the signal in the RoutineControl request/response. The position is defined in bits. The value of the parameter should be configured a multiple of 〈8〉

(3) DcmDspRoutineSignalType:

Provide the type of the signal in the RoutineControl request/response.

BOOLEAN	Not supported
SINT16	type of the signal is sint16. 〈Not Supported yet〉
SINT32	type of the signal is sint32. 〈Not Supported yet〉
SINT8	type of the signal is sint8. 〈Not Supported yet〉
UINT16	type of the signal is uint16.
UINT32	type of the signal is uint32.
UINT8	type of the signal is uint8.
VARIABLE_LENGTH	type of the signal is uint8[(DcmDspRoutineSignalLength+7)/8]. This is only valid for the last signal and when DcmDspRoutineFixedLength is set to FALSE.

(4) DcmDspRequestRoutineResultsRole:

Provide the Role in the RoutineControl Results request/response

7.1.5.14.3 DcmDspRoutineStopIn

Provide description of input parameter of Stop subservice for RoutineControl service.

Sub Container(s)	Value	Category
DcmDspRoutineStopInSignal	User Defined	C

DcmDspRoutineStopInSignal

Provide description of a routine signal used in RoutineControl service.

Parameter Name	Value	Category
DcmDspRoutineSignalLength(1)	User Defined	C
DcmDspRoutineSignalPos(2)	User Defined	C
DcmDspRoutineSignalType(3)	User Defined	C
DcmDspRoutineStopInEndianness		N

(1) DcmDspRoutineSignalLength:

Provide the length in bits of the signal in the RoutineControl request/response.

(2) DcmDspRoutineSignalPos:

Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.

The value of the parameter should be configured a multiple of 〈8〉

(3) DcmDspRoutineSignalType:

BOOLEAN	Not supported
SINT16	type of the signal is sint16. 〈Not Supported yet〉
SINT32	type of the signal is sint32. 〈Not Supported yet〉
SINT8	type of the signal is sint8. 〈Not Supported yet〉
UINT16	type of the signal is uint16.
UINT32	type of the signal is uint32.
UINT8	type of the signal is uint8.
VARIABLE_LENGTH	type of the signal is uint8[(DcmDspRoutineSignalLength+7)/8]. This is only valid for the last signal and when DcmDspRoutineFixedLength is set to FALSE.

7.1.5.14.4 DcmDspRoutineStopOut

Provide description of output parameter of Stop subservice for RoutineControl service.

Sub Container(s)	Value	Category
DcmDspRoutineStopOutSignal	User Defined	C

DcmDspRoutineStopOutSignal

Parameter Name	Value	Category
DcmDspRoutineSignalLength(1)	User Defined	C
DcmDspRoutineSignalPos(2)	User Defined	C
DcmDspRoutineSignalType(3)	User Defined	C
DcmDspStopRoutineRole(4)	User Defined	C
DcmDspRoutineStopOutEndianness		N

(1) DcmDspRoutineSignalLength:

Provide the length in bits of the signal in the RoutineControl request/response.

(2) DcmDspRoutineSignalPos:

Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.

The value of the parameter should be configured a multiple of 〈8〉

(3) DcmDspRoutineSignalType:

BOOLEAN	Not supported
SINT16	type of the signal is sint16. 〈Not Supported yet〉
SINT32	type of the signal is sint32. 〈Not Supported yet〉
SINT8	type of the signal is sint8. 〈Not Supported yet〉
UINT16	type of the signal is uint16.
UINT32	type of the signal is uint32.
UINT8	type of the signal is uint8.
VARIABLE_LENGTH	type of the signal is uint8[(DcmDspRoutineSignalLength+7)/8]. This is only valid for the last signal and when DcmDspRoutineFixedLength is set to FALSE.

(4) DcmDspStopRoutineRole:

Provide the Role in the RoutineControl Stop request/response

7.1.5.14.5 DcmDspStartRoutineIn

Provide description of input parameter of Start subservice for RoutineControl service

Sub Container(s)	Value	Category
DcmDspStartRoutineInSignal	User Defined	C

DcmDspStartRoutineInSignal

Parameter Name	Value	Category
DcmDspRoutineSignalLength(1)	User Defined	C
DcmDspRoutineSignalPos(2)	User Defined	C
DcmDspRoutineSignalType(3)	User Defined	C

Parameter Name	Value	Category
DcmDspRoutineStartInEndianness		N

(1) DcmDspRoutineSignalLength:

Provide the length in bits of the signal in the RoutineControl request/response.

(2) DcmDspRoutineSignalPos:

Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.

The value of the parameter should be configured a multiple of 〈8〉

(3) DcmDspRoutineSignalType:

BOOLEAN	Not supported
SINT16	type of the signal is sint16. 〈Not Supported yet〉
SINT32	type of the signal is sint32. 〈Not Supported yet〉
SINT8	type of the signal is sint8. 〈Not Supported yet〉
UINT16	type of the signal is uint16.
UINT32	type of the signal is uint32.
UINT8	type of the signal is uint8.
VARIABLE_LENGTH	type of the signal is uint8[(DcmDspRoutineSignalLength+7)/8]. This is only valid for the last signal and when DcmDspRoutineFixedLength is set to FALSE.

7.1.5.14.6 DcmDspStartRoutineOut

Provide description of output parameter of Start subservice for RoutineControl service.

Sub Container(s)	Value	Category
DcmDspStartRoutineOutSignal	User Defined	C

DcmDspStartRoutineOutSignal

Parameter Name	Value	Category
DcmDspRoutineSignalLength(1)	User Defined	C
DcmDspRoutineSignalPos(2)	User Defined	C
DcmDspRoutineSignalType(3)	User Defined	C

Parameter Name	Value	Category
DcmDspStartRoutineRole	User Defined	C
DcmDspRoutineStartOutEndianness		N

(1) DcmDspRoutineSignalLength:

Provide the length in bits of the signal in the RoutineControl request/response.

(2) DcmDspRoutineSignalPos:

Provide the position of the signal in the RoutineControl request/response. The position is defined in bits. The value of the parameter should be configured a multiple of 〈8〉

(3) DcmDspRoutineSignalType:

BOOLEAN	Not supported
SINT16	type of the signal is sint16. 〈Not Supported yet〉
SINT32	type of the signal is sint32. 〈Not Supported yet〉
SINT8	type of the signal is sint8. 〈Not Supported yet〉
UINT16	type of the signal is uint16.
UINT32	type of the signal is uint32.
UINT8	type of the signal is uint8.
VARIABLE_LENGTH	type of the signal is uint8[(DcmDspRoutineSignalLength+7)/8]. This is only valid for the last signal and when DcmDspRoutineFixedLength is set to FALSE.

(4) DcmDspStartRoutineRole:

Provide the Role in the RoutineControl Start request/response

7.1.5.15 DcmDspSecurity

This container contains the configuration (DSP parameter) for security level configuration (per security level)

Description This container contains Rows of DcmDspSecurityRow

Parameter Name	Value	Category
DcmDspSecurityMaxAttemptCounterReadoutTime(1)	User Defined	C

(1) DcmDspSecurityMaxAttemptCounterReadoutTime:

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

Sub Container(s)	Value	Category
DcmDspSecurityRow	User Defined	C

7.1.5.15.1 DcmDspSecurityRow

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_<LEVEL>. The R-Port is named SecurityAccess_<LEVEL> where _<LEVEL> is the name of the container DcmDspSecurityRow. If there is no reference, no check of security level shall be done.

Parameter Name	Value	Category
DcmDspSecurityDelayTime	User Defined	C
DcmDspSecurityKeySize	User Defined	C
DcmDspSecurityLevel	From SRS	F
DcmDspSecurityNumAttDelay	User Defined	C
DcmDspSecuritySeedSize	User Defined	C
DcmDspSecurityGetSeedFnc(1)	User Defined	C
DcmDspSecurityUsePort(2)	User Defined	C
DcmDspSecurityGetCompareFnc(3)	User Defined	C
DcmDspSecurityADRSize	User Defined	C
DcmDspSecurityDelayTimeOnBoot(4)		N
DcmDspSecurityAttemptCounterEnabled(5)	User Defined	C
DcmDspSecurityGetAttemptCounterFnc(6)	User Defined	C
DcmDspSecuritySetAttemptCounterFnc(7)	User Defined	C

Note: Use fixed settings for (1), (2), and (3) if the CSAC algorithm is in use.

Note : (4) is Not supported in ES specification

(5) DcmDspSecurityAttemptCounterEnabled:

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.

(6) DcmDspSecurityGetAttemptCounterFnc:

Function name to request the value of an attempt counter. This parameter is related to the interface Xxx_ GetSecurityAttemptCounter

(7) DcmDspSecuritySetAttemptCounterFnc:

Function name to set the value of an attempt counter. This parameter is related to the interface Xxx_ SetSecurityAttemptCounter.

Note : (2) (5) (6) (7)

If (DcmDspSecurityAttemptCounterEnabled == TRUE) && (DcmDspSecurityUsePort == USE_ASYNCH_CLIENT_SERVER), Xxx_GetSecurityAttemptCounter()/Xxx_SetSecurityAttemptCounter() are generated as operations in the SecurityAccess_{SecurityLevel} Client-Server-Interface.

else If (DcmDspSecurityAttemptCounterEnabled == TRUE) && (DcmDspSecurityUsePort == USE_ASYNCH_FNC), Xxx_GetSecurityAttemptCounter()/Xxx_SetSecurityAttemptCounter() are generated as functions set in DcmDspSecurityGetAttemptCounterFnc/ DcmDspSecuritySetAttemptCounterFnc.

7.1.5.16 DcmDspSession

This container contains the configuration (DSP parameter) session control configuration (per session control) This container contains Rows of DcmDspSessionRow.

Sub Container(s)	Value	Category
DcmDspSessionRow	From SRS	F

7.1.5.16.1 DcmDspSessionRow

Parameter Name	Value	Category
DcmDspSessionForBoot		C
DcmDspSessionLevel	From SRS	F
DcmDspSessionP2ServerMax	0.05	F
DcmDspSessionP2StarServerMax	5.0	F

7.1.5.17 DcmDspTestResultByObdmid

Sub Container(s)	Value	Category
DcmDspTestResultObdmidTid	User Defined	C
DcmDspTestResultTid	User Defined	C

7.1.5.17.1 DcmDspTestResultByObdmidTid

Sub Container(s)	Value	Category
DcmDspTestResultObdmidTids	User Defined	C

7.1.5.17.2 DcmDspTestResultObdmidTids

Parameter Name	Value	Category
DcmDspTestResultObdmidTidUaSid	User Defined	C
DcmDspTestResultObdmidTidRef	User Defined	C

7.1.5.17.3 DcmDspTestResultTid

Parameter Name	Value	Category
DcmDspTestResultTestId	User Defined	C

7.1.5.18 DcmDspVehInfo

Parameter Name	Value	Category
DcmDspVehInfoInfoType	User Defined	C

Sub Container(s)	Value	Category
DcmDspVehInfoData	User Defined	C

7.1.5.18.1 DcmDspVehInfoData

Parameter Name	Value	Category
DcmDspVehInfoDataOrder	User Defined	C

Parameter Name	Value	Category
DcmDspVehInfoDataSize	User Defined	C
DcmDspVehInfoDataUsePort	User Defined	C
DcmDspVehInfoDataReadFnc	User Defined	C

7.1.5.19 DcmDspCallbackPresentDate

Container Name	Value	Category
DcmDspCallbackPresentDate (1)	User Defined	C

(1) DcmDspCallbackPresentDate:

The name of this container is used to define the name of the R-Port through which the DCM access the interface CallbackDCMPresentDate.

7.1.5.20 DcmDspAuthentication

This container contains the configuration of Authentication Service. This container contains Rows of DcmDspAuthenticationConnection.

Parameter Name	Value	Category
DcmDspAuthenticationDeauthenticatedRole(1)	User Defined	C
DcmDspAuthenticationDataBufferLength(2)	User Defined	C
DcmDspAuthenticationDefaultSessionTimeOut(3)	User Defined	C
DcmDspAuthenticationAsyncTimeOut(4)	User Defined	C
DcmDspAuthenticationRoleSize(5)	1	F
DcmDspAuthenticationWhiteListServicesMaxSize(6)	User Defined	C
DcmDspAuthenticationWhiteListDIDMaxSize(7)	User Defined	C
DcmDspAuthenticationWhiteListRIDMaxSize(8)	User Defined	C
DcmDspAuthenticationWhiteListMemorySelectionMaxSize(9)	User Defined	C
DcmDspAuthenticationGeneralNRCModeRuleRef	-	N
DcmDspAuthenticationPersistStateModeRuleRef	-	N

Parameter Name	Value	Category
DcmDspAuthenticationPersitStateNvMBlockIdRef	-	N

(1) DcmDspAuthenticationDeauthenticatedRole:
Role used for service authentication verification in the deauthenticated state.

(2) DcmDspAuthenticationDataBufferLength
The maximum returned data length when Dcm call Csm_RandomGenerate, KeyM_GetCertificate and Csm_SignatureGenerate

(3) DcmDspAuthenticationDefaultSessionTimeOut
The configuration number of seconds after which the Dcm makes a transition to deauthenticated state, in case of no active communication.

(4) DcmDspAuthenticationAsyncTimeOut
The configuration number of seconds waiting time after Dcm call an asynchronous function (KeyM and Csm).

(5) DcmDspAuthenticationRoleSize
Defines the size in bytes for the role element within a certificate.
This configuration maximum size role.

(6) DcmDspAuthenticationWhiteListServicesMaxSize
Defines the maximum size in bytes for the white list element within a certificate.
Maximum white list service data which get by call KeyM API.

(7) DcmDspAuthenticationWhiteListDIDMaxSize
Defines the maximum size in bytes for the white list element within a certificate.
Maximum white list DID data which get by call KeyM API.

(8) DcmDspAuthenticationWhiteListRIDMaxSize
Defines the maximum size in bytes for the white list element within a certificate.
Maximum white list RID data which get by call KeyM API.

(9) DcmDspAuthenticationWhiteListMemorySelectionMaxSize
Defines the maximum size in bytes for the white list element within a certificate.
Maximum white list Memory Selection data which get by call KeyM API.

Sub Container Name	Value	Category
DcmDspAuthenticationConnection	User Defined	C

7.1.5.20.1 DcmDspAuthenticationConnection

Parameter Name	Value	Category
DcmDspAuthenticationCertificatePublicKeyStoreJobRef (1)	User Defined	C
DcmDspAuthenticationClientCertificateRef (2)	User Defined	C
DcmDspAuthenticationClientChallengeSignJobRef (3)	User Defined	C
DcmDspAuthenticationConnectionCertificateRef (4)	User Defined	C
DcmDspAuthenticationConnectionMainConnectionRef (5)	User Defined	C
DcmDspAuthenticationECUCertificateRef (6)	User Defined	C
DcmDspAuthenticationPublicKeyElementRef (7)	User Defined	C
DcmDspAuthenticationRandomJobRef(8)	User Defined	C
DcmDspAuthenticationRoleElementRef(9)	User Defined	C
DcmDspAuthenticationVerifyProofOfOwnerShipClientJobRef(10)	User Defined	C
DcmDspAuthenticationWhiteListServicesElementRef(11)	User Defined	C
DcmDspAuthenticationECUCertificateKeyElementRef(12)	User Defined	C
DcmDspAuthenticationWhiteListDIDElementRef(13)	User Defined	C
DcmDspAuthenticationWhiteListMemorySelectionElementRef(14)	User Defined	C
DcmDspAuthenticationWhiteListRIDElementRef(15)	User Defined	C
DcmDspAuthenticationTargetIdentificationModeRuleRef	-	N

(1) DcmDspAuthenticationCertificatePublicKeyStoreJobRef:

Reference to a CsmJob used to store the public key within the Csm.

(2) DcmDspAuthenticationClientCertificateRef:
Reference to a KeyMCertificate used to handle the client certificate for this connection.
This configuration reference to a KeyMCertificateId
Dcm will call KeyM_CertElementGetFirst with certId = DcmDspAuthenticationClientCertificateRef

(3) DcmDspAuthenticationClientChallengeSignJobRef
Reference to a job used to sign the client challenge.
This configuration reference to a CsmJobId
Dcm will call Csm_SignatureGenerate with jobId = DcmDspAuthenticationClientChallengeSignJobRef

(4) DcmDspAuthenticationConnectionCertificateRef
Reference to a KeyMCertificate used to store the certificate within the KeyM.
This configuration reference to a KeyMCertificateId
Dcm will call KeyM_SetCertificate with certId = DcmDspAuthenticationConnectionCertificateRef
Dcm will call KeyM_VerifyCertificate with CertificateId = DcmDspAuthenticationConnectionCertificateRef

(5) DcmDspAuthenticationConnectionMainConnectionRef
Reference to the dsl diagnostic connection that uses this authentication configuration

(6) DcmDspAuthenticationECUCertificateRef
Reference to a KeyMCertificate with the server certificate for bi-directional authentication
This configuration reference to a KeyMCertificateId
Dcm will call KeyM_VerifyCertificate with CertificateId = DcmDspAuthenticationECUCertificateRef

(7) DcmDspAuthenticationPublicKeyElementRef
Reference to a certificate data element with the public key in the certificate.

(8) DcmDspAuthenticationRandomJobRef
Reference to a certificate parse job used to parse the authentication certificate.
This configuration reference to a CsmJobId
Dcm will call Csm_RandomGenerate with jobId = DcmDspAuthenticationRandomJobRef

(9) DcmDspAuthenticationRoleElementRef
Reference to a certificate data element with the role in the certificate
This configuration reference to a KeyMCertificateElementId
Dcm will call KeyM_CertElementGet with CertElementId = KeyMCertificateElementId

(10) DcmDspAuthenticationVerifyProofOfOwnerShipClientJobRef
Reference to a CsmJob used to verify the proof of ownership client in the Csm.
This configuration reference to a CsmJobId
Dcm will call Csm_SignatureVerify with jobId = DcmDspAuthenticationVerifyProofOfOwnerShipClientJobRef

(11) DcmDspAuthenticationWhiteListServicesElementRef
Reference to a certificate data element with the white list in the certificate.
This configuration reference to a KeyMCertificateElementId
Dcm will call KeyM_CertElementGetFirst and KeyM_CertElementGetNext with CertElementId = DcmDspAuthenticationWhiteListServicesElementRef

(12) DcmDspAuthenticationECUCertificateKeyElementRef

Reference to a CryptoKeyElement used as server certificate during bi-directional authentication.

(13) DcmDspAuthenticationWhiteListDIDElementRef
Reference to a certificate data element with the white list in the certificate.
This configuration reference to a KeyMCertificateElementId

Dcm will call KeyM_CertElementGetFirst and KeyM_CertElementGetNext with CertElementId = DcmDspAuthenticationWhiteListDIDElementRef

(14) DcmDspAuthenticationWhiteListMemorySelectionElementRef
Reference to a certificate data element with the white list in the certificate.
This configuration reference to a KeyMCertificateElementId
Dcm will call KeyM_CertElementGetFirst and KeyM_CertElementGetNext with CertElementId = DcmDspAuthenticationWhiteListMemorySelectionElementRef

(15) DcmDspAuthenticationWhiteListRIDElementRef
Reference to a certificate data element with the white list in the certificate.
This configuration reference to a KeyMCertificateElementId
Dcm will call KeyM_CertElementGetFirst and KeyM_CertElementGetNext with CertElementId = DcmDspAuthenticationWhiteListRIDElementRef

7.1.5.20.2 DcmDspAuthenticationConnectionES

Parameter Name	Value	Category
DcmDspAuthenticatedRole(1)	User Defined	C
DcmDspAuthenticationWLServicesWithoutSubfunction(2)	User Defined	C
DcmDspAuthenticationWLServicesWithSubfunction (3)	User Defined	C
DcmDspAuthenticationUniDirectionalFunc	User Defined	C
DcmDspAuthenticationProofOfOwnerShipClientFunc	User Defined	C
DcmDspAuthenticationCertificateClientSize	User Defined	C
DcmDspAuthenticationProofOfOwnerShipClientSize	User Defined	C
DcmDspAuthenticationWLDID(4)	User Defined	C
DcmDspAuthenticationWLRID(5)	User Defined	C
DcmDspAuthenticationWhiteListMemorySelection	User Defined	C
DcmDspAuthenticationSettingAccessRighsFaileddFunc(6)	User Defined	C
DcmDspAuthenticationDeauthenticationFailedFunc(7)	User Defined	C
DcmDspAuthenticationUsePort	User Defined	C
DcmDspAuthenticationConnectionMainConnectionRef (8)	User Defined	C

(1) DcmDspAuthenticatedRole
Role used for service authentication verification in the authenticated state.

(2) DcmDspAuthenticationWLServicesWithoutSubfunction
If set Service SID Without Subfunction, Service can be used when authenticated state.

(3) DcmDspAuthenticationWLServicesWithSubfunction
If set Service SID With Subfunction, Service can be used when authenticated state.

(4) DcmDspAuthenticationWLDID
If set DID, DID can be used when authenticated state. DID must be set as 3 byte: data identifiers and access information.

(5) DcmDspAuthenticationWLRID
If set RID, RID can be used when authenticated state. RID must be set as 3 byte: data identifiers and access information.

(6) DcmDspAuthenticationSettingAccessRighsFailedFunc
When using NRC 5A, Configure this option when DcmDspAuthenticationUsePort is set as USE_ASYNCH_FNC /USE_SYUNCH_FNC.

(7) DcmDspAuthenticationDeauthenticationFailedFunc
When using NRC 5D, Configure this option when DcmDspAuthenticationUsePort is set as USE_ASYNCH_FNC /USE_SYUNCH_FNC.

(8) DcmDspAuthenticationConnectionMainConnectionRef
Reference to the dsl diagnostic connection that uses this authentication configuration

7.1.5.21 DcmDspReadDTCInformation

Container Name	Value	Category
DcmDspReadDTCInformationSupportedObdUdsDtcSeparation (1)	User Defined	C

(1) DcmDspReadDTCInformationSupportedObdUdsDtcSeparation:

Set only when using J1979-2
It needs to be set the same as DemSupportedObdUdsDtcSeparation in Dem to operate normally.

7.1.5.22 DcmDspRequestFileTransfer

Parameter Name	Value	Category
DcmRequestFileTransferUsePort (1)	User Defined	C
DcmRequestFileTransferFileSizeOrDirInfoParameterLength	User Defined	C

Parameter Name	Value	Category
(2)		
DcmRequestFileTransferLengthFormatIdentifier (3)	User Defined	C
DcmRequestFileTransferMaxFileAndDirName (4)	User Defined	C

(1) DcmRequestFileTransferUsePort:

Defines if a C/S or C function call shall be used for RequestFileTransfer processing.

(2) DcmRequestFileTransferFileSizeOrDirInfoParameterLength:

Defines the length (number of bytes, i.e. the value of fileSizeOrDirInfoParameterLength) of the fileSizeUncompressedOrDirInfoLength and fileSizeCompressed in the response of RequestFileTransfer.

(3) DcmRequestFileTransferLengthFormatIdentifier:

Defines the length (number of bytes) of the maxNumberOfBlockLength parameter in the response of RequestFileTransfer.

(4) DcmRequestFileTransferMaxFileAndDirName:

Defines the maximum size allowed for the FileAndDirName parameter with RTE interfaces used for RequestFileTransfer.

7.1.6 DcmProcessingConditions

Sub Container(s)	Value	Category
DcmModeCondition		N
DcmModeRule		N

7.1.6.1 DcmModeCondition

Parameter Name	Value	Category
DcmConditionType		N
DcmBswModeRef		N
DcmSwcModeRef		N

7.1.6.2 DcmModeRule

Parameter Name	Value	Category
DcmLogicalOperator		N
DcmModeRuleNrcValue		N
DcmArgumentRef		N

8 Application Programming Interface (API)

8.1 Type Definitions

8.1.1 Dcm_StatusType

Name:	Dcm_StatusType	
Type:	uint8	
Range:	DCM_E_OK	0x00
	DCM_E_COMPARE_KEY_FAILED	*0x01*
	DCM_E_TI_PREPARE_LIMITS	0x02
	DCM_E_TI_PREPARE_INCONSTENT	0x03
	DCM_E_SESSION_NOT_ALLOWED	0x04
	DCM_E_PROTOCOL_NOT_ALLOWED	0x05
	DCM_E_ROE_NOT_ACCEPTED	0x06
	DCM_E_PERIODICID_NOT_ACCEPTED	0x07
	DCM_E_REQUEST_NOT_ACCEPTED	0x08
	DCM_E_REQUEST_ENV_NOK	0x09
Description:	Base item type to transport status information.	

8.1.2 Dcm_SecLevelType

Name:	Dcm_SecLevelType		
Type:	uint8		
Range:	DCM_SEC_LEV_LOCKED	0x01	--
	DCM_SEC_LEV_L1	0x02	--
	configuration dependent	0x03	--

	DCM_SAFETY_SYSTEM_DIAGNOSTIC_SESSION	0x04	--
	configuration dependent	0x02...0x7F	(according to "diagnosticSessionType" parameter of DiagnosticSessionControl request)
	Reserved by Document	0x80...0xFE	--
	DCM_SEC_LEV_ALL	0xFF	--
Description:	Security Level type definition		

Note : This type is defined in Rte_Dcm_Type.h header file, which is generated by the RTE generator.

8.1.3 Dcm_SesCtrlType

Name:	Dcm_SesCtrlType		
Type:	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 DiagnosticSessionControl request)
	Reserved by Document	0x7F...0xFE	--
	DCM_ALL_SESSION_LEVEL	0xFF	--
Description:	Session type definition		

8.1.4 Dcm_ProtocolType

Name:	Dcm_ProtocolType		
Type:	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
	Reserved for further AUTOSAR implementation	0x07..0xEF	--
	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_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		

Note : This type is defined in Rte_Dcm_Type.h header file, which is generated by the RTE generator.

8.1.5 Dcm_NegativeResponseCodeType

Name:	Dcm_NegativeResponseCodeType		
Type:	uint8		
Range:	DCM_E_POSITIVERESPONSE	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_FAILUREPREVENTSEXECUTIONOFREQUESTEDACTION	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
	DCM_E_REQUESTCORRECTLYRECEIVEDRESPONSEPENDING	0x78	RCRRP
	range of values 0x79..0x7D reserved by ISO 14229	0x79..0x7D	ISOSAERESRVD
	DCM_E_SUBFUNCTIONNOTSUPPORTEDINACTIVESESSION	0x7E	SFNSIAS
	DCM_E_SERVICENOTSUPPORTEDINACTIVESESSION	0x7F	SNSIAS
	value 0x80 reserved by ISO 14229	0x80	ISOSAERESRVD
	DCM_E_RPMTOOHIGH	0x81	RPMTH
	DCM_E_RPMTOOLOW	0x82	RPMTL
	DCM_E_ENGINEISRUNNING	0x83	EIR
	DCM_E_ENGINEISNOTRUNNING	0x84	EINR
	DCM_E_ENGINERUNTIMETOOLOW	0x85	ERTTL
	DCM_E_TEMPERATURETOOHIGH	0x86	TEMPTH
	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_TRANSMISSIONRANGENOTINNEUTRAL	0x8C	TRNIN
	DCM_E_TRANSMISSIONRANGENOTINGEAR	0x8D	TRNIG
	value 0x8E reserved by ISO 14229	0x8E	ISOSAERESRVD
	DCM_E_BRAKESWITCH_NOTCLOSED	0x8F	BSNC
	DCM_E_SHIFTERLEVERNOTINPARK	0x90	SLNIP
	DCM_E_TORQUECONVERTERCLUTCHLOCKED	0x91	TCCL
	DCM_E_VOLTAGETOOHIGH	0x92	VTH

	DCM_E_VOLTAGETOOLOW	0x93	VTL
	range of values 0x94..0xEF reserved by ISO 14229	0x94..0xEF	RFSCNC
	DCM_E_CRLINTEGRITYCHECKFAILED	0xF0	CICF
	DCM_E_CRLEXPIRED	0xF1	CE
	DCM_E_CERTVERIFICATIONFAILED	0xF2	CVF
	range of values 0xF3..0xFE reserved by ISO 14229	0xF3...0xFE	RFSCNC
	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).		

Note : This type is defined in Rte_Dcm_Type.h header file, which is generated by the RTE generator.

8.1.6 Dcm_CommunicationModeType

Name:	Dcm_CommunicationModeType		
Type:	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	Diable 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:	--		

8.1.7 Dcm_ConfigType

Name:	Dcm_ConfigType
Type:	Structure
Range:	Implementation specific
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.

8.1.8 Dcm_ConfirmationStatusType

Name:	Dcm_ConfirmationStatusType		
Type:	uint8		
Range:	DCM_RES_POS_OK	0x00	Indicates the type of the positive response

			when E_OK is returned.
	DCM_RES_POS_NOT_OK	0x01	Indicates the type of the positive response when E_NOT_OK is returned.
	DCM_RES_NEG_OK	0x02	Indicates the type of the Negative response when E_NOT_OK is returned.
	DCM_RES_NEG_NOT_OK	0x03	Indicates the type of the Negative response when E_PENDING is returned.
Description:	--		

8.1.9 Dcm_OpStatusType

Name:	Dcm_OpStatusType		
Type:	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:	--		

8.1.10 Dcm_ReturnReadMemoryType

Name:	Dcm_ReturnReadMemoryType		
Type:	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		

8.1.11 Dcm_ReturnWriteMemoryType

Name:	Dcm_ReturnWriteMemoryType		
Type:	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		

8.1.12 Dcm_RoeStateType

Name:	Dcm_RoeStateType		
Type:	uint8		
Range:	DCM_ROE_ACTIVE	0x00	--
	DCM_ROE_UNACTIVE	0x01	--
Description:	--		

8.1.13 Dcm_EcuStartModeType

Name:	Dcm_EcuStartModeType		
Type:	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		

8.1.14 Dcm_ProgConditionsType

Name:	Dcm_ProgConditionsType		
Type:	Structure		
Element:	uint8	ProtocolId	Id of the protocol on which the request has been received
	uint8	TesterSourceAddr	Tester source address configured per protocol
	uint8	Sid	Service identifier of the received request
	uint8	SubFncId	Identifier of the received subfunction
	boolean	ReprogramingRequest	Set to true in order to request reprogramming of the ECU. HIS representation of FL_ExtProgRequestType.
	boolean	ApplUpdated	Indicate whether the application has been updated or not. HIS representation of FL_ApplicationUpdateType.
	boolean	ResponseRequired	Set to true in case the flashloader or application shall send a response. HIS representation of FL_ResponseRequiredType.
Description:	Used in Dcm_SetProgConditions() to allow the integrator to store relevant information prior to jumping to bootloader.		

8.1.15 Dcm_MsgItemType

Name:	Dcm_MsgItemType
Type:	uint8
Description:	Base type for diagnostic message item

8.1.16 Dcm_MsgType

Name:	Dcm_MsgType
Type:	Dcm_MsgItemType*
Description:	Base type for diagnostic message (request, positive or negative response)

8.1.17 Dcm_MsgLenType

Name:	Dcm_MsgLenType
Type:	uint32
Description:	Length of diagnostic message (request, positive or negative response). The maximum length is dependent of the underlying transport protocol/media. E. g. the maximum message length for CAN Transport Layer is 4095bytes.

8.1.18 Dcm_MsgAddInfoType

Name:	Dcm_MsgAddInfoType		
Type:	Structure		
Element:	Bit0	reqType	0 = physical request 1 = functional request
	Bit1	suppressPosResponse	0 = no (do not suppress) 1 = yes (no positive response will be sent)
Description:	Additional information on message request. Datastructure: Bitfield		

8.1.19 Dcm_IdContextType

Name:	Dcm_IdContextType
Type:	uint8
Description:	This message context identifier can be used to determine the relation between request and response confirmation.

8.1.20 Dcm_MsgContextType

Name:	Dcm_MsgContextType		
Type:	Structure		
Element:	Dcm_MsgType	reqData	Request data, starting directly after service identifier (which is not part of this data)
	Dcm_MsgLenType	reqDataLen	Request data length (excluding service identifier)
	Dcm_MsgType	resData	Positive response data, starting directly after service identifier (which is not part of this data).
	Dcm_MsgLenType	resDataLen	Positive response data length (excluding service identifier)
	Dcm_MsgAddInfoType	msgAddInfo	Additional information about service request and response (see: Dcm_MsgAddInfo)
	Dcm_MsgLenType	resMaxDataLen	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.
	Dcm_IdContextType	idContext	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.
	PduIdType	dcmRxPduId:	Pdu identifier on which the request was received. The PduId of the request can have consequences for message processing. E. g. an OBD request will be received on the OBD PduId 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.		

8.1.21 Dcm_AuthenticationRoleType

Name:	Dcm_AuthenticationRoleType
Type:	Array
Description:	This array type of a Role for Authentication Service.

8.2 Macro Constants

None

8.3 Interfaces

8.3.1 DCMServices

8.3.1.1 GetSecurityLevel

Function Name	Xxx_GetSecurityLevel	
Syntax:	FUNC(Std_ReturnType, DCM_CODE) Xxx_GetSecurityLevel (P2VAR(Dcm_SecLevelType, AUTOMATIC, DCM_APPL_DATA) SecLevel)	
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.	
Preconditions	The Dcm module must be initialized	
Configuration Dependency	This API is available only if configuration parameter DcmDspSecurityLevel in the container DcmDspSecurityRow is configured.	

8.3.1.2 GetSesCtrlType

Function Name	Xxx_GetSesCtrlType	
Syntax:	FUNC(Std_ReturnType, DCM_CODE) Xxx_GetSesCtrlType (P2VAR(Dcm_SesCtrlType, AUTOMATIC, DCM_APPL_DATA) SesCtrlType)	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	None	
Parameters (Inout)	None	
Parameters (Out)	SesCtrlType	Active Session Control Type value Content is according to “Diagnostic Session Type” of Diagnostic Session Control Request
Return Value	Std_ReturnType	E_OK: This Value is always returned.
Description	This function provides the active session control type value.	
Preconditions	The Dcm module must be initialized	
Configuration Dependency	This API is available only if configuration parameter DcmDspSessionLevel in the container DcmDspSessionRow is configured.	

8.3.1.3 GetActiveProtocol

Function Name	Xxx_GetActiveProtocol	
Syntax:	FUNC(Std_ReturnType, DCM_CODE Xxx_GetActiveProtocol (P2VAR(Dcm_ProtocolType, AUTOMATIC, DCM_APPL_DATA)ActiveProtocol))	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	None	
Parameters (Inout)	None	
Parameters (Out)	ActiveProtocol	ActiveProtocol Type value
Return Value	Std_ReturnType	E_OK: This Value is always returned.
Description	This service reads and returns the value of current active protocol	

Preconditions	The Dcm module must be initialized
Configuration Dependency	None

8.3.1.4 ResetToDefaultSession

Function Name	Xxx_ResetToDefaultSession	
Syntax:	FUNC(void, DCM_CODE) Xxx_ResetToDefaultSession(void)	
Service ID	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 Defaultsession. Example: Automatic termination of an extended diagnostic session upon exceeding of a speed limit.	
Preconditions	None.	
Configuration Dependency	None	

8.3.1.5 Dcm_SetDeauthenticatedRole

Function Name	Dcm_SetDeauthenticatedRole
Syntax:	FUNC(void, DCM_CODE) Dcm_SetDeauthenticatedRole (uint16 connectionId, Dcm_AuthenticationRoleType deauthenticatedRole)
Service ID	0x79
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	connectionId

	deauthenticatedRole	
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	
Preconditions	None.	
Configuration Dependency	None	

8.3.2 Memory Callout

Note Refer to Dcm_Callouts.c

8.3.2.1 Dcm_ReadMemory

Function Name	Dcm_ReadMemory	
Syntax:	FUNC(Dcm_ReturnReadMemoryType, DCM_CODE) Dcm_ReadMemory(Dcm_OpStatusType OpStatus, uint8 MemoryIdentifier, uint32 MemoryAddress,uint32 MemorySize, P2VAR(uint8, AUTOMATIC,DCM_APPL_DATA) MemoryData)	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	DCM_INITIAL: All In-parameters are valid DCM_PENDING: All In-parameters are set to 0x0 DCM_CANCEL: All In-parameters are set to 0x0 DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x0

	MemoryIdentifier	Identifier of the Memory Block 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)
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	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: - ReadMemoryByAdress - RequestUpload - ReadDataByIdentifier (in case of Dynamical DID defined by memory address)	
Preconditions	DCM module must be initialised	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 35 and 53 and the macro(s)DCM_READ_MEMORY_BY_ADDRESS and DCM_REQUEST_UPLOAD_SERVICE are STD_ON.	

8.3.2.2 Dcm_WriteMemory

Function Name	Dcm_WriteMemory	
Syntax:	FUNC(Dcm_ReturnReadMemoryType, DCM_CODE) Dcm_ReadMemory(Dcm_OpStatusType LddOpStatus, uint8 LucMemoryIdentifier, uint32 LulMemoryAddress, uint32 LulMemorySize, P2VAR(uint8, AUTOMATIC, DCM_APPL_DATA) LpMemoryData)	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	DCM_INITIAL: All In-parameters are valid DCM_PENDING: All In-parameters are set to 0x0 DCM_CANCEL: All In-parameters are set to 0x0 DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x0
	MemoryIdentifier	Identifier of the Memory Block 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.
	MemorySize	Number of bytes in the MemoryData
	MemoryData	Data to write (Points to the diagnostic buffer in DCM)
Parameters (Inout)	None	
Parameters (Out)	None	
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 : -WriteMemoryByAdress - RequestDownload	
Preconditions	DCM module must be initialised	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 61 and 52 and the macro(s)DCM_WRITE_MEMORY_BY_ADDRESS and DCM_REQUEST_DOWNLOAD_SERVICE are STD_ON.	

8.3.3 ProgConditions Callout

Note Refer to Dcm_Callouts.c. If you use the AutoEver Fbl, don't modify callout code provided.

8.3.3.1 Dcm_SetProgConditions

Function Name	Dcm_SetProgConditions	
Syntax:	Std_ReturnType Dcm_SetProgConditions(Dcm_OpStatusType OpStatus, Dcm_ProgConditionsType * ProgConditions)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	DCM_INITIAL: All In-parameters are valid DCM_PENDING: All In-parameters are set to

		0x0 DCM_CANCEL: All In-parameters are set to 0x0 DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x0
	ProgConditions	Conditions on which the jump to bootloader has been requested
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	Std_ReturnType	E_OK: Transfer was successful E_NOT_OK: Transfer was not successful DCM_E_PENDING: Transfer is not yet finished
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.	

8.3.3.2 Dcm_GetProgConditions

Function Name	Dcm_GetProgConditions	
Syntax:	Dcm_EcuStartModeType Dcm_GetProgConditions(Dcm_ProgConditionsType * ProgConditions)	
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 parameter are defined in Dcm_ProgConditionsType.	

8.3.4 RequestDownload and Transfer Callout

8.3.4.1 Dcm_ProcessRequestTransferExit

Function Name	Dcm_ProcessRequestTransferExit	
Syntax:	Std_ReturnType Dcm_ProcessRequestTransferExit(Dcm_OpStatusType LucOpStatus, P2VAR(uint8, AUTOMATIC, DCM_APPL_DATA) LpMemoryData, uint32* LulParameterRecordSize, P2VAR(Dcm_NegativeResponseCodeType, AUTOMATIC, DCM_PRIVATE_DATA)LpNegativeErrorCode)	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	DCM_INITIAL: All In-parameters are valid DCM_PENDING: All In-parameters are set to 0x0 DCM_CANCEL: All In-parameters are set to 0x0 DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x0
	ParameterRecord	(Optional) Pointer to vehicle-manufacturer-specific data
	ParameterRecordSize	(Optional) Length of ParameterRecord in bytes
Parameters (Inout)	None	
Parameters (Out)	ErrorCode	See below
Return Value	Std_ReturnType	E_OK: Transfer was successful E_NOT_OK: Transfer was not successful DCM_E_PENDING: Transfer is not yet finished
Description	Calloutfunction.	

	DCM shall call this callout function to terminate a download or upload process. This service is needed for the implementation of UDS service RequestTransferExit.
Preconditions	None
Configuration Dependency	None

8.3.4.2 Dcm_ProcessRequestUpload

Function Name	Dcm_ProcessRequestUpload	
Syntax:	FUNC(Std_ReturnType, DCM_CODE) Dcm_ProcessRequestUpload(Dcm_OpStatusType OpStatus, uint8 DataFormatIdentifier, uint32 MemoryAddress, uint32 MemorySize, P2VAR(uint32,AUTOMATIC,DCM_PRIVATE_DATA)LpBlockLength, P2VAR(Dcm_NegativeResponseCodeType, AUTOMATIC, DCM_PRIVATE_DATA)LpNegativeErrorCode)	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	DCM_INITIAL: All In-parameters are valid DCM_PENDING: All In-parameters are set to 0x0 DCM_CANCEL: All In-parameters are set to 0x0 DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x0
	DataFormatIdentifier	Bit 7 - 4: Compression Method - 0x0: not compressed - 0x1..F: vehicle-manufacturer-

		specific Bit 3 - 0: Encrypting method - 0x0: not encrypted - 0x1..F: vehicle-manufacturer-specific
	MemoryAddress	Starting address of server memory from which data are to be copied
	MemorySize	Uncompressed memory size in bytes
Parameters (Inout)	None	
Parameters (Out)	ErrorCode	See below
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. If you need to check the memory address range, you can implement this function.	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 35 and 53 and the macro(s)DCM_READ_MEMORY_BY_ADDRESS and DCM_REQUEST_UPLOAD_SERVICE are STD_ON.	

8.3.4.3 Dcm_ProcessRequestDownload

Function Name	Dcm_ProcessRequestDownload

Syntax:	FUNC(Std_ReturnType, DCM_CODE) Dcm_ProcessRequestDownload(Dcm_OpStatusType OpStatus, uint8 DataFormatIdentifier, uint32 MemoryAddress, uint32 MemorySize, P2VAR(uint32, AUTOMATIC, CM_PRIVATE_DATA)LpBlockLength, P2VAR(Dcm_NegativeResponseCodeType, AUTOMATIC, DCM_PRIVATE_DATA)LpNegativeErrorCode)	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	DCM_INITIAL: All In-parameters are valid DCM_PENDING: All In-parameters are set to 0x0 DCM_CANCEL: All In-parameters are set to 0x0 DCM_FORCE_RCRRP_OK: All In-parameters are set to 0x0
	DataFormatIdentifier	Bit7-4: Compression Method -0x0:not compressed -0x1..F: vehicle-manufacturer-specific Bit3-0: Encrypting method -0x0: not encrypted -0x1..F:vehicle-manufacturer-specific
	MemoryAddress	Starting address of server memory to which data is to be written
	MemorySize	Uncompressed memory size in bytes
Parameters (Inout)	None	
Parameters (Out)	BlockLength	Max. Number of bytes for one

		Dcm_WriteMemory
	ErrorCode	See below
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. If you need to check the memory address range, you can implement this function.	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 61 and 52 and the macro(s)DCM_WRITE_MEMORY_BY_ADDRESS and DCM_REQUEST_DOWNLOAD_SERVICE are STD_ON.	

8.3.5 DataService_{Data}

8.3.5.1 Read Asynchronous

8.3.5.1.1 Xxx_ReadData

Function Name	Xxx_ReadData
Syntax:	Std_ReturnTypeXxx_ReadData(Dcm_OpStatusType OpStatus, uint8*Data)
Sync/Async	Asynchronous
Reentrancy	Non Reentrant
Parameters (Inout)	None

Parameters (Out)	Data	Buffer where the requested data shall be copied to
Return Value	Std_ReturnType	E_OK: Request was successful.
Description	This function requests to the application a data value of a DID/PID if DcmDspDataUsePort is set to USE_DATA_SYNCH_CLIENT_SERVER.	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 34, and the configuration parameter DcmDspDataUsePort is configured as either “USE_DATA_ASYNCH_CLIENT_SERVER”	

8.3.5.1.2 Xxx_ConditionCheckRead

Function Name	Xxx_ConditionCheckRead	
Syntax:	Std_ReturnTypeXxx_ConditionCheckRead(Dcm_NegativeResponseCodeType *ErrorCode)	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	Status of the current operation
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 if the conditions to read the Data are correct.
Preconditions	None
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 34, and the configuration parameter DcmDspDataUsePort is configured as either " USE_DATA_ASYNCH_CLIENT_SERVER/ USE_DATA_ASYNCH_FNC"

8.3.5.2 Read Synchronous

8.3.5.2.1 Xxx_ReadData

Function Name	Xxx_ReadData	
Syntax:	Std_ReturnType Xxx_ReadData(uint8*Data)	
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: 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_ASYNCH_CLIENT_SERVER.	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 34, and the configuration parameter DcmDspDataUsePort is configured as either "USE_DATA_SYNCH_CLIENT_SERVER"	

8.3.5.2.2 Xxx_ConditionCheckRead

Function Name	Xxx_ConditionCheckRead	
Syntax:	Std_ReturnTypeXxx_ConditionCheckRead(Dcm_NegativeResponseCodeType*ErrorCode)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	Status of the current operation
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 if the conditions to read the Data are correct.	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 34, and the configuration parameter DcmDspDataUsePort is configured as either “USE_DATA_SYNCH_CLIENT_SERVER/ USE_DATA_SYNCH_FNC”	

8.3.5.3 Write Fixed Length

8.3.5.3.1 Xxx_WriteData

Function Name	Xxx_WriteData
Syntax:	Std_ReturnTypeXxx_WriteData(uint8*Data, Dcm_OpStatusType OpStatus,

	Dcm_NegativeResponseCodeType*ErrorCode)	
Sync/Async	Asynchronous/Synchronous	
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	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 the application to write a data value of a DID.	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 46, and the configuration parameter DcmDspDataUsePort is configured as either “USE_DATA_SYNCH_CLIENT_SERVER/ USE_DATA_ASYNCH_CLIENT_SERVER/ USE_DATA_SYNCH_FNC/ USE_DATA_ASYNCH_FNC” and DcmDspDataFixedLength is set to TRUE.	

8.3.5.4 Write Variable Length

8.3.5.4.1 Xxx_WriteData

Function Name	Xxx_WriteData
Syntax:	Std_ReturnTypeXxx_WriteData(uint8*Data,

	uint16 DataLength, Dcm_OpStatusType OpStatus, Dcm_NegativeResponseCodeType*ErrorCode)	
Sync/Async	Asynchronous/Synchronous	
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	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 the application to write a data value of a DID.	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 46, and the configuration parameter DcmDspDataUsePort is configured as either “USE_DATA_SYNCH_CLIENT_SERVER/ USE_DATA_ASYNCH_CLIENT_SERVER/ USE_DATA_SYNCH_FNC/ USE_DATA_ASYNCH_FNC” and DcmDspDataFixedLength is set to TRUE	

8.3.5.5 Xxx_ReadDataLength Variable Length

Function Name	Xxx_ReadDataLength
Syntax:	Std_ReturnTypeXxx_ReadDataLength(

	uint16*DataLength)	
Sync/Async	Asynchronous/Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	None	
Parameters (Inout)	None	
Parameters (Out)	DataLength	Length of the data to be writen/read
Return Value	Std_ReturnType	E_OK: this value is always returned.
Description	This function requests the application to return the data length of a Data.	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 34, and the configuration parameter DcmDspDataUsePort is configured as either “USE_DATA_SYNCH_CLIENT_SERVER/ USE_DATA_ASYNCH_CLIENT_SERVER/ USE_DATA_SYNCH_FNC/ USE_DATA_ASYNCH_FNC”	

8.3.5.6 Xxx_GetScalingInformation

Function Name	Xxx_GetScalingInformation	
Syntax:	Std_ReturnTypeXxx_GetScalingInformation(Dcm_OpStatusType OpStatus, uint8*ScalingInfo, Dcm_NegativeResponseCodeType*ErrorCode)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	Status of the current operation
Parameters (Inout)	None	
Parameters (Out)	ScalingInfo	Scaling information
	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 for the scaling information of a Data.	
Preconditions	None	
Configuration Dependency	None	

8.3.5.7 Xxx_ReturnControlToECU

Function Name	Xxx_ReturnControlToECU	
Syntax:	Std_ReturnTypeXxx_ReturnControlToECU(Dcm_OpStatusTypeOpStatus, Dcm_NegativeResponseCodeType*ErrorCode)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	Status of the current operation
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 return control to ECU of an IOControl.
Preconditions	None
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 47, and the configuration parameter DcmDspDataUsePort is configured as either “USE_DATA_SYNCH_CLIENT_SERVER/ USE_DATA_ASYNCH_CLIENT_SERVER/ USE_DATA_SYNCH_FNC/ USE_DATA_ASYNCH_FNC”

8.3.5.8 Xxx_ResetToDefault

Function Name	Xxx_ResetToDefault	
Syntax:	Std_ReturnTypeXxx_ResetToDefault(Dcm_OpStatusTypeOpStatus, Dcm_NegativeResponseCodeType*ErrorCode)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	Status of the current operation
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 reset an IOControl to default value.	
Preconditions	None	
Configuration	This API is available only if configuration parameter	

Dependency	DcmDsdSidTabServiceId is configured as 47, and the configuration parameter DcmDspDataUsePort is configured as either “USE_DATA_SYNCH_CLIENT_SERVER/ USE_DATA_ASYNCH_CLIENT_SERVER/ USE_DATA_SYNCH_FNC/ USE_DATA_ASYNCH_FNC”

8.3.5.9 Xxx_FreezeCurrentState

Function Name	Xxx_FreezeCurrentState	
Syntax:	Std_ReturnTypeXxx_FreezeCurrentState(Dcm_OpStatusTypeOpStatus, Dcm_NegativeResponseCodeType*ErrorCode)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	Status of the current operation
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 freeze the current state of an IOControl.	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 47, and the configuration parameter DcmDspDataUsePort is configured as either “USE_DATA_SYNCH_CLIENT_SERVER/ USE_DATA_ASYNCH_CLIENT_SERVER/ USE_DATA_SYNCH_FNC/	

	USE_DATA_ASYNCH_FNC”

8.3.5.10 Xxx_ShortTermAdjustment

Function Name	Xxx_ShortTermAdjustment	
Syntax:	Std_ReturnTypeXxx_ShortTermAdjustment(uint8*ControlOptionRecord, Dcm_OpStatusTypeOpStatus, Dcm_NegativeResponseCodeType*ErrorCode)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	ControlOptionRecord	Control option parameter for the adjustment request
	OpStatus	Status of the current operation
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.	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 47, and the configuration parameter DcmDspDataUsePort is configured as either “USE_DATA_SYNCH_CLIENT_SERVER/ USE_DATA_ASYNCH_CLIENT_SERVER/ USE_DATA_SYNCH_FNC/ USE_DATA_ASYNCH_FNC”	

8.3.6 DataServices_DIDRange_{Range}

8.3.6.1 Xxx_IsDidAvailable

Function Name	Xxx_IsDidAvailable	
Syntax:	Std_ReturnTypeXxx_IsDidAvailable(uint16DID, uint8*supported)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	DID	DID value
Parameters (Inout)	None	
Parameters (Out)	supported	Indicate if the DID is available within the range or not
Return Value	Std_ReturnType	E_OK: this value is always returned.
Description	This function requests if a specific DID is available within the range or not.	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 34, and the container DcmDspDidRange needs to be configured.	

8.3.6.2 Xxx_ReadDidData

Function Name	Xxx_ReadDidData
Syntax:	Std_ReturnTypeXxx_ReadDidData(uint16DID, uint8*Data, Dcm_OpStatusTypeOpStatus, uint16DataLength, Dcm_NegativeResponseCodeTypeErrorCode)

Sync/Async	Synchronous	
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	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 a data value of a DID	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 34, and the container DcmDspDidRange needs to be configured.	

8.3.6.3 Xxx_WriteDidData

Function Name	Xxx_WriteDidData
Syntax:	Std_ReturnTypeXxx_WriteDidData(uint16DID, uint8*Data, Dcm_OpStatusTypeOpStatus, uint16DataLength, Dcm_NegativeResponseCodeTypeErrorCode)
Sync/Async	Synchronous

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	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 the application to write a data value of a DID.	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 47, and the container DcmDspDidRange needs to be configured.	

NOTE : when DcmDspSecurityADRSize is present following signature will be used.

8.3.7 SecurityAccess_{SecurityLevel}

8.3.7.1 Asynchronous Operations

8.3.7.1.1 Xxx_GetSeed [SecurityAccessDataRecord on]

Function Name	Xxx_ GetSeed
Syntax:	Std_ReturnType Xxx_GetSeed(uint8* SecurityAccessDataRecord,

	Dcm_OpStatusType OpStatus, uint8* Seed, Dcm_NegativeResponseCodeType* ErrorCode)	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (In)	SecurityAccessDataRecord	Contain security access data record to be written.
	OpStatus	Status of the current operation
Parameters (Inout)	None	
Parameters (Out)	ErrorCode	NRC to be sent in the negative response in case of failure
	Seed	Buffer where the requested seed value shall be copied to
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 get seed value	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDspSecurityADRSize is configured and configuration parameter DcmDspSecurityUsePort is equal to either USE_ASYNCH_CLIENT_SERVER or USE_ASYNCH_FNC	

8.3.7.1.2 Xxx_GetSeed [SecurityAccessDataRecord off]

Function Name	Xxx_ GetSeed
Syntax:	Std_ReturnType Xxx_GetSeed(Dcm_OpStatusType OpStatus, uint8* Seed,

	Dcm_NegativeResponseCodeType* ErrorCode)	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
	OpStatus	Status of the current operation
Parameters (Inout)	None	
Parameters (Out)	ErrorCode	NRC to be sent in the negative response in case of failure
	Seed	Buffer where the requested seed value shall be copied to
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 get seed value	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDspSecurityADRSize is not configured and configuration parameter DcmDspSecurityUsePort is equal to either USE_ASYNCH_CLIENT_SERVER or USE_ASYNCH_FNC.	

8.3.7.1.3 Xxx_CompareKey

Function Name	Xxx_CompareKey	
Syntax:	Std_ReturnType Xxx_CompareKey(uint8* Key, Dcm_OpStatusType OpStatus,)	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (Out)	Key	Key, which needs to be compared

	OpStatus	Status of the current operation
Parameters (Inout)	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	Request to application for comparing key	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDspSecurityUsePort is equal to either USE_ASYNCH_CLIENT_SERVER or USE_ASYNCH_FNC.	

8.3.7.1.4 Xxx_GetSecurityAttemptCounter

Function Name	Xxx_GetSecurityAttemptCounter	
Syntax:	Std_ReturnType Xxx_GetSecurityAttemptCounter (Dcm_OpStatusType OpStatus, uint8* AttemptCounter)	
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.
Preconditions	None
Configuration Dependency	This API is available only if configuration parameter DcmDspSecurityAttemptCounterEnabled is equal to TRUE and DcmDspSecurityUsePort is equal to either USE_ASYNCH_CLIENT_SERVER or USE_ASYNCH_FNC.

8.3.7.1.5 Xxx_SetSecurityAttemptCounter

Function Name	Xxx_SetSecurityAttemptCounter	
Syntax:	Std_ReturnType Xxx_SetSecurityAttemptCounter (Dcm_OpStatusType OpStatus, uint8 AttemptCounter)	
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	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDspSecurityAttemptCounterEnabled is equal to TRUE and	

	DcmDspSecurityUsePort is equal to either USE_ASYNCH_CLIENT_SERVER or USE_ASYNCH_FNC.

8.3.7.2 Synchronous Operations

8.3.7.2.1 Xxx_ GetSeed [SecurityAccessDataRecord on]

Function Name	Xxx_ GetSeed	
Syntax:	Std_ReturnType Xxx_GetSeed(uint8* SecurityAccessDataRecord, uint8* Seed, Dcm_NegativeResponseCodeType* ErrorCode)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	SecurityAccessDataRecord	Contain security access data record to be written.
	OpStatus	Status of the current operation
Parameters (Out)	ErrorCode	NRC to be sent in the negative response in case of failure
	Seed	Buffer where the requested seed value shall be copied to
Return Value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests the application to get seed value	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDspSecurityADRSize is configured and configuration parameter DcmDspSecurityUsePort is equal to either USE_SYNCH_CLIENT_SERVER and USE_SYNCH_FNC.	

8.3.7.2.2 Xxx_ GetSeed [SecurityAccessDataRecord off]

Function Name	Xxx_ GetSeed	
Syntax:	Std_ReturnType Xxx_GetSeed(uint8* Seed, Dcm_NegativeResponseCodeType* ErrorCode)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (Inout)	None	
Parameters (Out)	ErrorCode	NRC to be sent in the negative response in case of failure
	Seed	Buffer where the requested seed value shall be copied to
Return Value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This function requests the application to get seed value	
Preconditions	None	
Configuration Dependency	This API is available only if configuration parameter DcmDspSecurityADRSize is not configured and configuration parameter DcmDspSecurityUsePort is equal to either USE_SYNCH_CLIENT_SERVER and USE_SYNCH_FNC.	

8.3.7.2.3 Xxx_CompareKey

Function Name	Xxx_CompareKey	
Syntax:	Std_ReturnType Xxx_CompareKey(uint8* Key)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (Out)	Key	Key, which needs to be compared
Parameters (Inout)	None	
Return Value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.

Description	Request to application for comparing key
Preconditions	None
Configuration Dependency	This API is available only if configuration parameter DcmDspSecurityUsePort is equal to either USE_SYNCH_CLIENT_SERVER and USE_SYNCH_FNC.

8.3.8 ServiceRequestNotification

8.3.8.1 Xxx_Indication

Function Name	Xxx_Indication	
Syntax:	Std_ReturnType Xxx_Indication (uint8 SID, uint8* RequestData, uint16 DataSize, uint8 ReqType, uint16 SourceAddress, Dcm_NegativeResponseCodeType* ErrorCode)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	SID	Value of service identifier
	RequestData	This parameter contains the complete request data (diagnostic buffer), except the service ID.
	DataSize	This parameter defines how many bytes in the RequestData parameter are valid
	ReqType	Addressing type of the request(0=physical request 1=functional request)
	SourceAddress	Dcm client description
Parameters (Out)	ErrorCode	E_REQUEST_NOT_ACCEPTED, E_NOT_OK
Parameters (Inout)	None	
Return Value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not

		successful. E_REQUEST_NOT_ACCEPTED: Request not accepted
Description	Indication of the successful reception of a new request to application and it is called right before the DSD verification (SID, security access, diagnostic session). Within this function application can examine the permission of the diagnostic service / environment (e.g. ECU state afterrun).	
Preconditions	None	
Configuration Dependency	This API is available only if configuration container DcmDslServiceRequestSupplierNotification is configured.	

8.3.8.2 Xxx_Confirmation

Function Name	Xxx_Confirmation	
Syntax:	Std_ReturnType Xxx_Confirmation (uint8 SID, uint8 ReqType, uint16 SourceAddress, Dcm_ConfirmationStatusType ConfirmationStatus)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	SID	Value of service identifier
	ConfirmationStatus	Confirmation of a successful transmission or a transmission error of a diagnostic service.
	ReqType	Addressing type of the request(0=physical request 1=functional request)
	SourceAddress	Dcm client description
Parameters (Inout)	None	
Return Value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	Confirmation of the successful reception of a new request to	

	application.
Preconditions	None
Configuration Dependency	This API is available only if configuration container DcmDslServiceRequestSupplierNotification is configured.

8.3.9 CallbackDCMRequestServices

8.3.9.1 Xxx_StartProtocol

Function Name	Xxx_StartProtocol	
Syntax:	Std_ReturnType Xxx_StartProtocol (Dcm_ProtocolType ProtocolID)	
Sync/Async	NA	
Reentrancy	Non Reentrant	
Parameters (In)	ProtocolID	Name of the protocol(IDs configured within DCM_PROTOCOL_ID)
Parameters (Inout)	None	
Return Value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. E_PROTOCOL_NOT_ALLOWED: conditions in application allows no further procession of protocol
Description	Indication of protocol start. Application is able to reject further processing of requested protocol due to failed conditions.	
Preconditions	None	
Configuration Dependency	This API is available only if configuration container DcmDslCallbackDCMRequestService is configured.	

8.3.9.2 Xxx_StopProtocol

Function Name	Xxx_StopProtocol	
Syntax:	Std_ReturnType Xxx_StopProtocol (Dcm_ProtocolType ProtocolID)	
Sync/Async	NA	
Reentrancy	Non Reentrant	
Parameters (In)	ProtocolID	Name of the protocol(IDs configured within DCM_PROTOCOL_ID)
Parameters (Inout)	None	
Return Value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful. E_PROTOCOL_NOT_ALLOWED: conditions in application allows no further procession of protocol
Description	Indication of protocol stop. If a running diagnostic requested is preempted by a higher prior request (of another protocol, e.g. OBD), application is requested to abort further processing of running request ProtocolID: Name of the protocol(IDs configured within DCM_PROTOCOL_ID).	
Preconditions	None	
Configuration Dependency	This API is available only if configuration container DcmDslCallbackDCMRequestService is configured.	

8.3.10 InfotypeServices_{VehInfoData}

8.3.10.1 Xxx_RequestControl

This API is not supported yet

Function Name	Xxx_RequestControl
Syntax:	Std_ReturnType Xxx_RequestControl (uint8* OutBuffer, uint8* InBuffer)

Sync/Async	NA	
Reentrancy	Non Reentrant	
Parameters (In)	InBuffer	Provodes input buffer
Parameters (out)	OutBuffer	Provodes output buffer
Return Value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This interface allows the DCM to provide OBD Service $08.	
Preconditions	None	
Configuration Dependency	None	

8.3.10.2 Xxx_GetDTRValue

This API is not supported yet

Function Name	Xxx_GetDTRValue	
Syntax:	Std_ReturnType Xxx_GetDTRValue(Dcm_OpStatusType OpStatus, uint16* Testval, uint16* Minlimit, uint16* Maxlimit, uint8* Status)	
Sync/Async	NA	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	Status of the current operation
Parameters (out)	Testval	Returns the test value
	Minlimit	Returns the minimum value
	Maxlimit	Returns the maximum value
	Status	Returns the status of test
Return Value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This interface used to get DTR data from SW-C for service 6.	
Preconditions	None	

Configuration Dependency	None

8.3.10.3 Xxx_GetInfotypeValueData

This API is not supported yet

Function Name	Xxx_GetInfotypeValueData	
Syntax:	Std_ReturnType Xxx_GetInfotypeValueData (Dcm_OpStatusType OpStatus, uint8* DataValueBuffer)	
Sync/Async	NA	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	Status of the current operation
Parameters (out)	DataValueBuffer	Provides the value of requested infotype
Return Value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	This interface used to get Infotype data from SW-C.	
Preconditions	None	
Configuration Dependency	None	

8.3.11 CallbackDCMPresentDate

8.3.11.1 Xxx_GetPresentDate

Function Name	Xxx_GetPresentDate	
Syntax:	Std_ReturnType Xxx_GetPresentDate (OUT P2VAR(uint8, AUTOMATIC, RTE_APPL_DATA) Data)	
Sync/Async	NA	
Reentrancy	Non Reentrant	
Parameters (In)	None	
Parameters (out)	Data	Provides the value of present date.

Return Value	Std_ReturnType	E_OK: Request was successful. E_NOT_OK: Request was not successful.
Description	Application provides present date.	
Preconditions	None	
Configuration Dependency	This API is available only if configuration container DcmDspCallbackDCMPresentDate is configured.	

8.3.12 RoutineServices_{RoutineName}

8.3.12.1 Fixed length

8.3.12.1.1 Xxx_Start

Function Name	Xxx_Start	
Syntax:	Std_ReturnType Xxx_Start (〈datatype〉 dataIn1,…,uint8* dataInN, Dcm_OpStatusType OpStatus, 〈datatype〉 dataOut1,…,uint8* dataOutN, uint16* currentDataLength, Dcm_NegativeResponseCodeType* ErrorCode)	
Sync/Async	NA	
Reentrancy	Non Reentrant	
Parameters (In)	〈datatype〉 dataIn1	Provides the input data of type 〈datatype〉
	uint8* dataInN	Provides the input data of type
	OpStatus	Status of the current operation
Parameters (out)	〈datatype〉 dataOut1	output data of type 〈datatype〉
	uint8* dataOutN	Provide the buffer for dataout
	ErrorCode	E_NOT_OK, DCM_E_PENDING, E_FORCE_RCRRP
Parameters (Inout)	currentDataLength	Provides current data length
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. E_FORCE_RCRRP: application request the transmission of a response Pending (NRC 0x78)
Description	This interface used start the routine service.	
Preconditions	None	
Configuration Dependency	This API is available only If configuration parameter DcmDspRoutineFixedLength is set to FALSE	

8.3.12.1.2 Xxx_Stop

Function Name	Xxx_Stop	
Syntax:	Std_ReturnType Xxx_Stop (〈datatype〉 dataIn1,…,uint8* dataInN, Dcm_OpStatusType OpStatus, 〈datatype〉 dataOut1,…,uint8* dataOutN, uint16* currentDataLength, Dcm_NegativeResponseCodeType* ErrorCode)	
Sync/Async	NA	
Reentrancy	Non Reentrant	
Parameters (In)	〈datatype〉 dataIn1	Provides the input data of type 〈datatype〉
	uint8* dataInN	Provides the input data
	OpStatus	Status of the current operation
Parameters (Inout)	currentDataLength	Provides the current data length
Parameters (out)	〈datatype〉 dataOut1	Provides the output data of type 〈datatype〉
	uint8* dataOutN	Provides the buffer for dataout

	ErrorCode	E_NOT_OK, DCM_E_PENDING, E_FORCE_RCRRP
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. E_FORCE_RCRRP: application request the transmission of a response Pending (NRC 0x78)
Description	This interface used stop the routine service.	
Preconditions	None	
Configuration Dependency	This API is available only If configuration parameter DcmDspRoutineFixedLength is set to FALSE	

8.3.12.1.3 Xxx_RequestResults

Function Name	Xxx_RequestResults	
Syntax:	Std_ReturnType Xxx_RequestResults (Dcm_OpStatusType OpStatus, 〈datatype〉 dataOut1,…,uint8* dataOutN, uint16* currentDataLength, Dcm_NegativeResponseCodeType* ErrorCode)	
Sync/Async	NA	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	Status of the current operation
Parameters (out)	〈datatype〉 dataOut1	Provides the output data of type 〈datatype〉
	currentDataLength	Provides the current data length

	ErrorCode	E_NOT_OK, DCM_E_PENDING, E_FORCE_RCRRP
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. E_FORCE_RCRRP: application request the transmission of a response Pending (NRC 0x78)
Description	This interface used request the result of routine service.	
Preconditions	None	
Configuration Dependency	This API is available only If configuration parameter DcmDspRoutineFixedLength is set to FALSE	

8.3.12.2 Variable lenth

8.3.12.2.1 Xxx_Start

Function Name	Xxx_Start	
Syntax:	Std_ReturnType Xxx_Start(〈datatype〉 dataIn1,…,〈datatype〉 dataInN, Dcm_OpStatusType OpStatus, 〈datatype〉* dataOut1,…, 〈datatype〉* dataOutN, Dcm_NegativeResponseCodeType* ErrorCode)	
Sync/Async	NA	
Reentrancy	Non Reentrant	
Parameters (In)	〈datatype〉 dataIn1	Provides the input data of type 〈datatype〉
	uint8* dataInN	Provides the input data

	OpStatus	Status of the current operation
Parameters (out)	〈datatype〉 dataOut1	onput data of type 〈datatype〉
	uint8* dataOutN	Provides buffer for dataout
	ErrorCode	E_NOT_OK, DCM_E_PENDING, E_FORCE_RCRRP
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. E_FORCE_RCRRP: application request the transmission of a response Pending (NRC 0x78)
Description	This interface used start the routine service.	
Preconditions	None	
Configuration Dependency	This API is available only If configuration parameter DcmDspRoutineFixedLength is set to TRUE.	

8.3.12.2.2 Xxx_Stop

Function Name	Xxx_Stop	
Syntax:	Std_ReturnType Xxx_Stop (〈datatype〉 dataIn1,…,uint8* dataInN, Dcm_OpStatusType OpStatus, 〈datatype〉 dataOut1,…,uint8* dataOutN, Dcm_NegativeResponseCodeType* ErrorCode)	
Sync/Async	NA	
Reentrancy	Non Reentrant	
Parameters (In)	〈datatype〉 dataIn1	Provides the input data of type 〈datatype〉
	uint8* dataInN	Provides the input data

	OpStatus	Status of the current operation
Parameters (out)	〈datatype〉 dataOut1	Provides the output data of type 〈datatype〉
	uint8* dataOutN	Provides the buffer for dataout
	ErrorCode	E_NOT_OK, DCM_E_PENDING, E_FORCE_RCRRP
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. E_FORCE_RCRRP: application request the transmission of a response Pending (NRC 0x78)
Description	This interface used stop the routine service.	
Preconditions	None	
Configuration Dependency	This API is available only If configuration parameter DcmDspRoutineFixedLength is set to TRUE.	

8.3.12.2.3 Xxx_RequestResults

Function Name	Xxx_RequestResults	
Syntax:	Std_ReturnType Xxx_RequestResults (Dcm_OpStatusType OpStatus, 〈datatype〉 dataOut1,…,uint8* dataOutN, Dcm_NegativeResponseCodeType* ErrorCode)	
Sync/Async	NA	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	Status of the current operation
Parameters (out)	〈datatype〉 dataOut1	Provides the input data of type 〈datatype〉

	ErrorCode	E_NOT_OK, DCM_E_PENDING, E_FORCE_RCRRP
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. E_FORCE_RCRRP: application request the transmission of a response Pending (NRC 0x78)
Description	This interface used request the result of routine service.	
Preconditions	None	
Configuration Dependency	This API is available only If configuration parameter DcmDspRoutineFixedLength is set to TRUE	

8.3.13 External Diagnostic Service Processing

8.3.13.1 Dcm_ExternalSetNegResponse

Function Name	Dcm_ExternalSetNegResponse	
Syntax:	FUNC(void, DCM_CODE) Dcm_ExternalSetNegResponse (P2VAR(Dcm_MsgContextType, AUTOMATIC, DCM_APPL_DATA) pMsgContext, Dcm_NegativeResponseCodeType ErrorCode)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	pMsgContext	Message-related information for one diagnostic protocol identifier
	ErrorCode	NRC to be sent in the negative response in case of failure.

Parameters (Inout)	None
Parameters (Out)	None
Return Value	None
Description	Used by service interpreter outside of DCM to indicate that a the final response shall be a negative one. Dcm_ExternalSetNegResponse will not finalize the response processing.
Preconditions	Dcm_Init should be called before calling this API.
Configuration Dependency	None

8.3.13.2 Dcm_ExternalProcessingDone

Function Name	Dcm_ExternalProcessingDone	
Syntax:	FUNC(void, DCM_CODE) Dcm_ExternalProcessingDone (P2VAR(Dcm_MsgContextType, AUTOMATIC, DCM_APPL_DATA) pMsgContext)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	pMsgContext	Message-related information for one diagnostic protocol identifier
Parameters (Inout)	None	
Parameters (Out)	None	
Return Value	None	
Description	Used by service interpreter outside of DCM to indicate that a final response can be sent.	
Preconditions	Dcm_Init should be called before calling this API.	
Configuration Dependency	None	

8.3.13.3 〈Module〉_〈DiagnosticService〉

Function Name	〈Module〉_〈DiagnosticService〉

Syntax:	Std_ReturnType <Module>_<DiagnosticService>(Dcm_OpStatusTypeOpStatus, constDcm_MsgContextType*pMsgContext)	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (In)	OpStatus	DCM_INITIAL: Indicates the initial call to the operation DCM_PENDING : Indicates that a pending return has been done on the previous call of the operation DCM_CANCEL: Indicates that the DCM requests to cancel the pending operation
	pMsgContext	Message-related information for one diagnostic protocol identifier The pointers in pMsgContext shall point behind the SID
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
Description	Callout function. 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 at it is reentrant. Caller is responsible for the lifetime of the argument pMsgContext.	

	The name of the callout is defined within parameter DcmDsdSidTabFnc
Preconditions	None
Configuration Dependency	None

8.3.13.4 〈Module〉_〈DiagnosticService〉_〈SubService〉

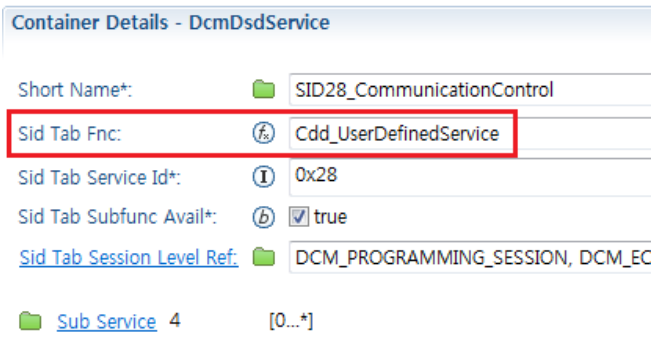
Function Name	〈Module〉_〈DiagnosticService〉_〈SubService〉	
Syntax:	Std_ReturnType〈Module〉_〈DiagnosticService〉_〈SubService〉(Dcm_OpStatusTypeOpStatus, constDcm_MsgContextType*pMsgContext)	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (In)	OpStatus	DCM_INITIAL: Indicates the initial call to the operation DCM_PENDING : Indicates that a pending return has been done on the previous call of the operation DCM_CANCEL: Indicates that the DCM requests to cancel the pending operation
	pMsgContext	Message-related information for one diagnostic protocol identifier The pointer in pMsgContext shall point behind the SubFunction
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
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.	
Preconditions	None	
Configuration Dependency	None	

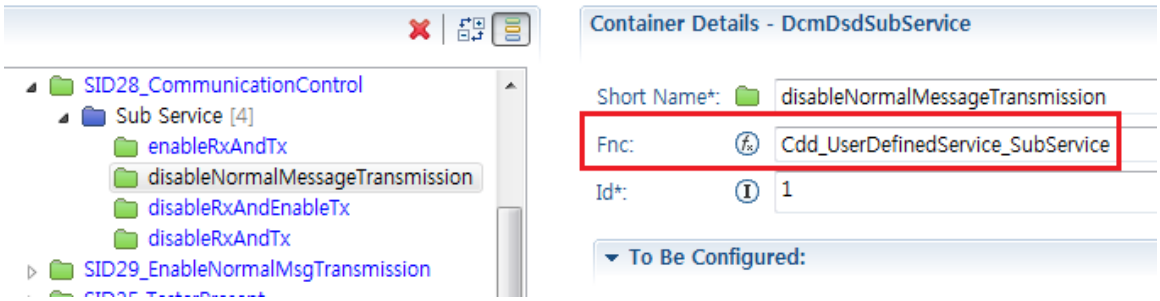
8.3.14 User defined Service Functions

User-defined services and subservices can be used instead of platform-provided services. To use the user-defined service function, the followings should be configured.

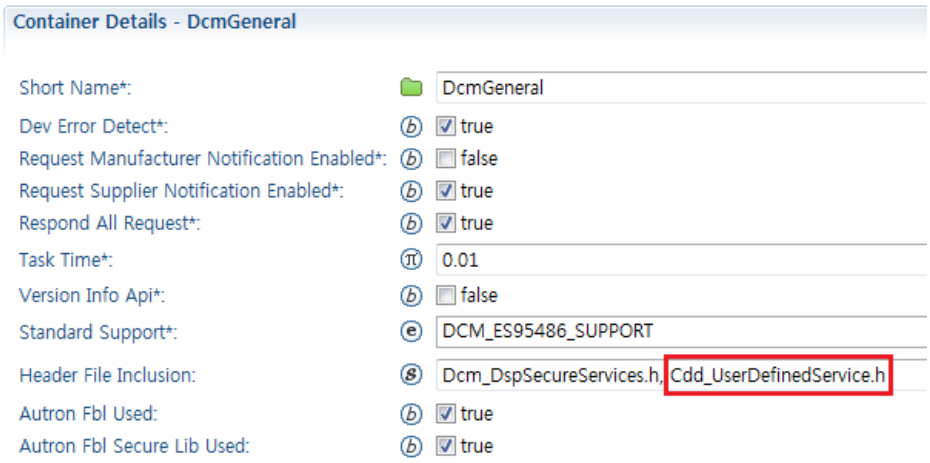
1-1. To use user-defined service functions Register symbol into DcmDsdService/DcmDsdSidTabFnc



1-2. To use user-defined subservice functions Register symbol into DcmDsdSubService/DcmDsdSubServiceFnc



2. Add the header file declaring the user-defined function to DcmGeneral/Header File Inclusion



The followings are examples of user-defined service function by use case.

Note: The following sample code is only for reference and should not be simply applied to a project. Logic must be configured in a line that does not violate diagnostic specifications.

Note: Calling an AutoEver internal function from a user-defined function is violation of the specifications. We are not responsible for any problems caused by implementing the code in such a way.

Note: When implementing User Defined Subservice Function, pay attention to the time of invocation (User Defined Subservice Function is called from AutoEver internal function), and other implementation methods are the same as for User Defined Service Function.

```
FUNC(Std_ReturnType, DCM_CODE) Cdd_UserDefinedService (
  Dcm_OpStatusType OpStatus,
  P2VAR(Dcm_MsgContextType, AUTOMATIC, DCM_APPL_DATA) pMsgContext)
```

```
{
  /* return value is used after initialization to E_NOT_OK. */
  Std_ReturnType retVal = E_NOT_OK;
  Dcm_NegativeResponseCodeType ErrorCode = DCM_E_POSITIVERESPONSE;

  /* Branching is processed according to the Input Parameter OpStatus.
  OpStatus == DCM_INITIAL : Fist entry of function
  OpStatus == DCM_PENDING : Reentry after PENDING
  OpStatus == DCM_CANCEL : End of service */
  switch(OpStatus)
  {
  case DCM_INITIAL:
    /* Do something */

    Use /* pMsgContext:
    By using pMsgContext structure, users can identify Request Message and implement the Response Message format.
    For more information, see Dcm_MsgContextType of Type Definitions chapter.
    The followings are example of usage. */

    /* Request Length Check */
    if(pMsgContext->reqDataLen != 0x01)
    {
      /* Check Subfunction */
      if(pMsgContext->reqData[0] == 0x01)
      {
        if(Not Ready)
        {
          /* Pending processing: A logic is needed to complete service in OpStatus == DCM_PENDING condition */
          retVal = DCM_E_PENDING;
        }
        else
        {
```

```
        /* Subfunction 0x01 */
        pMsgContext->resDataLen = 2U;
        pMsgContext->resData[0] = 0x01; /* Subfunction Id */
        pMsgContext->resData[1] = 0x11; /* User Response Data */
        retVal = E_OK;
      }
    }
    else if(pMsgContext->reqData[0] == 0x02)
    {
      if(Not Ready)
      {
        /* Pending processing */
        retVal = DCM_E_PENDING;
      }
      else
      {
        /* Subfunction 0x02 */
        pMsgContext->resDataLen = 2U;
        pMsgContext->resData[0] = 0x02; /* Subfunction Id */
        pMsgContext->resData[1] = 0x22; /* User Response Data */
        retVal = E_OK;
      }
    }
    else
    {
      /* Not Supported Subfunction Error : DCM_E_SUBFUNCTIONNOTSUPPORTED (NRC12) */
      ErrorCode = DCM_E_SUBFUNCTIONNOTSUPPORTED;
      retVal = E_NOT_OK;
    }
  }
  else
  {
    /* Request Length Error : DCM_E_INCORRECTMESSAGELENGTHORINVALIDFORMAT (NRC13) */
    ErrorCode = DCM_E_INCORRECTMESSAGELENGTHORINVALIDFORMAT;
```

```
      retVal = E_NOT_OK;
    }

    break;

  case DCM_PENDING:
    /* Do something */

    /* Check Subfunction */
    if(pMsgContext->reqData[0] == 0x01)
    {
      if(Not Ready)
      {
        /* Pending processing: A logic is needed to complete service in OpStatus == DCM_PENDING condition
*/
        retVal = DCM_E_PENDING;
      }
      else
      {
        /* Subfunction 0x01 */
        pMsgContext->resDataLen = 2U;
        pMsgContext->resData[0] = 0x01; /* Subfunction Id */
        pMsgContext->resData[1] = 0x11; /* User Response Data */
        retVal = E_OK;
      }
    }
    else if(pMsgContext->reqData[0] == 0x02)
    {
      if(Not Ready)
      {
        /* Pending processing */
        retVal = DCM_E_PENDING;
      }
      else
```

```
    {
      /* Subfunction 0x02 */
      pMsgContext->resDataLen = 2U;
      pMsgContext->resData[0] = 0x02; /* Subfunction Id */
      pMsgContext->resData[1] = 0x22; /* User Response Data */
      retVal = E_OK;
    }
  }
  else
  {
    /* Not Supported Subfunction Error : DCM_E_SUBFUNCTIONNOTSUPPORTED (NRC12) */
    ErrorCode = DCM_E_SUBFUNCTIONNOTSUPPORTED;
    retVal = E_NOT_OK;
  }

  break;
case DCM_CANCEL:
  /* Do something */

  break;

default:

  break;
}

/* Processing of ReturnValue and Response */
if(retVal == E_NOT_OK)
{
  /* Negative Response */
  Dcm_ExternalSetNegResponse(pMsgContext, ErrorCode);
  Dcm_ExternalProcessingDone(pMsgContext);
}
else if(retVal == E_OK)
```

```
  {
    /* Positive Response */
    Dcm_ExternalProcessingDone(pMsgContext);
  }
  else
  {
    /* Pending Response */
  }

  return retVal;
}
```

Dcm_Authentication_User_CallOut

Function Name	Dcm_Authentication_User_CallOut	
Syntax:	FUNC(void, DCM_CALL_OUT_CODE) Dcm_Authentication_User_CallOut (Dcm_OpStatusType OpStatus, P2VAR(Dcm_MsgContextType, AUTOMATIC, DCM_APPL_DATA) pMsgContext), P2VAR(Dcm_NegativeResponseCodeType, AUTOMATIC, DCM_APPL_DATA) pErrorCode))	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	OpStatus	Operation status
Parameters (Inout)	pMsgContext	Message-related information for one diagnostic protocol identifier
	pErrorCode	Negative Error code
Parameters (Out)	None	
Return Value	None	
Description	Used user for update other value for NRC or AuthenticationReturnParameter.	
Preconditions	Dcm_Init should be called before calling this API. Authentication service used.	
Configuration Dependency	None	

8.3.15 Notes

8.3.15.1 In Communication with application SW-C

For information on prototype of functions created based on RTE, see AUTOSAR BSW Service API Guide.doc.

8.3.16 RequestFileTransfer Callout

Note Refer to Dcm_Callouts.c

8.3.16.1 Dcm_ProcessRequestAddFile

Function Name	Dcm_ProcessRequestAddFile	
Syntax:	Std_ReturnType Dcm_ProcessRequestAddFile(Dcm_OpStatusType OpStatus, uint16 filePathAndNameLength, const uint8* filePathAndName, uint8 dataFormatIdentifier, uint64 fileSizeUncompressed, uint64 fileSizeCompressed, uint64* maxNumberOfBlockLength, Dcm_NegativeResponseCodeType* ErrorCode)	
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.
	filePathAndName Length	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.
	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)	maxNumberOfBlockLength	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 modeOfOperation equal to 0x01 (AddFile).	
Preconditions	DCM module must be initialised	

Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 38 and DCM_REQUEST_FILE_TRANSFER_SERVICE are STD_ON.

8.3.16.2 Dcm_ProcessRequestDeleteFile

Function Name	Dcm_ProcessRequestDeleteFile	
Syntax:	Std_ReturnType Dcm_ProcessRequestDeleteFile (Dcm_OpStatusType OpStatus, uint16 filePathAndNameLength, const uint8* filePathAndName, Dcm_NegativeResponseCodeType* ErrorCode)	
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.
	filePathAndName Length	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_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 modeOfOperation equal to 0x02 (DeleteFile).	
Preconditions	DCM module must be initialised	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 38 and DCM_REQUEST_FILE_TRANSFER_SERVICE are STD_ON.	

8.3.16.3 Dcm_ProcessRequestReplaceFile

Function Name	Dcm_ProcessRequestReplaceFile	
Syntax:	Std_ReturnType Dcm_ ProcessRequestReplaceFile (Dcm_OpStatusType OpStatus, uint16 filePathAndNameLength, const uint8* filePathAndName, uint8 dataFormatIdentifier, uint64 fileSizeUncompressed, uint64 fileSizeCompressed, uint64* maxNumberOfBlockLength, Dcm_NegativeResponseCodeType* ErrorCode)	
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.
	filePathAndName Length	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.
	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)	maxNumberOfBlockLength	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 modeOfOperation equal to 0x03 (ReplaceFile).	
Preconditions	DCM module must be initialised	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 38 and DCM_REQUEST_FILE_TRANSFER_SERVICE are STD_ON.	

8.3.16.4 Dcm_ProcessRequestReadFile

Function Name	Dcm_ProcessRequestReadFile
Syntax:	Std_ReturnType Dcm_ProcessRequestReadFile (Dcm_OpStatusType OpStatus, uint16 filePathAndNameLength, const uint8* filePathAndName, uint8 dataFormatIdentifier, uint64* fileSizeUncompressed, uint64* fileSizeCompressed, uint64* maxNumberOfBlockLength, Dcm_NegativeResponseCodeType* ErrorCode)

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.
	filePathAndName Length	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)	maxNumberOfBlockLength	Max number of bytes to be

		included in each TransferData request 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.
	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.
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 modeOfOperation equal to 0x04 (ReadFile).	
Preconditions	DCM module must be initialised	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 38 and DCM_REQUEST_FILE_TRANSFER_SERVICE are STD_ON.	

8.3.16.5 Dcm_ProcessRequestReadDir

Function Name	Dcm_ProcessRequestReadDir
Syntax:	Std_ReturnType Dcm_ProcessRequestReadDir (Dcm_OpStatusType OpStatus, uint16 filePathAndNameLength, const uint8* filePathAndName,

	uint64* dirInfoLength, uint64* maxNumberOfBlockLength, Dcm_NegativeResponseCodeType* ErrorCode)	
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.
	filePathAndName Length	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 modeOfOperation equal to 0x05 (ReadDir).	
Preconditions	DCM module must be initialised	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 38 and DCM_REQUEST_FILE_TRANSFER_SERVICE are STD_ON.	

8.3.16.6 Dcm_WriteFile

Function Name	Dcm_WriteFile	
Syntax:	Std_ReturnType Dcm_WriteFile (Dcm_OpStatusType OpStatus, uint64 DataLength, uint8* Data, Dcm_NegativeResponseCodeType* ErrorCode)	
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 maxNumberOfBlockLength return in Dcm_ProcessRequestFileTransfer.
	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 TransferData if there's an ongoing RequestFileTransfer process started with 0x01 (AddFile) or 0x03 (ReplaceFile).	
Preconditions	DCM module must be initialised	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 38 and DCM_REQUEST_FILE_TRANSFER_SERVICE are STD_ON.	

8.3.16.7 Dcm_ReadFileOrDir

Function Name	Dcm_ReadFileOrDir
Syntax:	Std_ReturnType Dcm_ReadFileOrDir (Dcm_OpStatusType OpStatus, uint64 DataLength, uint8* Data, Dcm_NegativeResponseCodeType* ErrorCode)
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	As in, the parameter defines the maximum block length to be used, i.e. the value of maxNumberOf-BlockLength 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.
	Data	Pointer to the data to be written.
Parameters (Inout)	None	
Parameters (Out)	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 is received using UDS service TransferData if there's an ongoing RequestFileTransfer process started with 0x04 (ReadFile) or 0x05 (ReadDir).	
Preconditions	DCM module must be initialised	
Configuration Dependency	This API is available only if configuration parameter DcmDsdSidTabServiceId is configured as 38 and DCM_REQUEST_FILE_TRANSFER_SERVICE are STD_ON.	

9 Generator

9.1 Generator Option

Option	Description
-S	To create software component description
-bend	To use big endian (default little endian)
-H/-Help	To display help regarding usage of the tool.
-O/-Output	To generate the output files in the specified directory location.
-V/-Version	To display the copyright information and the tool version.
-L/-Log	To generate \"$BswConfig::Lis_File_Name\" file.
-D/-DryRun	To execute in validation mode.
-I/-Info	To disable Information Messages.
-W/-Warn	To disable Warning Messages.
-DDT	To disable the generation of Date and Time Information in the Tool Generated Output Files.

9.2 Generator Error Message

This section helps to analyze the errors or warnings displayed during the execution of the tool. It ensures conformance of input file(s) with syntax and semantics.

The Generation Tool displays errors or warnings or information when the user has configured incorrect inputs. The format of Error/Warning/Information message is as shown below:

- ERR/WRN/INF<mid><xxx>: < Error/Warning/Information Message>

Where,

<mid>: 053 – Dcm Module Id (53) for user configuration checks.

000 – for command line checks.

<xxx>: 001 – 999 – Message ID.

- File Name : Name of the file in which the error has occurred
- Path : Absolute path of the container in which the parameter is present

'File Name' and 'Path' are optional.

Below section provides the list of module specific error, warning and information messages.

9.2.1 Error Messages

ERR053001: Unexpected Error Found. Please contact AUTOEVER AUTOSAR Support System.

This is an Unexpected Error. On the occurrence of this error contact AUTOEVER AUTOSAR Support System.

ERR053002: Unexpected Error Found. This error may be due to the incorrect configuration of the element(s) <Parameter Name/ Container Name>. If the error is not resolved, then please contact AUTOEVER AUTOSAR Support System.

This error occurs, if the structure fields that are to be generated in the C Source file are empty. Contact AUTOEVER AUTOSAR Support System.

ERR053003: 'Component Name' Component is not present in the input file(s).

This error occurs, if any of the component DCM PDUR DEM NVM and COMM (are) not present in any of the input ECU Configuration Description File(s).

ERR053004: The reference path is empty for the parameter 'parameter name' in the container 'container name', having short name 'short name'.

This error occurs, if no reference path is provided for any of the below mentioned parameters:

Container Name	Parameter Name
DcmDcmDslPeriodicTransmission	DcmPeriodicTransmissionConRef
DcmDcmDslResponseOnEvent	DcmROEConnectionRef
DcmDslBuffer	DcmProtocolRxBufferId
	DcmProtocolTxBufferId
	DcmPeriodicTxBufferRef
DcmDsdServiceIdTable	DcmProtocolSIDTable
DcmDspSession	DcmSessionRef
DcmDspDid	DcmDidRef
DcmDspDidInfo	DcmDidInfoRef
DcmData	DcmDataRef
DcmDataInfo	DcmDataInfoRef
DcmNvMBlockDescriptor	DcmDataBlockIdRef
DcmDemPidDataElement	DcmPidDataDemRef
DcmDspRoutineAuthorization	DcmRoutineSessionRef
	DcmRoutineModeRuleRef
	DcmReadSessionRef
	DcmControlSessionRef
	DcmSubServiceSessionRef
	DcmSidTabSessionLevelRef
DspTestResultTid	DcmTestResultObdmidTidRef
DcmDspSecurityRow	DcmReadMemoryRangeSecurityRef

Container Name	Parameter Name
	DcmWriteMemoryRangeSecurityRef
	DcmRoutineSecurityLevelRef
	DcmSourceDidSecurityLevelRef
	DcmReadSecurityLevelRef
	DcmControlSecurityLevelRef
	DcmSubServiceSecurityRef
	DcmSidTabSecLevelRef
DcmDspRoutineInfo	DcmRoutineInfoRef
DcmModeRule	DcmComReEnabledModeRuleRef
	DcmControlDTCReEnableModeRuleRef
	DcmWriteModeRuleRef
	DcmReadModeRuleRef
	DcmControlModeRuleRef
	DcmSubServiceModeRuleRef
	DcmSidTabModeRuleRef
DcmModeRule|ModeCondition	DcmArgumentRef

ERR053005: The parameter 'Parameter Name' in the container 'Container Name' should be configured.

This error occurs, if the parameter 'Parameter Name' in the container 'Container Name' is not configured.

Container Name	Parameter Name
DcmGeneral	DcmDevErrorDetect

Container Name	Parameter Name
	DcmVersionInfoApi
	DcmRespondAllRequest
	DcmManufacturerEnabled
	DcmSupplierEnabled
	DcmTaskTime
DcmDsdServiceIdTable	DcmSidTabId
DcmDsdService	DcmSidTabServiceId
	DcmSidTabSubfuncAvail
DcmDslBuffer	DcmBufferSize
DcmDslDiagResp	DcmMaxNumRespPend
DcmDslProtocolRow	DcmProtocolId
	DcmProtocolIsParallelExecutab
	DcmProtocolPreemptTimeout
	DcmProtocolPriority
	DcmProtocoltransType
DcmDslProtocolRx	DcmProtocolRxAddrType
	DcmRxChannelId
	DcmRxTesterSource
	DcmProtocolRxPduId
DcmDslProtocolTx	DcmConfirmationPduId
DcmData	DcmDataSize

Container Name	Parameter Name
	DcmDataUsePort
DcmDataInfo	DcmDataFixedLength
DcmDspDid	DcmDidIdentifier
	DcmDidUsed
DcmDidInfo	DcmDynamicallyDefined
DcmReadMemoryRangeInfo	DcmReadMemoryRangeHigh
	DcmReadMemoryRangeLow
DcmWriteMemoryRangeInfo	DcmWriteMemoryRangeHigh
	DcmWriteMemoryRangeLow
DcmDspRoe	DcmInitOnDSC
	DcmInterMessageTime
	DcmRoeMaxNumberOfRetry
DcmDspRoeQueue	DcmMaxEventLength
	DcmRoeQueueEnab
	DcmMaxQueueLength
DcmDspRequestControl	DcmRequestControlInBufferSize
	DcmDspRequestControlOutBufferSize
	DcmRequestControlTestId
DcmDspRoutine	DcmRoutineIdentifier
	DcmRoutineUsePort
	DcmRoutineFixedLength

Container Name	Parameter Name
	DcmRoutineUsed
	DcmRequestResultsRoutineSupported
	DcmStopRoutineSupported
DcmRoutineStopInSignal	DcmRoutineSignalLength
	DcmRoutineSignalPos
	DcmRoutineSignalType
DcmRoutineStopOutSignal	DcmRoutineSignalLength
	DcmRoutineSignalPos
	DcmRoutineSignalType
DcmStartRoutineInSignal	DcmRoutineSignalLength
	DcmRoutineSignalPos
	DcmRoutineSignalType
DcmStartRoutineOutSignal	DcmRoutineSignalLength
	DcmRoutineSignalPos
	DcmRoutineSignalType
DcmRoutineRequestResOutSignal	DcmRoutineSignalLength
	DcmRoutineSignalPos
	DcmRoutineSignalType
DcmDspPid	DcmPidIdentifier
	DcmPidSize
	DcmPidService

Container Name	Parameter Name
	DcmPidUsed
DcmDspPidData	DcmPidDataPos
	DcmPidDataSize
DcmPidService	DcmPidDataUsePort
DcmPidSupportInfo	DcmPidSupportInfoLen
	DcmPidSupportInfoPos
DcmDspSecurity	DcmDspSecurityMaxAttemptCounterReadoutTime
DcmDspSecurityRow	DcmSecurityDelayTime
	DcmSecurityDelayTimeOnBoot
	DcmSecurityKeySize
	DcmSecurityLevel
	DcmSecurityNumAttDelay
	DcmSecuritySeedSize
	DcmDspSecurityAttemptCounterEnabled
DcmDspSessionRow	DcmSessionP2ServerMax
	DcmSessionP2StarServerMax
	DcmSessionBoot
DcmDspTestResultObdMidTid	DcmTestResultObdmid
DcmDspTestResultObdMidTids	DcmTestResultObdmidTidUaSid
DcmDspTestResultTid	DcmTestResultTestId
DcmDspVehInfo	DcmVehInfoInfoType

Container Name	Parameter Name
DcmDspVehInfoData	VehInfoDataOrder
	DcmVehInfoDataSize
	DcmVehInfoDataUsePort
DcmPageBufferCfg	DcmPagedBufferEnabled

ERR053006: The value configured for the parameter 'Parameter Name' in the container 'Container Name' should follow the pattern: <Pattern>.

This error occurs, when the parameter 'Parameter Name' is not configured as per the pattern.

Parameter Name	Container Name	Pattern	Example
ArReleaseVersion	BSW-IMPLEMENTATION	4. of [0-9]+. of [0-9]+	4.0.3
SwVersion		1. of [0-9]+. of [0-9]+	1.0.0
DcmDataConditionCheckreadFnc	DcmData	[a-zA-Z][a-zA-Z0-9₩_]*	Adc
DcmDataEcuSignal			
DcmDataFreezeCurrentStateFnc			
DcmDataGetScalingInfoFnc			
DcmDataReadDataLengthFnc			
DcmDataReadEcuSignal			
DcmDataReadFnc			
DcmDataResetToDefaultFnc			
DcmDataReturnControlToEcuFnc			
DcmDataShortTermAdjustmentFnc			
DcmDataWriteFnc			
DcmDataWriteFnc	DcmDspRequestControl	[a-zA-Z][a-zA-Z0-9₩_]*	Adc
DcmRequestResultsRoutineFnc	DcmDspRoutine	[a-zA-Z][a-zA-Z0-	Adc

Parameter Name	Container Name	Pattern	Example
DcmStartRoutineFnc		9₩_]*	
DcmStopRoutineFnc			
DcmPidDataReadFnc	DcmPidService01	[a-zA-Z][a-zA-Z0-9₩_]*	Adc
DcmGetSeedFnc	DcmDspSecurityRow	[a-zA-Z][a-zA-Z0-9₩_]*	Adc

ERR053013: The reference path ‹value› provided for the parameter 'Parameter' in the container 'Container Name', having short name 'short name' is incorrect.

This error occurs, if reference provided for any of the below mentioned parameters is incorrect:

Container Name	Parameter Name
DcmDcmDslPeriodicTransmission	DcmPeriodicTransmissionConRef
DcmDcmDslResponseOnEvent	DcmROEConnectionRef
DcmDslBuffer	DcmProtocolRxBufferId
	DcmProtocolTxBufferId
	DcmPeriodicTxBufferRef
DcmDsdServiceIdTable	DcmProtocolSIDTable
DcmDspSession	DcmSessionRef
DcmDspDid	DcmDidRef
DcmDspDidInfo	DcmDidInfoRef
DcmData	DcmDataRef
DcmDataInfo	DcmDataInfoRef
DcmNvMBlockDescriptor	DcmDataBlockIdRef

Container Name	Parameter Name
DcmDemPidDataElement	DcmPidDataDemRef
DcmDspSessionRow	DcmRoutineSessionRef
	DcmRoutineModeRuleRef
	DcmReadSessionRef
	DcmControlSessionRef
	DcmSubServiceSessionRef
	DcmSidTabSessionLevelRef
DspTestResultTid	DcmTestResultObdmidTidRef
DcmDspSecurityRow	DcmReadMemoryRangeSecurityRef
	DcmWriteMemoryRangeSecurityRef
	DcmRoutineSecurityLevelRef
	DcmSourceDidSecurityLevelRef
	DcmReadSecurityLevelRef
	DcmControlSecurityLevelRef
	DcmSubServiceSecurityRef
	DcmSidTabSecLevelRef
DcmDspRoutineInfo	DcmRoutineInfoRef
DcmModeRule	DcmComReEnabledModeRuleRef
	DcmControlDTCReEnableModeRuleRef
	DcmWriteModeRuleRef
	DcmReadModeRuleRef

Container Name	Parameter Name
	DcmControlModeRuleRef
	DcmSubServiceModeRuleRef
	DcmSidTabModeRuleRef
DcmDslProtocolRx	DcmDslProtocolRxComMChannelRef
DcmDspComControlSpecificChannel	DcmDspSpecificComMChannelRef
DcmModeRuleModeCondition	DcmArgumentRef

ERR053017: Value of the parameter 'DcmDspMaxPeriodicScheduler' in the container ' DcmDsp' should be greater than the value of the parameter 'DcmDspMaxPeriodicDidToRead' in the container 'DcmDsp'.

This error occurs, if value of the parameter DcmDspMaxPeriodicScheduler is less than DcmDspMaxPeriodicDidToRead

ERR053022: Value of the parameter 'DcmTaskTime' in the container 'DcmGeneral' should not be configured as <0>.

This error occurs, if value of the parameter DcmTaskTime == 0

ERR053051: The reference parameter 'DcmDslPeriodicTxPduRef ' should have a corresponding match in PduR module.

This error occurs, if value of the parameter 'DcmDslPeriodicTxPduRef' is not having a corresponding match in PduR module.

ERR053052: As value of parameter 'DcmDsdSidTabServiceId' is configured as <16 or 17 or 39 or 62 or 133 or 44 or 49 or 25 or 134>, then value of the parameter 'DcmDsdModeCondition' should be configured as <true/1> in the container ' DcmDsdService'.

This error occurs, if, the value of parameter 'DcmDsdSidTabServiceId' is configured as <16 or 17 or 39 or 62 or 133 or 44 or 49 or 25 or 134>, and the value of the parameter 'DcmDsdSidTabSubfuncAvail' is not configured as <true/1> in the container 'DcmDsdService'

ERR053053: The value of parameters 'DcmBswModeRef' and 'DcmSwcModeRef' should not be configured at a time in the container 'DcmDsdModeCondition'.

This error occurs, if the value of parameters 'DcmBswModeRef' and 'DcmSwcModeRef' is configured at a time in the container 'DcmDsdModeCondition'

ERR053054: The value of parameter 'Parameter Name' should be unique for each configured 'Container Name' container.

This error occurs, if the value of parameter 'Parameter Name' is not unique for each configured 'Container Name' container

Container Name	Parameter Name
DcmModeRule	DcmArgumentRef
DcmDspSecurityRow	DcmDspSecurityLevel
DcmDsdModeCondition	DcmBswModeRef
DcmDspSessionRow	DcmDspSessionLevel
DcmDsdServiceIdTable	DcmDsdSidTabId
DcmDsdServiceIdTable	DcmDsdSubService
DcmDslProtocolRow	DcmDslProtocolID
DcmDslPeriodicConnection	DcmDslPeriodicTxPduRef
DcmDslProtocolTx	DcmDslProtocolTxPduRef
DcmDspDid	DcmDspDidIdentifier
DcmDspPid	DcmDspPidIdentifier

ERR053055: The value of parameter 'DcmLogicalOperator' should be configured in the container 'DcmModeRule' when argumentref is configured more than one.

This error occurs, if Value of the parameter ‘DcmLogicalOperator’ in the container ‘DcmModeRule’ is not configured, if the parameter ‘DcmArgumentRef’ is more than 〈1〉)

ERR053056: As ServiceId is configured as 〈9〉 and ' DcmDspVehInfoInfoType' is configured as 〈8 or 11〉, the function name configured for parameter 'DcmDspVehInfoDataReadFnc' in container 'DcmDspVehInfo' should point to correct function provided by DEM and parameter 'DcmDspVehInfoUsePort' should be configured as 〈false〉.

This error occurs, if the value of parameter 'DcmDspVehInfoDataReadFnc' is configured as ‘Dem API(s)’ and the value of the parameter ‘DcmDspVehInfoUsePort’ is configured as 〈false/0〉, if the value of the parameter ‘DcmDspVehInfoInfoType’ is configured as 〈8 or 11〉

ERR053057: If the Value of the parameter 'DcmDslProtocolTransType' is Configured as 〈TYPE2〉 and also the value for the parameter ‘DcmDslProtocolIsParallelExecutab' is configured as 〈true/1〉, then atleast one periodic connection needs to be configured.

This error occurs, if the value of parameter 'DcmDslProtocolIsParallelExecutab' is configured as 〈false/0〉 and the value of the parameter ‘DcmDslProtocolTransType’ is configured as 〈TYPE2〉 and container ‘DcmDslPeriodicConnection’ is not configured

ERR053058: As the parameter 'DcmDslProtocolId’ is configured as 〈DCM_OBD_ON_CAN〉, value of the parameter 'DcmDsdSidTabServiceId’' in the container 'DcmDsdService’ should be configured as one of the following Protocol IDs 〈1,2,3,4,6,7,8,9,10〉.

This error occurs, if the value of parameter 'DcmDslProtocolId' is configured as 〈DCM_OBD_ON_CAN〉 in the container ‘DcmDslProtocolRow’, then the value of the parameter ‘DcmDsdSidTabServiceId’ is not configured as one of the following 〈1 or 2 or 3 or 4 or 6 or 7 or 8 or 9 or 10

ERR053059: Value 〈0 or 32 or 64 or 96 or 128 or 160 or 192 or 224〉 configured for parameter 'Parameter Name' in container 'Container Name' is invalid.

This error occurs, if the parameter ‘Parameter Name’ is configured as 〈0 or 32 or 64 or 96 or 128 or 160 or 192 or 224 in the container ‘Container Name’.

Container Name	Parameter Name
DcmDspVehInfo	DcmDspVehInfoInfoType
DcmDspTestResultObdMid	DcmDspTestResultObdMidTid
DcmDspRequestControlTestId	DcmDspRequestControl

ERR053060: If the value of the parameter 'DcmDspRoeQueueEnabled' is configured as 〈TRUE/1〉, then the container 'DcmDspExtRoe' needs to be configured.

This error occurs, if the value of the parameter 'DcmDspRoeQueueEnabled' is configured as 〈TRUE/1〉 then the container 'DcmDspExtRoe' needs to be configured.

ERR053061: Since the value of the parameter 'DcmDsdSidTabServiceId' is configured as 〈54〉, Atleast one instance of the parameter 'DcmDsdSidTabServiceId' in the container 'DcmDsdService' should be configured as 〈52〉 and 〈53〉.

This error occurs, if Atleast one instance of the parameter 'DcmDsdSidTabServiceId' in the container 'DcmDsdService' is not configured as 〈52〉 and 〈53〉, since the value of the parameter 'DcmDsdSidTabServiceId' is configured as 〈54〉

ERR053062: Since the value of the parameter ' DcmDsdSidTabServiceId ' is configured as 〈55〉, the value of the parameter ' DcmDsdSidTabServiceId ' should be configured as 〈54〉.

This error occurs, if the value of the parameter 'DcmDsdSidTabServiceId' is not configured as 〈54〉, since the value of the parameter ' DcmDsdSidTabServiceId ' is configured as 〈55〉.

ERR053063: The value of parameter ‘Parameter Name1' should be multiple of the configured value for parameter 'Parameter Name2'.

This error occurs, if the value of the parameter 'Parameter Name1' in the container ‘Container Name’ is not multiple of the parameter ‘Parameter Name2’.

Container Name	Parameter Name
DcmDspSecurity	DcmDspSecurityMaxAttemptCounterReadoutTime
DcmDspSecurityRow	DcmDspSecurityDelayTime
	DcmDspSecurityDelayTimeOnBoot
DcmDspSessionRow	DcmDspSessionP2ServerMax
	DcmDspSessionP2StarServerMax
DcmDslProtocolRow	DcmDslProtocolPreemptTimeout
	DcmTimStrP2ServerAdjust

ERR053064: As parameter 'DcmDsdSidTabServiceId' in container 'DcmDsdService' is configured as 〈Value〉, container 'Container Name' should be configured."

This error occurs, if the value of the parameter DcmDsdSidTabServiceId' is configured as 〈Value〉, then the container 'Container Name' is not configured.

Container Name	Value
DcmDspPid	1 or 2
DcmDspTestResultObdMid	6
DcmDspRequestControl	8
DcmDspVehInfo	9
DcmDspDid	34 or 36 or 42 or 44 or 46 or 47
DcmDspRoutine	49
DcmDspReadMemoryRangeInfo	61
DcmDspWriteMemoryRangeInfo	35

Container Name	Value
DcmDslReponseOnEvent	134
DcmDspComControl	40

ERR053065: The value of parameter 'DcmDsdSidTabServiceId' should be <Value> in the container 'DcmDsdService'.

This error occurs, if the value of the parameter DcmDsdSidTabServiceId' is not configured as <Value> in the container 'DcmDsdService'.

Value
16
39
62
133
20
25
34
36
42
44
46
47
49
52

Value
134
53
54
55
61
132
1
2
3
4
6
7
8
9
10

ERR053066: OBD services<DCM_OBD_ON_CAN/DCM_OBD_ON_FLEXRAY/DCM_OBD_ON_IP> should always have higher priority than other UDS services.

This error occurs, if the OBD Services <DCM_OBD_ON_CAN/DCM_OBD_ON_FLEXRAY/DCM_OBD_ON_IP> are configured a lower priority than other UDS services.

ERR053067: The Value of the parameter ' DcmDspMaxPeriodicDidScheduler ' should be configured as 0x01, when value of the parameter 'DcmDslProtocolTransType' is configured as Type1, for all the instances of the container

'DcmDslProtocolRow'.

This error occurs, if the Value of the parameter 'DcmDspMaxPeriodicDidScheduler' is not 0x01, when the value of the parameter 'DcmDslProtocolTransType' is configured as Type1, for all the instances of the container 'DcmDslProtocolRow'.

ERR053068: The Value of the parameter 'DcmDslProtocolRxBufferId' and value of the parameter ' DcmDslProtocolTxBufferId' in the container 'DcmDslProtocolRow ' should always be unique.

This error occurs, if the Value of the parameter 'DcmDslProtocolRxBufferId' and value of the parameter ' DcmDslProtocolTxBufferId' in the container 'DcmDslProtocolRow' are same.

ERR053069: The value of the parameter 'DcmDspMaxPeriodicDidScheduler' should be equal to the number of instances of the container 'DcmDslPeriodicConnection'.

This error occurs, if the value of the parameter 'DcmDspMaxPeriodicDidScheduler' is not equal to the number of instances of the container 'DcmDslPeriodicConnection'.

ERR053070: The Value(s) configured for the parameter 'DcmDslProtocolRxPduId' in the Container 'DcmDslProtocolRx' should be sequential.

This error occurs, if the value(s) configured for the parameter 'DcmDslProtocolRxPduId' in the Container 'DcmDslProtocolRx' are not sequential.

ERR053071: The value of the parameter "DcmDspDataSize" should be configured in the range of ((1 – 8) or (9 – 16) or (17 – 32)) (a value 8 or 16 or 32) when the parameter "DcmDspDataType" in container "DcmDspData" is configured as <UINT8/UINT16/UINT32/ SINT8/SINT16/SINT32>.

This error occurs, if the parameter 'DcmDspDataSize' is not configured in the range of ((1 – 8) or (9 – 16) or (17 – 32)), when the parameter 'DcmDspDataType' in container 'DcmDspData' is configured as <UINT8/UINT16/UINT32/SINT8/SINT16/SINT32> in case S/R.

ERR053074: The value of the parameter 'DcmDspDataType' should be configured as UINT8, since the value of the parameter 'DcmDspDataUsePort' is configured as <USE_BLOCK_ID/

USE_DATA_SYNCH_FNC/USE_DATA_ASYNCH_CLIENT_SERVER/ USE_DATA_SYNCH_CLIENT_SERVER/ /USE_DATA_ASYNCH_FNC> in the container 'DcmDspData'.

To check whether tool provides an error message, if the value of the parameter 'DcmDspDataType' is not configured as <UINT8>, when the value of the parameter 'DcmDspDataUsePort' is configured as <USE_BLOCK_ID/USE_DATA_SYNCH_FNC/USE_DATA_ASYNCH_CLIENT_SERVER/USE_DATA_SYNCH_CLIENT_SERVER/ /USE_DATA_ASYNCH_FNC> in the container 'DcmDspData'.

ERR053075: The value configured for the parameter 'DcmDspSessionP2StarServerMax' should be greater than 0.01s (10 ms) and should be multiple of 10.

To check whether tool provides an error message, if the value configured for the parameter 'DcmDspSessionP2StarServerMax' is less than 0.01s (10 ms) or is not a multiple of 10.

ERR053076: Atleast one instance of the container DcmDslMainConnection should be configured in the container DcmDslConnection.

To check whether tool provides an error message, if atleast one instance of the container 'DcmDslMainConnection' is not configured in the container DcmDslConnection.

ERR053076: Atleast one instance of the container DcmDslMainConnection should be configured in the container DcmDslConnection.

To check whether tool provides an error message, if atleast one instance of the container 'DcmDslMainConnection' is not configured in the container DcmDslConnection.

ERR053200: 'DcmDspDataSize' should not be configured as <0>.

To check whether DcmDspDataSize is a greater than <0> or not.

ERR053201: 'DcmDspDataUsePort' should not be configured as {USE_DATA_SENDER_RECEIVER, USE_BLOCK_ID, USE_ECU_SIGNAL} , since value of the 'DcmDspDataFixedLength' in the container 'DcmDspDataInfo' is configured as <FALSE/0>

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_BLOCK_ID, USE_ECU_SIGNAL}, the usage of variable data length shall be not allowed.

ERR053202: The value of the parameter 'DcmDspDataType' in the container 'DcmDspData' should be configured as <UINT8>, since value of the 'DcmDspDataFixedLength' in the container 'DcmDspDataInfo' is configured as <FALSE/0>"

Variable data length is only possible with UINT8 arrays with DcmDspDataType set to UINT8 and 'DcmDspDataFixedLength' set to <FALSE>.

ERR053203: DcmDspData elements used in service 0x2F shall not have 'DcmDspDataUsePort' set to USE_DATA_SENDER_RECEIVER.

ERR053204: The value of the parameter 'DcmDspDidDataPos' should be configured a multiple of <8> , since the values of the parameter 'DcmDspDataUsePort' is configured as <'one of all use port'> is configured as <'one of all type'> in the container 'DcmDspData'

Restrictions on bit-wise placement DcmDspDidDataPos Parameter shall address always a byte boundary, except DcmDspDataType is set to BOOLEAN, UINT8 or UINT16 with DcmDspDataSize lower than or equal 16.

ERR053205: "The value of the parameter 'DcmDspDataSize' should be configured a multiple of 8, since the value of the parameter 'DcmDspDataType' is configured as <NvM or C/S> in the container 'DcmDspData'

Restrictions on bit-wise access DcmDspDataSize shall be a multiple of 8 in case NvM or C/S.

ERR053206: The position of the current signal overlap the previous signals

ERR053207: The value of the parameter DcmDspRoutineSignalPos should address always a byte boundary, since the value of the parameter 'DcmDspRoutineSignalType' is configured as < UINT16/UINT32/SINT8/SINT16/SINT32>.

Restrictions on bit-wise placement DcmDspRoutineSignalPos parameter shall address always a byte boundary, except DcmDspRoutineSignalType is set to BOOLEAN or UINT8

ERR053208: The value of the parameter 'DcmDspRoutineSignalLength' should be configured a multiple of 8, since the value of the parameter 'DcmDspRoutineSignalType' is configured as < VARIABLE_LENGTH>.

Restrictions on bit-wise placement DcmDspRoutineSignalPos parameter shall address always a byte boundary, except DcmDspRoutineSignalType is set to BOOLEAN or UINT8

ERR053209: The value of parameter 'DcmDspRoutineSignalLength' should be in the range of ((1 - 8) or (9 - 16) or (17 - 32)) (a value 8 or 16 or 32) since value of the 'DcmDspRoutineSignalType' is configured as <UINT8/UINT16/UINT32/SINT8/SINT16/SINT32>.

This error occurs, if the parameter DcmDspRoutineSignalLength is not configured in the range of ((1 - 8) or (9 - 16) or (17 - 32)), when the parameter DcmDspRoutineSignalType is configured as <UINT8/UINT16/UINT32/SINT8/SINT16/SINT32> in case S/R.

ERR053210: The 'DcmDspRoutineSignalType' could be configured to <VARIABLE_LENGTH> for the last signal and when 'DcmDspRoutineFixedLength' is set to FALSE.

ERR053211: The position of the current signal overlap the previous signals

ERR053212: The position of the current signal overlap the previous signals

If you set the DcmPagedBufferEnabled to true, The value of parameter 'DcmPagedBufferTimeout' should be set to more than twice the configured value for parameter 'DcmTaskTime'.

ERR053213: The value of parameter 'DcmDspSecurityDelayTime' should be 180 to satisfy the DCM_ES95486_SUPPORT/DCM_ES95486_02_SUPPORT/DCM_ES95486_50_SUPPORT

If you set the standard Support value to DCM_ES95486_SUPPORT/DCM_ES95486_02_SUPPORT/DCM_ES95486_50_SUPPORT , The value of parameter 'DcmDspSecurityDelaytime' should be set to 180 to satisfy the ES Document.

ERR053214: The value of parameter 'DcmDspSecurityNumAttDelay' should be 3 to satisfy the DCM_ES95486_SUPPORT/DCM_ES95486_02_SUPPORT/DCM_ES95486_50_SUPPORT

If you set the standard Support value to DCM_ES95486_SUPPORT/DCM_ES95486_02_SUPPORT/DCM_ES95486_50_SUPPORT , The value of parameter 'DcmDspSecurityNumAttDelay' should be set to 3 to satisfy the ES Document.

ERR053215: Because of maximum read memory high range is 32 bits value. If parameter DcmDspSupportedAddressAndLengthFormatIdentifier configured with <CONFIG_VALUE> value. The total of MemoryAddress + MemorySize maybe be over 32 bits.

If user configure parameter DcmDspSupportedAddressAndLengthFormatIdentifier with **<CONFIG_VALUE>.**

The low nible byte (bit 3-0) of **<CONFIG_VALUE>** must be less or equal than 4. For make sure the The total of MemoryAddress + MemorySize which requested by user should not be overlap (over 32 bits).

ERR053216: Because of parameter DcmDspAuthenticationRoleSize is configured with value <VALUE>. The parameter DcmDsdServiceRole should be configured in range <RANGE>.

If user configurate the parameter **DcmDspAuthenticationRoleSize** with value **<VALUE>** (range 1 to 4). User must configure the parameter DcmDsdServiceRole in **<RANGE>.**

<VALUE> = 1, <RANGE> is 0..255

<VALUE> = 2, <RANGE> is 0..65535

<VALUE> = 3, <RANGE> is 0.. 16777215

<VALUE> = 4, <RANGE> is 0.. 4294967295

ERR053217: If Authentication (0x29) service available, the parameter DcmDslProtocolAuthenticaionConnectionId must be configured.

If user use Authentication Service, user must configure the parameter DcmDslProtocolAuthenticaionConnectionId.

ERR053218: If Authentication (0x29) service available, at least one of container DcmDspAuthenticationConnection must be configured.

If user use Authentication Service, user must configure at least one of container DcmDspAuthenticationConnection.

ERR053219: Because of parameter DcmDspAuthenticationRoleSize is configured with value <VALUE>. The parameter DcmDsdSubServiceRole should be configured in range <RANGE>.

If user configurate the parameter **DcmDspAuthenticationRoleSize** with value **<VALUE>** (range 1 to 4). User must configure the parameter DcmDsdSubServiceRole in **<RANGE>.**

<VALUE> = 1, <RANGE> is 0..255

<VALUE> = 2, <RANGE> is 0..65535

<VALUE> = 3, <RANGE> is 0.. 16777215

<VALUE> = 4, <RANGE> is 0.. 4294967295

ERR053220: If Authentication (0x29) service available, there is a container DcmDsdService must be configured with parameter DcmDsdSidTabServiceId set to 0x29.

If user use Authentication Service, user must configure at least one of container DcmDsdService with DcmDsdSidTabServiceId set to 0x29.

~~**ERR053221: Value(s) configured for the parameter DcmDslProtocolConnectionId in the container DcmDslMainConnection should be unique**~~

~~If user configure value(s) for the parameter DcmDslProtocolConnectionId in the container DcmDslMainConnection duplicately, this error shall occur.~~

~~**ERR053222: Value(s) configured for the parameter DcmDslProtocolConnectionId in the container DcmDslMainConnection should be configured.**~~

~~If user don't configure for the parameter DcmDslProtocolConnectionId in the container DcmDslMainConnection, this error shall occur.~~

ERR053224: If Authentication (0x29) service available and the paramter DcmDspAuthenticationESUsed set to True, Container DcmDspAuthenticationConnection should not be configured
If user set DcmDspAuthenticationConnectionES and DcmDspAuthenticationConnection together, this error shall occur.

ERR053226: The configured white list contain invalid data
If user set invalid value at white list, this error shall occur.

ERR053227: Because of parameter DcmDspAuthenticationRoleSize is configured with value '<role_size>'. The parameter DcmDspAuthenticationAuthenticatedRole should be configured in range <range>.
If user set wrong value at role size, this error shall occur.

ERR053230: If SecurityAccess (0x27) service is used with Crypto R44, the container DcmDspSecurityInfoRow must be configured correctly.
If user don't configure the container DcmDspSecurityInfoRow, this error shall occur.

ERR053231: If SecurityAccess (0x27) service is used with Crypto R40, the parameter related Crypto_R44 should be not set.
If user use Crypto R40 stack but the container DcmDspSecurityInfoRow is set, this error shall occur.

ERR053232: If RequestFileTransfer (0x38) service available, the container DcmDspRequestFileTransfer must be configured.
If user use RequestFileTransfer Service but the container DcmDspRequestFileTransfer isn't configued, this error shall occur.

ERR053233: If RequestFileTransfer (0x38) service use port, the parameter DcmRequestFileTransferMaxFileAndDirName must be configured.
If user use RequestFileTransfer Service with port but the paramter DcmRequestFileTransferMaxFileAndDirName isn't configued, this error shall occur.

ERR053234: The value of parameter 'parameter' shall be present only if 'DcmDspSecurityUsePort' is set to USE_ASYNCH_FNC and 'DcmDspSecurityAttemptCounterEnabled' is set to TRUE.
This error occurs, when a user uses DcmDspSecurityGetAttemptCounterFnc or DcmDspSecuritySetAttemptCounterFnc without DcmDspSecurityUsePort set to USE_ASYNCH_FNC and DcmDspSecurityAttemptCounterEnabled set to TRUE.

Container Name	Parameter Name
DcmDspSecurityRow	DcmDspSecurityGetAttemptCounterFnc
	DcmDspSecuritySetAttemptCounterFnc

9.2.2 Warning Messages

None.

9.2.3 Information Messages

INF053015: AUTOSAR Release version <Version> configured for the parameter 'AR-RELEASE-VERSION' in provided MDT file is not correct. AUTOSAR Release version should be one of the following: <Versions>.

This information message occurs, if the value of the element AR-RELEASE-VERSION present in the BSW Module Description template is configured other than 4.0.3

INF053051: Tool Expects user to configure DcmCpuByteOrder as (0/1), by default tool will consider LITTLE_ENDIAN (0). For BIG_ENDIAN (1), (-bend) needs to be passed in Commandline.

This information message occurs, if the user does not configure DcmCpuByteOrder as (0/1) in Commandline.

10 Det Errror

Detected development errors shall be reported to the Det_ReportError() service of the Development Error Tracer (DET) if Det error dection is enabled.

There is only one operation used as service from Development Error Tracer. In C-style, it looks as follows:

Std_ReturnType Xxx_ReportError(uint8 InstanceId, uint8 ApiId, uint8 ErrorId);

Note: ModuleId can be used in "port defined argument value".

10.1 Error classification

Type or error	*Relevance*	*Related error code*	*Value*
Interface: Timeout occurred during interaction with another module (e.g. maximum number of response pending is reached, refer	Development	DCM_E_INTERFACE_TIMEOUT	0x01

Type or error	*Relevance*	*Related error code*	*Value*
to SWS_Dcm_00120)			
Interface return-value is out of range	Development	DCM_E_INTERFACE_RETURN_VALUE	0x02
Interface: Boundary check of buffers provided by the Dcm failed during interaction with another module (application, Dem, PduR, etc.)	Development	DCM_E_INTERFACE_BUFFER_OVERFLOW	0x03
Internal: DCM not initialized	Development	DCM_E_UNINIT	0x05
DCM API function with invalid input parameter	Development	DCM_E_PARAM	0x06
DCM API service invoked with NULL POINTER as parameter	Development	DCM_E_PARAM_POINTER	0x07
Dcm initialisation failed	Development	DCM_E_INIT_FAILED	0x08

10.1.1 Service ID

Dcm function name	*Service ID[hex]*
Dcm_Init	0x01
Dcm_GetVersionInfo	0x24
Dcm_DemTriggerOnDTCStatus	0x2B
Dcm_<ModeName>ModeEntry	0x2C
Dcm_GetSecurityLevel	0x0D

Dcm function name	*Service ID[hex]*
Dcm_GetSesCtrlType	0x06
Dcm_GetActiveProtocol	0x0F
Dcm_ResetToDefaultSession	0x2A
Dcm_TriggerOnEvent	0x2D
Dcm_StartOfReception	0x00
Dcm_CopyRxData	0x02
Dcm_TpRxIndication	0x03
Dcm_CopyTxData	0x04
Dcm_TpTxConfirmation	0x05
Dcm_ComM_NoComModeEntered	0x21
Dcm_ComM_SilentComModeEntered	0x22
Dcm_ComM_FullComModeEntered	0x23
Dcm_Confirmation	0x29
Dcm_MainFunction	0x25
Dcm_StopROE	0x2e
Dcm_RestartROE	0x2f
Dcm_ExternalSetNegResponse	0x30
Dcm_ExternalProcessingDone	0x31
Dcm_SetDeauthenticatedRole	0x79
Dcm_Authentication_Function	0x41

Callout function name	*Service ID[hex]*
Dcm_ReadMemory	0x26
Dcm_WriteMemory	0x27
Dcm_ProcessRequestTransfertExit	0x32
Dcm_ProcessRequestUpload	0x31
Dcm_ProcessRequestDownload	0x30
Xxx_ReadData	0x34
Xxx_ReadData async	0x3b

Xxx_WriteData fixed	0x35
Xxx_WriteData	0x3e
Xxx_ReadDataLength	0x36
Xxx_ConditionCheckRead	0x37
Xxx_GetScalingInformation	0x38
Xxx_ReturnControlToECU	0x39
Xxx_ResetToDefault	0x3c
Xxx_FreezeCurrentState	0x3a
Xxx_ShortTermAdjustment	0x3d
Xxx_IsDidAvailable	0x3F
Xxx_ReadDidData	0x40
Xxx_WriteDidData	0x41
Dcm_ExternalSetNegResponse	0x30
Dcm_ExternalProcessingDone	0x31
〈Module〉_〈DiagnosticService〉	0x32
〈Module〉_〈DiagnosticService〉_〈SubService〉	0x33

11 Appendix

11.1ES95486 Support

11.1.1Type Definitions

11.1.1.1 Dcm_CertificationInfoType

Name:	Dcm_CertificationInfoType		
Type:	Structure		
Element:	uint8*	authorization	The authorization information of certification. This variable is array with 4 bytes size.
	uint8*	expirationDate	The expiration date information of certification. This variable is array with 3 bytes size.

11.1.2Interfaces

11.1.2.1 DCMServices

11.1.2.1.1 Dcm_GetCertificationInfo

Function Name	Dcm_GetCertificationInfo	
Syntax:	Std_ReturnType Dcm_GetCertificationInfo (Dcm_CertificationInfoType *LpCertifInfo)	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (In)	None	
Parameters (Inout)	None	
Parameters (Out)	LpCertiInfo	information of certification

Return Value	Std_ReturnType	E_OK: Current security level is 0x21. LpCertiInfo are valid. E_NOT_OK: RandomSeed is invalid
Description	This function returns the field information of certification. The field information includes 'Certificate Expiration Date' and 'Certificate Holder Reference : Authorization'. Certificate Holder Reference : Authorization structure is like below.	
Precondition	Security level using CSAC algorithm is unlock	

Certification Holder Reference

Bit31..24	Bit23	Bit22	Bit21	Bit20..16	Bit15	Bit2	Bit1	Bit0
…	SubCA(1) or End Entity(0)	CGW	All Internal ECU	For future use		Group#3	Group#2	Group#1
Reserved	Role	Target			Permission			

11.1.2.2 Callout Function

11.1.2.2.1 Dcm_GetRandomSeed

Function Name	Dcm_GetRandomSeed	
Syntax:	Std_ReturnType Dcm_GetRandomSeed (uint8* RandomSeed)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	None	
Parameters (Inout)	None	
Parameters (Out)	RandomSeed	RandomSeed To be applied to the C-SAC platform
Return Value	Std_ReturnType	E_OK: RandomSeed is valid Apply user randomseed in platform E_NOT_OK: RandomSeed is invalid
Description	This function is used to update the RandomSeed used in the C-SAC algorithm.	
Precondition	Use a SecurityLeveL L21 (CSAC)	

11.1.2.2.2 Dcm_GetPublicKey

Function Name	Dcm_GetPublicKey	
Syntax:	void Dcm_GetPublicKey (uint8* PublicKey)	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (In)	None	
Parameters (Inout)	None	
Parameters (Out)	PublicKey	Public key for certificate authentication
Return Value	none	-
Description	This function is used to update the RandomSeed used in the C-SAC algorithm.	
Precondition	Use a SecurityLeveL L21 (CSAC)	

11.1.3 Callout Function Guide

11.1.3.1 If C-SAC is applied, update the Seed through Dcm_GetRandomSeed()

Note: Under this guide, **for use of HAC Random, instead of True Random-driven** algorithm (Security Level 0x21 (C-SAC), mandatory applications are described.

To apply C-SAC, RandomSeed should be updated through Callout Code in the Application to increase randomness. Apply the RandomSeed value with guaranteed randomness to RandomSeed[16], the transfer factor of Dcm_GetRandomSeed() provided by Dcm. The followings are time points when the Dcm_GetRandomSeed() callout function is called.

1) When C-SAC is requested for the first time
2) When Entropy of HacRandomGenerate is exhausted (return CAL_E_ENTROPY_EXHAUSTION)

Since the platform determines whether user RandomSeed is applied by checking the return value of Dcm_GetRandomSeed, the Return value of the Callout function must be applied as E_OK after seed update in the application.

Return Value	Description
E_OK	User RandomSeed must be applied. In the event of RandomSeed update, must return E_OK.
E_NOT_OK	Negative response (0x33)

For more information on the Callout function, see Interface in [Appendix 10.1 ES95489 Support].

11.1.3.2 Provide PublicKey through Dcm_GetPublickKey() in C-SAC applied controller implemented with own Fbl

Note: This guide is mandatory if AutoEver Fbl is not used while Security Level 0x21 (C-SAC) algorithm is applied.

The PublicKey used during authentication of C-SAC certificate is provided by AutoEver Fbl. If the Bootloader is implemented by itself without using the AutoEver Fbl, the PublicKey for C-SAC certificate authentication must be provided to the platform.
If DcmGeneral/DcmAutonFblUsed setup is False in Dcm_Ecud.arxml configuration file, Dcm_GetPublicKey() callout function is provided. The internal PublicKey should be applied to the transfer factor PublicKey[256].

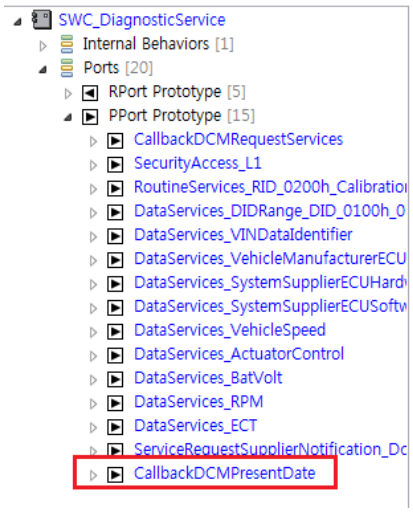
**Asims

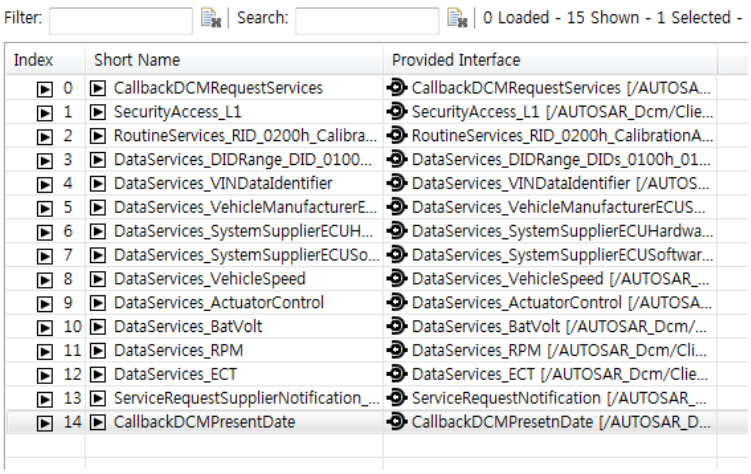
For more information on the Callout function, see Interface in [Appendix 10.1 ES95489 Support].

11.1.4 Security Access 2.0 Guide

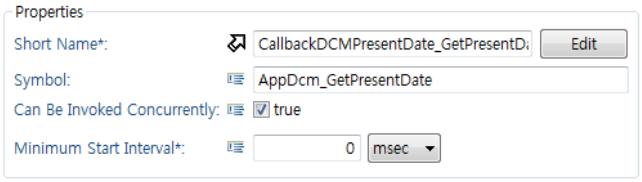
1. In order to use the Security Access 2.0 function successfully, the application must provide the current date information to Dcm, and to this end, P-Port, Runnable, and API must be added.

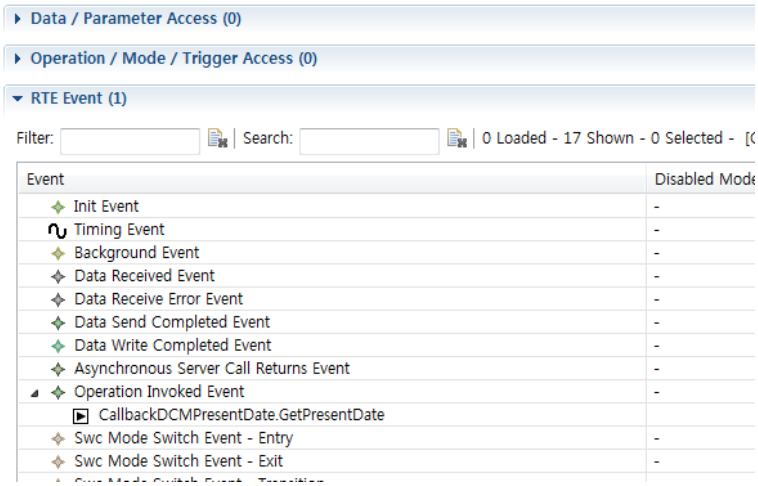
   (1) Add P-Port





   (2) Add Runnable





(3) Add API and save date

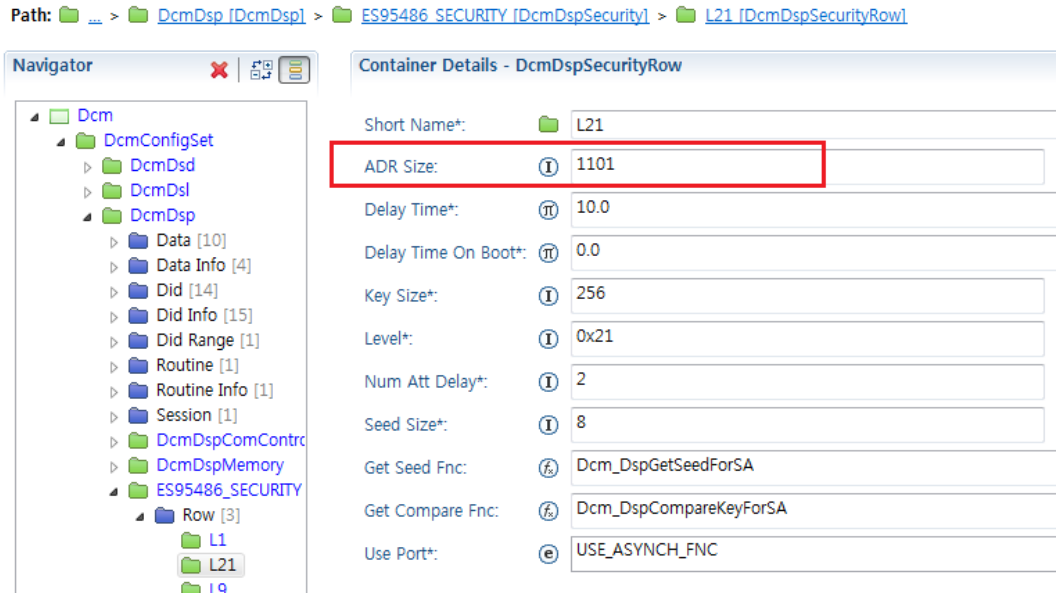
AppDcm_GetPresentDate

```
FUNC(Std_ReturnType, RTE_CODE) AppDcm_ GetPresentDate (
OUT P2VAR(uint8, AUTOMATIC, RTE_APPL_DATA) Data)
{
 Std_ReturnType LddRetVal = RTE_E_OK;

  /*
  * Apply present date to Data.
  * present date is a 3-row array.
  * Data[0] = year(h), Data[1] = month(h), Data[2] = day(h)
  * ex) 2020. 02.25
  * Data[0] = 0x20;
  * Data[1] = 0x02;
  * Data[2] = 0x25;
  */

  Data[0] = 0x20;
  Data[1] = 0x02;
  Data[2] = 0x25;

  return LddRetVal;
}
```

For example, if the current date is February 25, 2020 it should be saved in the format of
Date[0] = 0x20
Date[1] = 0x02
Date[2] = 0x25.
Also, the application must check validity of the current date saved.

2. ADR Size Setup

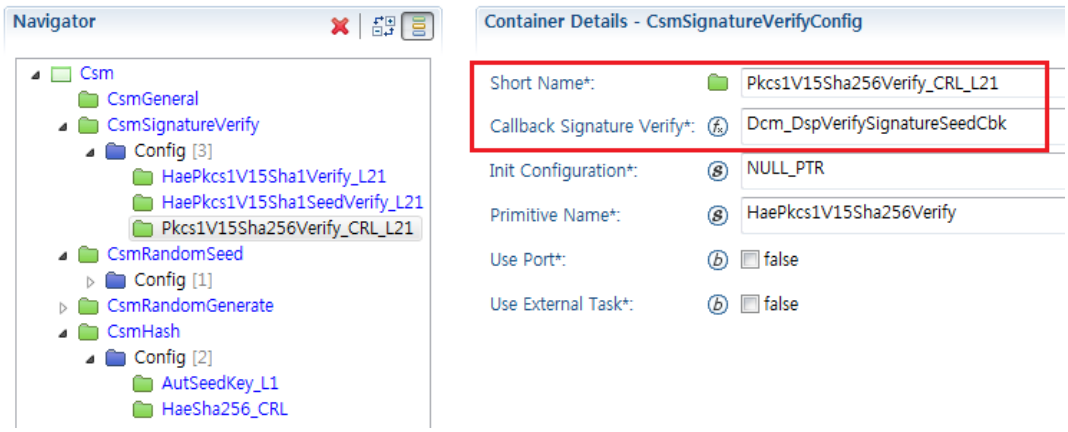
In Security Access 2.0, CRL 501 bytes (extendable) are attached to the 600 bytes of the certificate transmitted when seed is requested from the existing C-SAC and transmitted. Therefore, the CRL size should be added to the ADR size that used to be set as 600.



3. Addition of CSM setting

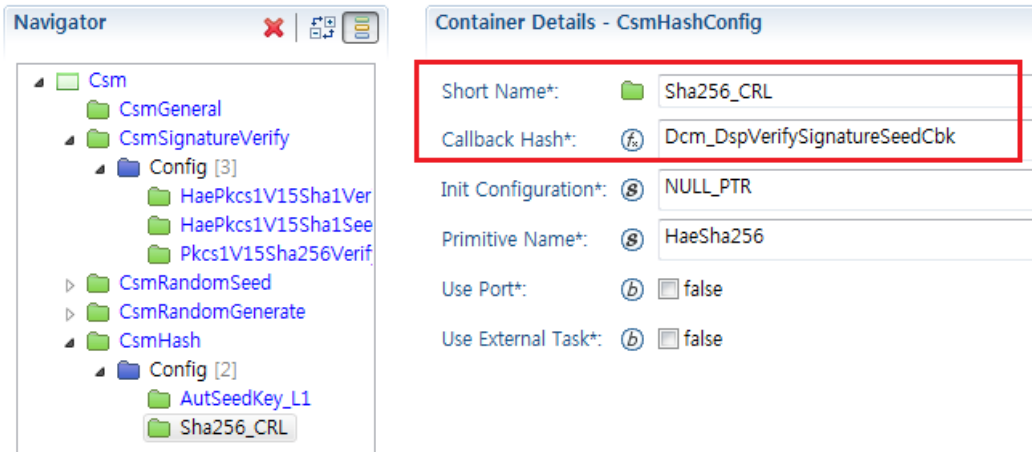
Security Access 2.0 requires hash operation for CRL signature verification and CRL issuer public key identifier verification in addition to the existing certificate signature verification. Therefore, CsmSignatureVerify for the CRL signature verification and CsmHash setting for the public key identifier verification should be added to CSM.

(1) The algorithm for the CRL signature verification should use SAH256WithRSA and Short Name must be set up as Pkcs1V15Sha256Verify_CRL_L21. The callback function should be configured as Dcm_DspVerifySignatureSeedCbk.



(2) The hash algorithm to verify the public key identifier should use SHA256 and Short Name should be

configured as Sha256_CRL for Dcm to recognize. The callback function should be configured as Dcm_DspVerifySignatureSeedCbk.

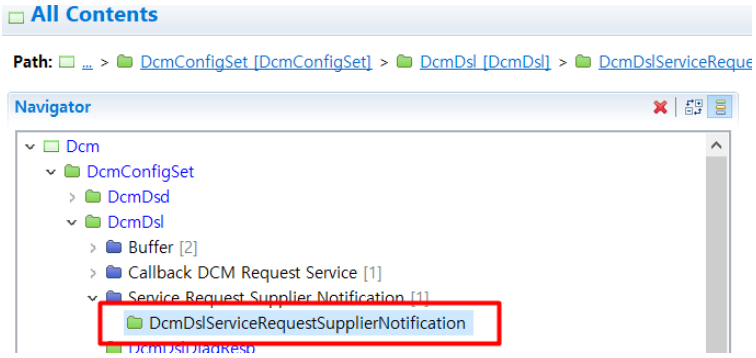


11.1.5 Constraints in diagnostic services depending on engine condition

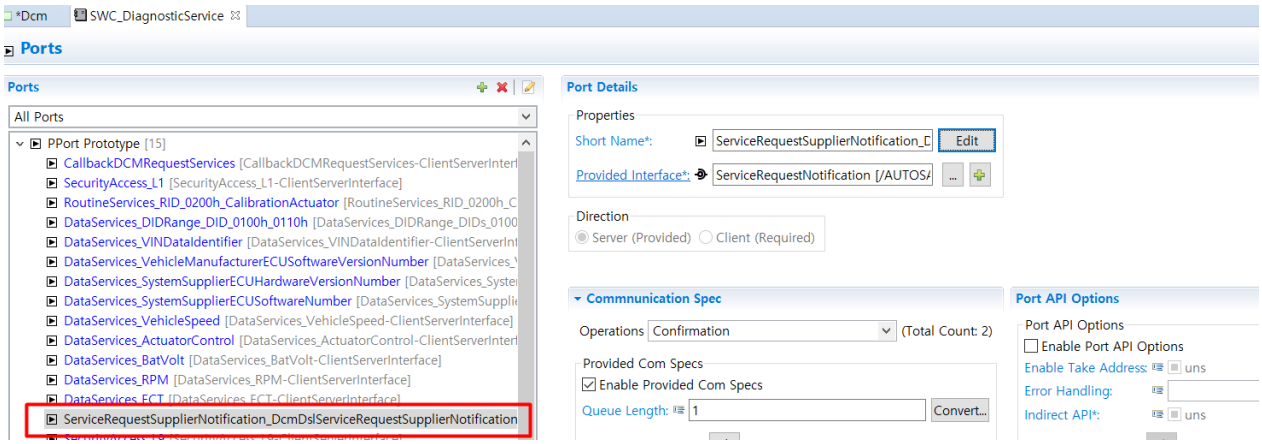
Depending on diagnostic specifications, functioning of controller diagnostic services can be determined.

11.1.5.1 Set DcmDslServiceRequestSupplierNotification

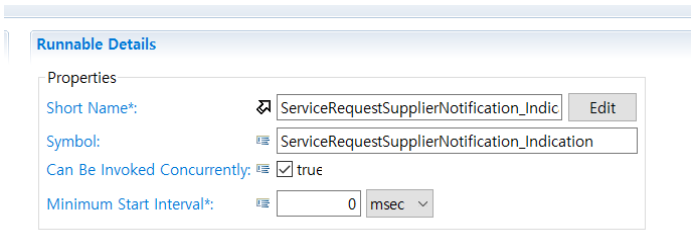
(1) Set DcmDsl/ DcmDslServiceRequestSupplierNotification



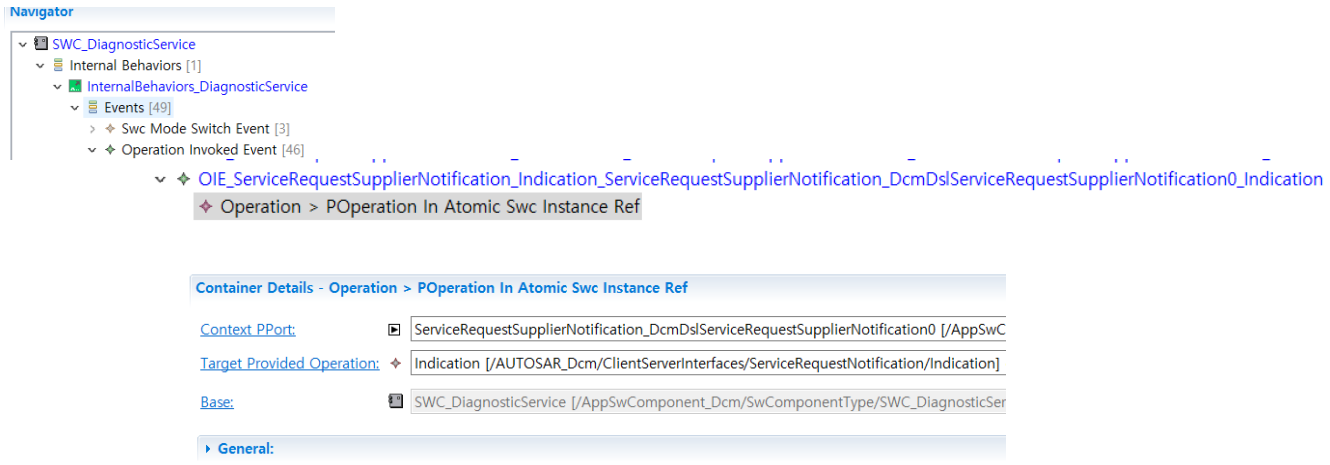
(2) Add P-Port



(3) Add Runnable



(4) Add Operation Invoked Event



11.1.5.2 Xxx_Indication

Function Name	Xxx_Indication
Syntax:	Std_ReturnType Xxx_Indication (uint8 SID, uint8* RequestData, uint16 DataSize, uint8 ReqType, uint16 SourceAddress, Dcm_NegativeResponseCodeType* ErrorCode)

Refer	For functional prototype, see _Xxx_Indication #Xxx_Indication.

The following codes are examples to ease understanding. The Callback function should be applied depending on the environment of the controller.

XXX_Indciation

```
/*******************************************************************************
** Function Name         : Xxx_Indication
** Input Parameters      :
** - SID : Service ID
** - RequestData : Pointer to received data
** - DataSize : Data length of received data
** - ReqType : Rx message address type
**               1 : Functional Address
**               0: Physical Address
**
** - SourceAddress : Source address (Refer to configuration DcmDslProtocolRxTesterSourceAddr)
**
** Output Parameters     :
** - ErrorCode : If this operation returns value E_NOT_OK, the Dcm module
**    shall send a negative response with NRC code equal to the parameter
**
**    ErrorCode parameter value. (Refer to the Rte_Dcm_Type.h)
**
** Return parameter      :
** - Std_ReturnType
**    - RTE_E_OK : Request was successful
**    - RTE_E_Xxx_E_NOT_OK : Request was not successful
*******************************************************************************/
Std_ReturnType Xxx_Indication (uint8 SID, uint8* RequestData, uint16 DataSize, uint8 ReqType, uint16 SourceAddress, Dcm_NegativeResponseCodeType* ErrorCode )
{

  Std_ReturnType retVal = - RTE_E_OK;

  If (engine condition is met) /* Ex)  Engine Stop, IGN on */
  {
    /* Return negative response if the service specified in the specifications cannot be run in a certain condition. */
    If (
        ( ##1 == SID)  ||
        ( ##1 == SID)  ||
        ( ##1 == SID)  ||
        (…            )
    )
    {
      *ErrorCode = DCM_E_CONDITIONSNOTCORRECT;
      retVal = RTE_E_Xxx_E_NOT_OK;
    }
  }
  return RetVal;
```

```
}
```

11.2 SecurityAccess Sample Code

Note: The following L1 and L9 examples ares for reference and cannot be simply applied to the project.

Note: The function name is based on the deployed project and users can change it arbitrarily.

Note: For the Csm API usage part, it is needed to refer to the latest Csm module.

11.2.1 Seed-Key Algorithm (L1)

Header File Include

```
/* Header File Inclusion for SecurityAccess */
#include "Rte_SWC_DiagnosticService.h"
```

Global Variables

```
/* Global Variables for SecurityAccess_L1 */
uint8 AppDcm_GaaSeed[4] = {0U, };
```

When SecurityAccess (Seed-Key, L1) is requested, the following function will be called according to Subfunction.

11.2.1.1 RequestSeed (27 01) : AppDcm_GetSeed_L1

- AppDcm_GetSeed_L1 is a function that platform calls when it received Seed Response request from Diagnostics.

Return the Seed that Application created in the function.

AppDcm_GetSeed_L1

```
FUNC(Std_ReturnType, RTE_CODE)AppDcm_GetSeed_L1(
  IN Dcm_OpStatusType OpStatus,
  OUT P2VAR(uint8, AUTOMATIC, RTE_APPL_DATA) Seed,
  OUT P2VAR(Dcm_NegativeResponseCodeType, AUTOMATIC, RTE_APPL_DATA) ErrorCode)
{
  Std_ReturnType LddRetVal = RTE_E_OK;

  /* Generate Seed from random generator */
  if (RTE_E_OK != Rte_Call_AutRandomGenerate_L1_RandomGenerate(&AppDcm_GaaSeed[0], 4U) )
  {
    *ErrorCode = DCM_E_CONDITIONSNOTCORRECT;
    LddRetVal = RTE_E_SecurityAccess_L1_E_NOT_OK;
  }
  else
  {
    uint8 LucIndex;
    for (LucIndex = 0U; LucIndex < 4U; LucIndex++)
    {
      Seed[LucIndex] = AppDcm_GaaSeed[LucIndex];
    }
  }
```

```
  return LddRetVal;
}
```

11.2.1.2 SendKey (27 02 XX XX XX XX) : AppDcm_CompareKey_L1

- AppDcm_CompareKey_L1 is a function that the platform calls when the diagnostics requested Key value and security level access. The security level access is finally determined by comparing the key of the function's transfer factor with the internal key in the application.

AppDcm_CompareKey_L1

```
FUNC(Std_ReturnType, RTE_CODE)AppDcm_CompareKey_L1(
  IN P2CONST(uint8, AUTOMATIC, RTE_APPL_DATA) Key,
  IN Dcm_OpStatusType OpStatus)
{
  Std_ReturnType LddRetVal = RTE_E_SecurityAccess_L1_E_NOT_OK;

  /* Generate Key based on SeedKey Algorithm */
  if (RTE_E_OK == Rte_Call_AutSeedKey_L1_HashStart() )
  {
    if (RTE_E_OK == Rte_Call_AutSeedKey_L1_HashUpdate(&AppDcm_GaaSeed[0], 4U);
    {
      uint8 LaaKey[4] = {0U, };
      uint32 LulLength = sizeof(LaaKey) / sizeof(LaaKey[0]);

      if (RTE_E_OK == Rte_Call_AutSeedKey_L1_HashFinish(LaaKey, &LulLength, FALSE);
      {
        uint8 LucIndex;
        for (LucIndex = 0U; LucIndex < LulLength; LucIndex++)
        {
          if (LaaKey[LucIndex] != Key[LucIndex])
          {
            break;
          }
        }
        if (LucIndex == LulLength)
        {
          /* key matches */
          LddRetVal = RTE_E_OK;
        }
        else
        {
          LddRetVal = RTE_E_SecurityAccess_L1_DCM_E_COMPARE_KEY_FAILED;
        }
      }
    }
  }
  return LddRetVal;
```

```
}
```

11.2.2 Advanced Seed-Key Algorithm (L9)

Header File Include
/* Header File Inclusion for SecurityAccess */ #include "Rte_SWC_DiagnosticService.h"

Global Variables
/* Global Variables for SecurityAccess_L9 */ uint8 AppDcm_GaaAdvSeed[8] = {0U, };

If SecurityAccess (Advanced Seed-Key, L9) is requested, the following function will be called according to Subfunction.

11.2.2.1 RequestSeed (27 11) : AppDcm_GetSeed_L9

- AppDcm_GetSeed_L9 is a function that platform calls when it received Seed Response request from Diagnostics. Return the Seed that Application created in the function.

11.2.2.1.1 If Pseudo Random is in use

If pseudo random, rather than true random, is used for Advanced Seed-Key Algorithm, prior to RequestSeed, for random (Seed(1)) creation, RandomSeed(2)must be updated. Based on the updated RandomSeed, Seed is created. The randomness is guaranteed by Hae_CryptoLib. Therefore, the designer of the Application should update RandomSeed for at least once prior to the first RequestSeed, and **Randomness of RandomSeed must be guaranteed in the Application.**

The following Sample Code is how to update RandomSeed only for once, internally using Flag when AppDcm_GetSeed_L9 is called for the first time.

(1) Seed : A seed value for generation 'key' in SecurityAccess requested

(2) RandomSeed : A seed value for generation 'Seed' in SecurityAccess requested

AppDcm_GetRandomSeed
/*** ** Function Name : AppDcm_GetRandomSeed ** ** Sync/Async : Sync ** Input Parameters : ** RandomSeed : A seed value for generation 'Seed' in SecurityAccess requested ** ** Return parameter : ** - Std_ReturnType ** - E_OK : Request was successful ** - E_NOT_OK : Request was not successful

```
**********************************************************************************/
FUNC(Std_ReturnType, RTE_CODE) AppDcm_GetRandomSeed(
  P2VAR(uint8, AUTOMATIC,DCM_APPL_DATA) RandomSeed)
{
  Std_ReturnType RetVal = E_NOT_OK;

 /*
  * Apply user RandomSeed to the argument in this callout function.
  * RandomSeed is a 16-row array. (RandomSeed[16])
  */

 /*
  * Apply user RandomSeed to the argument in this function.
  * RandomSeed is a 16-row array. (RandomSeed[16])
  * Assign a random number to this variable (RandomSeed)
  * ex) current time, tick count, event timer, ADC noise, sensor value, etc..
  */

 /*
  * If you apply RandomSeed, you must return E_OK.
  * E_OK : Use the user RandomSeed
  * E_NOT_OK : Use the Autoevern RandomSeed
  */

  #ifdef INSTST_TESTCODE_INTEGRATED
  RetVal = E_OK;
  #endif

  return RetVal;
}
```

AppDcm_GetSeed_L9

```
FUNC(Std_ReturnType, RTE_CODE) AppDcm_GetSeed_L9(
  IN Dcm_OpStatusType OpStatus,
  OUT P2VAR(uint8, AUTOMATIC, RTE_APPL_DATA) Seed,
  OUT P2VAR(Dcm_NegativeResponseCodeType, AUTOMATIC, RTE_APPL_DATA) ErrorCode)
{
  Std_ReturnType LddRetVal = RTE_E_SecurityAccess_L9_E_NOT_OK;
  volatile static boolean LblRandomSeedUpdated = FALSE;

  if (FALSE == LblRandomSeedUpdated)
  {
    uint8 LaaHaeHacSalt_L9[APPDCM_L9_HAE_PSEUDO_ENTROPY_LENGTH] = {0U};

    /* Get RandomSeed */
    LddRetVal = AppDcm_GetRandomSeed(LaaHaeHacSalt_L9);

    /* Update RandomSeed */
    if (RTE_E_OK == Rte_Call_HaePseudoRandomSeed_L9_RandomSeedStart())
    {
```

```
      if (RTE_E_OK == Rte_Call_HaePseudoRandomSeed_L9_RandomSeedUpdate(LaaHaeHacSalt_L9,
APPDCM_L9_HAE_PSEUDO_ENTROPY_LENGTH))
      {
        if (RTE_E_OK == Rte_Call_HaePseudoRandomSeed_L9_RandomSeedFinish())
        {
          LblRandomSeedUpdated = TRUE;
        }
      }
    }
  }

  if (TRUE == LblRandomSeedUpdated)
  {
    if (RTE_E_OK == Rte_Call_HaePseudoRandomGenerate_L9_RandomGenerate(&AppDcm_GaaAdvSeed[0],
8U))
    {
      uint8 LucIndex;
      for (LucIndex = 0U; LucIndex < 8U; LucIndex++)
      {
        Seed[LucIndex] = AppDcm_GaaAdvSeed[LucIndex];
      }
      LddRetVal = RTE_E_OK;
    }
  }
  if (RTE_E_OK != LddRetVal)
  {
    *ErrorCode = DCM_E_CONDITIONSNOTCORRECT;
  }

  return LddRetVal;
}
```

11.2.2.1.2 If True Random is in use

If True Random is applied to Seed creation, create Seed with True Random for the first time and then use Pseudo Random to create Seed from the next time.
To this end, as in **AppDcm_GetSeed_L9** of 10.2.2.1.2.1 and 10.2.2.1.2.2 PseudoRandomGenerateshould be called.

If Autoever HSM 2.x / AutoEver HSM 1.0 SPC58x is in use

AppDcm_GetSeed_L9

```
FUNC(Std_ReturnType, RTE_CODE) AppDcm_GetSeed_L9(
  IN Dcm_OpStatusType OpStatus,
  OUT P2VAR(uint8, AUTOMATIC, RTE_APPL_DATA) Seed,
  OUT P2VAR(Dcm_NegativeResponseCodeType, AUTOMATIC, RTE_APPL_DATA) ErrorCode)
{
  Std_ReturnType LddRetVal = RTE_E_SecurityAccess_L9_E_NOT_OK;

  #if defined(APPDCM_L9_AUTRON_HSM_RANDOM)
  /*****************************************************************************
   *                    true random generator(Chorus SPC58)                    *
   *****************************************************************************/
  if (RTE_E_OK == Rte_Call_AutHsmPseudoRandomGenerate_L9_RandomGenerate(&AppDcm_GaaAdvSeed[0],
8U))

  #elif defined(APPDCM_L9_HAE_HSM_RANDOM)
  /*****************************************************************************
   *                    true random generator(Autoever HSM)                    *
   *****************************************************************************/
  if (RTE_E_OK == Rte_Call_HaeHsmPseudoRandomGenerate_L9_RandomGenerate(&AppDcm_GaaAdvSeed[0],
8U))

  #endif
  {
    uint8 LucIndex;
    for (LucIndex = 0U; LucIndex < 8U; LucIndex++)
    {
      Seed[LucIndex] = AppDcm_GaaAdvSeed[LucIndex];
    }
    LddRetVal = RTE_E_OK;
  }
  else
  {
    *ErrorCode = DCM_E_CONDITIONSNOTCORRECT;
  }

  return LddRetVal;
}
```

◆ Cautions in using True Random Generate through Autoever HSM 2.0

When porting HSM, depending on the characteristics of MCU, it may be necessary to call HSM to Host Interrupt Enable API (ex)HSM_EnableService. **Therefore, HSM UM must be checked before implementation.**

Ex) In the case of RH850, mobilgene is using Renesas OS. Interrupt starts operating only if HSM to Host Interrupt Enable API(ex)HSM_EnableService) is called.

(In the case of MCU using AutoEver's own OS, the OS itself activates the interrupt when it is set in the OS)

If AutoEver HSM 1.0 RH850 F1KM / F1K ICUS is in use

Global Variables for RH850 F1K 2M

```
/* Global Variables for SecurityAccess_L9 */
uint8 AppDcm_GaaAdvSeed[8];
Std_ReturnType   AppDcm_SeedGenResult;
```

AppDcm_GetSeed_L9

```
FUNC(Std_ReturnType, RTE_CODE) AppDcm_GetSeed_L9(
  IN Dcm_OpStatusType OpStatus,
  OUT P2VAR(uint8, AUTOMATIC, RTE_APPL_DATA) Seed,
  OUT P2VAR(Dcm_NegativeResponseCodeType, AUTOMATIC, RTE_APPL_DATA) ErrorCode)
{
  Std_ReturnType LddRetVal = RTE_E_SecurityAccess_L9_E_NOT_OK;

  switch(OpStatus)
  {
  case DCM_INITIAL:
    #if defined(APPDCM_L9_ICUS_RANDOM)
    /*********************************************************************
     *                  true random generator(RH850 ICUS)                *
     *********************************************************************/
    if (RTE_E_OK == Rte_Call_AutIcusRandomGenerate_L9_RandomGenerate(&AppDcm_GaaAdvSeed[0], 8U))

    #elif defined(APPDCM_L9_AUTRON_HSM_RANDOM_ICUM)
    /*********************************************************************
     *                  true random generator(RH850 ICUM)                *
     *********************************************************************/
    if                                          (RTE_E_OK                                          ==
Rte_Call_AutHsmPseudoRandomGenerate_L9_RandomGenerate(&AppDcm_GaaAdvSeed[0], 8U))

    #endif
    {
      AppDcm_SeedGenResult = RTE_E_SecurityAccess_L9_DCM_E_PENDING;
      LddRetVal = RTE_E_SecurityAccess_L9_DCM_E_PENDING;
    }
    break;

  case DCM_PENDING:
    LddRetVal = AppDcm_SeedGenResult;
    if (RTE_E_OK == AppDcm_SeedGenResult)
    {
      uint8 LucIndex;
      for (LucIndex = 0U; LucIndex < 8U; LucIndex++)
      {
        Seed[LucIndex] = AppDcm_GaaAdvSeed[LucIndex];
      }
    }
    break;
```

```
  case DCM_CANCEL:
    LddRetVal = RTE_E_OK;
    break;
  default :
    break;
  }

  if (RTE_E_SecurityAccess_L9_E_NOT_OK == LddRetVal)
  {
    *ErrorCode = DCM_E_CONDITIONSNOTCORRECT;
  }
  return LddRetVal;
}
```

AppDcm_GetSeedCbk_L9

```
FUNC(Std_ReturnType, SWC_DiagnosticService_CODE) AppDcm_GetSeedCbk_L9(
  IN VAR(Std_ReturnType, AUTOMATIC) retVal)
{
  if (retVal == RTE_E_OK)
  {
    AppDcm_SeedGenResult = RTE_E_OK;
  }
  else
  {
    AppDcm_SeedGenResult = RTE_E_SecurityAccess_L9_E_NOT_OK;
  }

  return RTE_E_OK;
}
```

11.2.2.2 SendKey (27 12 XX XX XX XX XX XX XX XX) : AppDcm_CompareKey_L9

- AppDcm_CompareKey_L9 is a function called when the platform receives a security level access request along with the key value from the diagnostics. The security level access is finally determined by comparing the key of the function's transfer factor with the internal key in the application.

AppDcm_CompareKey_L9

```
FUNC(Std_ReturnType, RTE_CODE) AppDcm_CompareKey_L9 (
  IN P2CONST(uint8, AUTOMATIC, RTE_APPL_DATA) Key,
  IN Dcm_OpStatusType OpStatus)
{
  Std_ReturnType LddRetVal = RTE_E_SecurityAccess_L9_E_NOT_OK;
  /*********************************************************************
   *            generate the   key   based on the advanced seedKey algorithm                *
   *********************************************************************/
  if (RTE_E_OK == Rte_Call_HaeAdvSeedKey_L9_HashStart() )
  {
    if (RTE_E_OK == Rte_Call_HaeAdvSeedKey_L9_HashUpdate(&AppDcm_GaaAdvSeed[0], 8U) )
```

```
    {
      uint8 LaaKey[8] = {0U, };
      uint32 LulLength = sizeof(LaaKey) / sizeof(LaaKey[0]);
      if (RTE_E_OK == Rte_Call_HaeAdvSeedKey_L9_HashFinish(LaaKey, &LulLength, FALSE))
      {
        uint8 LucIndex;
        for (LucIndex = 0U; LucIndex < LulLength; LucIndex++)
        {
          if (LaaKey[LucIndex] != Key[LucIndex])
          {
            break;
          }
        }
        if (LucIndex == LulLength)
        {
          LddRetVal = RTE_E_OK; /* key matches */
        }
        else
        {
          LddRetVal = RTE_E_SecurityAccess_L9_DCM_E_COMPARE_KEY_FAILED;
        }
      }
    }
  }
  return LddRetVal;
}
```

11.2.3 Xxx_GetSecurityAttemptCounter

If DcmDspSecurityAttemptCounterEnabled == true,
Dcm determines the initial value of the Security Attempt Counter through the results of executing Xxx_GetSecurityAttemptCounter for all Security Levels during ECU startup.

Xxx_GetSecurityAttemptCounter reads the Security Attempt Counter for a specific Security Level from the application. Application must read the Security Attempt Counter of a specific Security Level from non-volatile memory and return the return value of Xxx_GetSecurityAttemptCounter and AttemptCounter appropriately.

If the result of this function is E_NOT_OK or the operation is canceled due to expiration of DcmDspSecurityMaxAttemptCounterReadoutTime, Dcm determines the Security Attempt Counter of the corresponding Security Level as DcmDspSecurityNumAttDelay and starts the Security Delay Timer.

Note: If the state of nov-volatile memory is virgin state, the application must return the return value of Xxx_GetSecurityAttemptCounter and AttemptCounter appropriately.

Below is an example of storing a Security Attempt Counter in NvM.

AppDcm_GetSecurityAttemptCounter_L9

```
/* Global variable for security attempt counter of security level 9 */
uint8 AppDcm_SecurityAttemptCounter_L9 = 0x00;
```

```
FUNC(Std_ReturnType, SWC_DiagnosticService_CODE) AppDcm_GetSecurityAttemptCounter_L9(
  IN VAR(Dcm_OpStatusType, AUTOMATIC) OpStatus,
  OUT P2VAR(uint8, AUTOMATIC, RTE_APPL_DATA) AttemptCounter)
{
  Std_ReturnType LddRetVal = RTE_E_SecurityAccess_L9_E_NOT_OK;
  Std_ReturnType LddNvMRetVal = E_NOT_OK;
  NvM_RequestResultType LddNvMRequestResult = NVM_REQ_NOT_OK;

  /*
    Dcm must receive the result of NvM_ReadBlock() job from the application
    to initialize the attempt counter of each security level.
    Therefore, NvM_GetErrorStatus() must be called sequentially
    after NvM_ReadBlock() returns E_OK.
  */
  switch (OpStatus)
  {
    case DCM_INITIAL:
      /* NOTE :
        Since the argument AttemptCounter is a local variable pointer passed from Dcm,
        it must not be passed as an argument to NvM_ReadBlock().
        Pass the NvMRamBlockDataAddress of this NvM Block as an argument to NvM_ReadBlock(). */
      if (E_OK == Rte_Call_SWC_DiagnosticService_NvMService_DcmSecurityAttemptCounter_L9_ReadBlock(
          (void *)AppDcm_SecurityAttemptCounter_L9))
      {
        LddRetVal = RTE_E_SecurityAccess_L9_DCM_E_PENDING;
      }
      else
      {
        LddRetVal = RTE_E_SecurityAccess_L9_E_NOT_OK;
      }
      break;

    case DCM_PENDING:
      LddNvMRetVal = Rte_Call_SWC_DiagnosticService_NvMService_DcmSecurityAttemptCounter_L9_GetErrorStatus(
       &LddNvMRequestResult);
      if (E_OK == LddNvMRetVal)
      {
        if (NVM_REQ_OK == LddNvMRequestResult)
        {
          /* Get the attempt counter of security level 9 */
          *AttemptCounter = AppDcm_SecurityAttemptCounter_L9;
          LddRetVal = RTE_E_OK;
        }
        else if (NVM_REQ_PENDING == LddNvMRequestResult)
        {
```

```
            LddRetVal = RTE_E_SecurityAccess_L9_DCM_E_PENDING;
          }
          else if (/* NvM block is in virgin state */)
          {
            /* If the NvM block is in virgin state,
               Application must determine AttemptCounter and return value appropriately. */
          }
          else
          {
            LddRetVal = RTE_E_SecurityAccess_L9_E_NOT_OK;
          }
        }
        else
        {
          LddRetVal = RTE_E_SecurityAccess_L9_E_NOT_OK;
        }
        break;

      case DCM_CANCEL:
        break ;

      default:
        break ;
    }

    return LddRetVal;
}
```

11.2.4 Xxx_SetSecurityAttemptCounter

If DcmDspSecurityAttemptCounterEnabled == true, Dcm will send the changed value to the application via Xxx_SetSecurityAttemptCounter when the Security Attempt Counter of a specific Security Level is changed.

Xxx_SetSecurityAttemptCounter passes a Security Attempt Counter of a specific Security Level to the application. The application must store the passed Security Attempt Counter in non-volatile memory and return the return value appropriately.

Below is an example of storing a Security Attempt Counter in NvM.

AppDcm_SetSecurityAttemptCounter_L9

```
/* Global variable for security attempt counter of security level 9 */
uint8 AppDcm_SecurityAttemptCounter_L9 = 0x00;

FUNC(Std_ReturnType, SWC_DiagnosticService_CODE) AppDcm_SetSecurityAttemptCounter_L9(
  IN VAR(Dcm_OpStatusType, AUTOMATIC) OpStatus,
```

```
  IN VAR(uint8, AUTOMATIC) AttemptCounter)
{
  Std_ReturnType LddRetVal = RTE_E_SecurityAccess_L9_E_NOT_OK;
  Std_ReturnType LddNvMRetVal = E_NOT_OK;
  NvM_RequestResultType LddNvMRequestResult = NVM_REQ_NOT_OK;

  /*
    Dcm must receive the result of NvM_WriteBlock() job from the application
    to store the attempt counter of current security level
    (1) to respond to a SecurityAccess request or (2) after the delay timer expires.
    Therefore, NvM_GetErrorStatus() must be called sequentially
    after NvM_WriteBlock() returns E_OK.
  */
  switch (OpStatus)
  {
    case DCM_INITIAL:
      /* Set the attempt counter of security level 9 */
      AppDcm_SecurityAttemptCounter_L9 = AttemptCounter;

      if (E_OK == Rte_Call_SWC_DiagnosticService_NvMService_DcmSecurityAttemptCounter_L9_WriteBlock(
          (const void *)&AppDcm_SecurityAttemptCounter_L9))
      {
        LddRetVal = RTE_E_SecurityAccess_L9_DCM_E_PENDING;
      }
      else
      {
        LddRetVal = RTE_E_SecurityAccess_L9_E_NOT_OK;
      }
      break;

    case DCM_PENDING:
      LddNvMRetVal = Rte_Call_SWC_DiagnosticService_NvMService_DcmSecurityAttemptCounter_L9_GetErrorStatus(
        &LddNvMRequestResult);
      if (E_OK == LddNvMRetVal)
      {
        if (NVM_REQ_OK == LddNvMRequestResult)
        {
          LddRetVal = RTE_E_OK;
        }
        else if (NVM_REQ_PENDING == LddNvMRequestResult)
        {
          LddRetVal = RTE_E_SecurityAccess_L9_DCM_E_PENDING;
        }
        else
        {
```

```
            LddRetVal = RTE_E_SecurityAccess_L9_E_NOT_OK;
          }
        }
        else
        {
          LddRetVal = RTE_E_SecurityAccess_L9_E_NOT_OK;
        }
        break;

      case DCM_CANCEL:
        break;

      default:
        break;
    }

    return LddRetVal;
}
```

11.3 Processing of NRC22 in relation to 'critical normal mode' in the application area

The following diagram shows part of negative response specifications of the CommunicationControl service as in ES95486-00E_V1.9.0.

22	conditionNotCorrect	M	Use when the server is in a critical normal mode activity and therefore cannot disable/enable the requested communication type.

The following diagram also shows part of Chapter 7.4.2.8 Service 0x28 - CommunicationControl in the AUTOSAR Diagnostic Communication Manager 4.2.2 Specifications.

> 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/manufacturer notifications can be used.

As in the case of SID28 CommunicationControl, critical normal mode has been applied to SID29 EnableNormalMessageTransmission (to be updated from specifications after ES95486-00E V1.9.0) and SID85 ControlDTCSetting for the negative response specifications. This means that the service cannot be processed for a reason the platform does not know.

Therefore, ServiceRequestSupplierNotification_Indication() callback service should be used to determine the condition during Service Request and let the application handle this. See below.

Note: The following pseudo code is for reference and cannot be simply applied to the project.

```
FUNC(Std_ReturnType,RTE_CODE)ServiceRequestSupplierNotification_Indication( ... )
{
  /* This is pseudo code... */

  If the condition cannot process /* SID28, SID29 and SID85 (critical normal mode)
     process DCM_E_CONDITIONSNOTCORRECT as negative response */
  if(Service is 0x28 && critical normal mode1 )
  {
    LddRetVal = E_NOT_OK;
    *ErrorCode = DCM_E_CONDITIONSNOTCORRECT;
  {
  else if(Service is 0x29 && critical normal mode2 )
  {
    LddRetVal = E_NOT_OK;
    *ErrorCode = DCM_E_CONDITIONSNOTCORRECT;
  }
  else if(Service is 0x85 && critical normal mode3 )
  {
    LddRetVal = E_NOT_OK;
    *ErrorCode = DCM_E_CONDITIONSNOTCORRECT;
  }

  return LddRetVal;
}
```

11.4 Implementation of NRC10 (General Reject) of the StopDiagnosticSession service

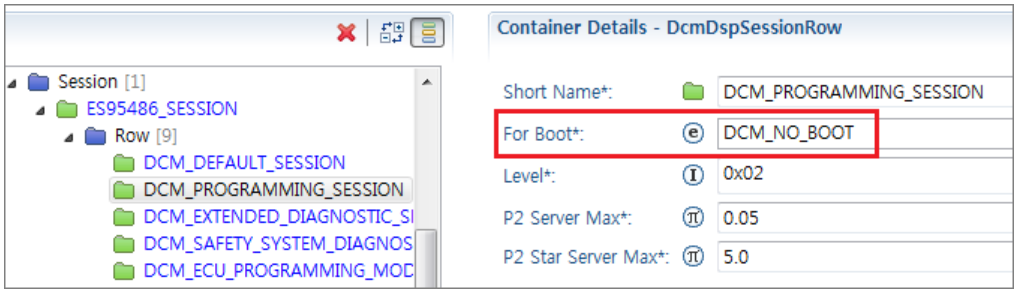
Ground of the specifications:

In the negative response specifications of the StopDiagnosticSession service, the condition that generates General Reject is described as follows.

GeneralReject (Programming session is running – Reprogramming routine is not completed yet.)

Prerequisite:

1) StopDiagnositcSession is in use
2) AutoEver Fbl is not in use
3) DcmDspSessionForBoot of Programming is DCM_NO_BOOT



If StopDiagnosticService is in use with 2) and 3) conditions met, the application area should determine "Reprogramming routine is not complete yet"…① condition during ProgrammingSession.

If Condition ① is met using the ServiceRequestNotificationIndication Callback feature, pushes NRC10 (General Reject) out (see the Sample Code below).

Note: The following codes are for sample and used only for reference.

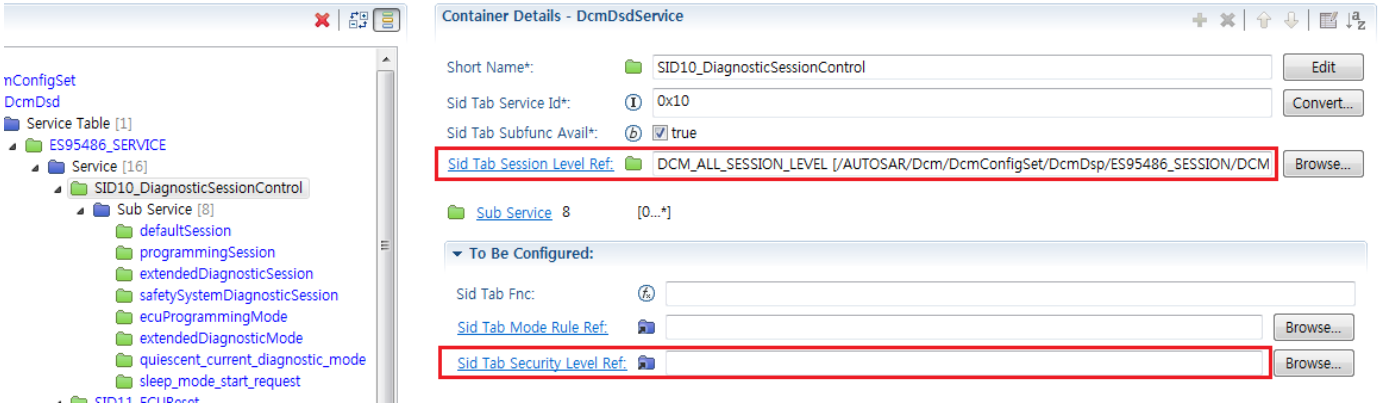
```
FUNC(Std_ReturnType,RTE_CODE) ServiceRequestSupplierNotification_Indication( ... )
{
  VAR(Std_ReturnType, RTE_DATA) LddRetVal;
  ...
  if(When StopDiagnosticSession service is requested)
  {
    if (Reprogramming routine starts || Reprogramming routine is not complete yet)
    {
      *ErrorCode = DCM_E_GENERALREJECT;
      LddRetVal = E_NOT_OK;
    }
```

```
    else
    {
      LddRetVal = E_OK;
    }
  }
  ...
  return LddRetVal;
}
```

11.5 Service, Session Level of the SubService, and Security Level Reference are set up

Note : Need to take a look from Dcm 1.4.0

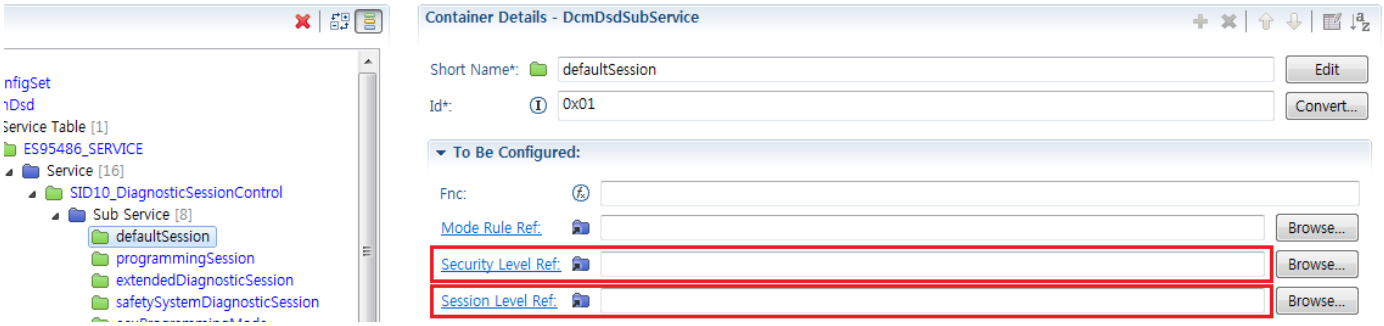
In DcmDsdService and SubService Container, there is a setup that can reference Session Level and Security Level as in the diagram below. The following setup was changed to changeable Parameter in Dcm 1.4.0.



[Figure 10.6-1: DcmDsdService Container]

- DcmDsdSidTabSessionLevelRef : It works only in the Session Level that Service is referenced according to the configuration. The session levels selectable are on the list below DcmDspSessionRow. NRC7F (serviceNotSupportedInActiveSession) occurs when Service is requested at a session level that is not referenced.

- DcmDsdSidTabSecurityLevelRef : It works only in the Security Level that Service is referenced according to the configuration. The security levels selectable are on the list below DcmDspSecurityRow. NRC33 (securityAccessDenied) occurs when Service is requested at a security level that is not referenced.



[Figure 10.6-2 : DcmDsdSubService Containter]

- DcmDsdSubServiceSessionLevelRef : It works only in the Session Level that SubService is referenced according to the configuration. The session levels selectable are on the list below DcmDspSessionRow. NRC7E (subFunctionNotSupportedInActiveSession) occurs when SubService is requested at a session level that is not referenced.

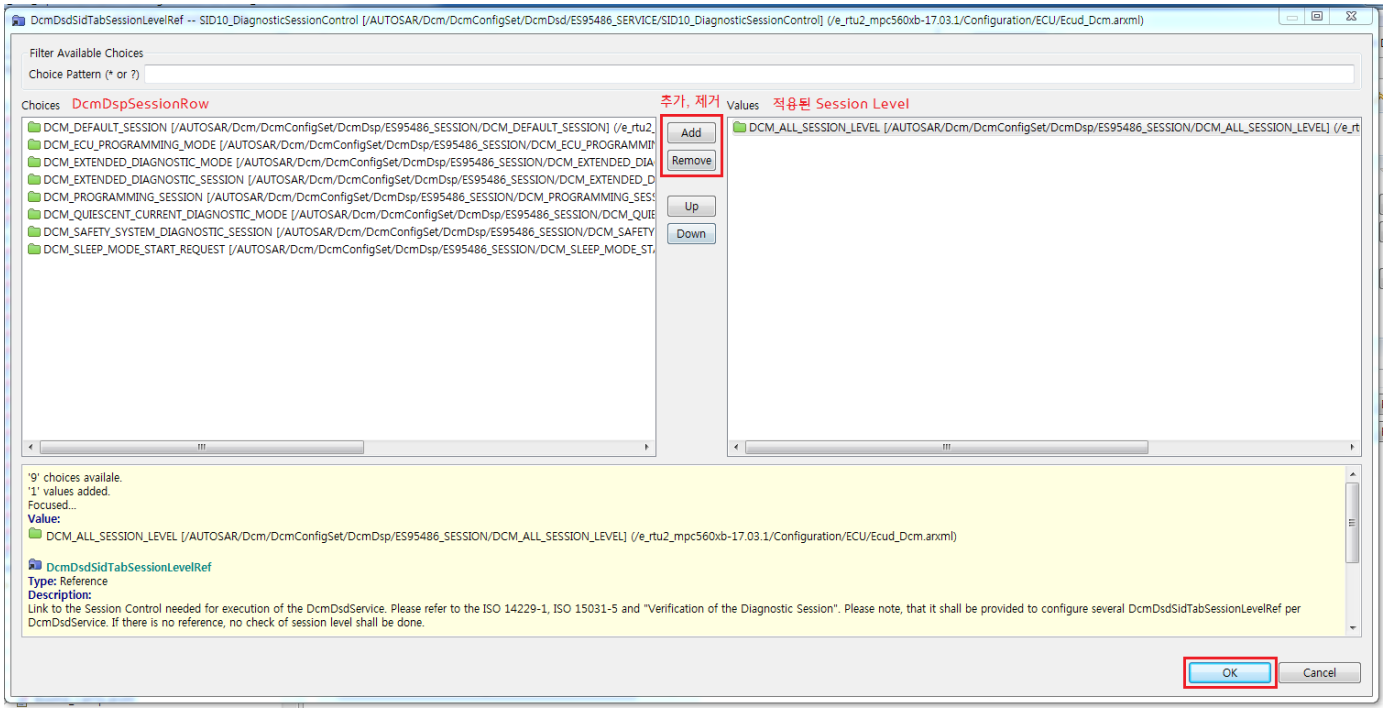
- DcmDsdSubServiceSecurityLevelRef : It works only in the Security Level that SubService is referenced according to the configuration. The security levels selectable are on the list below DcmDspSecurityRow. NRC33 (securityAccessDenied) occurs when SubService is requested at a security level that is not referenced.

The mobilgene sets and deploys the Session Level and Security Level based on the [Diagnostic Service List] of the ES95486-00 specification for the Service and SubService entered by users in the SRS.

If users want to change the above setting according to the controller specification, etc., it can be set in the following way.

How to Change Session Level

1) Find the Sid Tab Session Level Ref. (or Session Level Ref.) of the Service (or SubService) Container whose Session Level to be changed and click on the [Browse...] button.

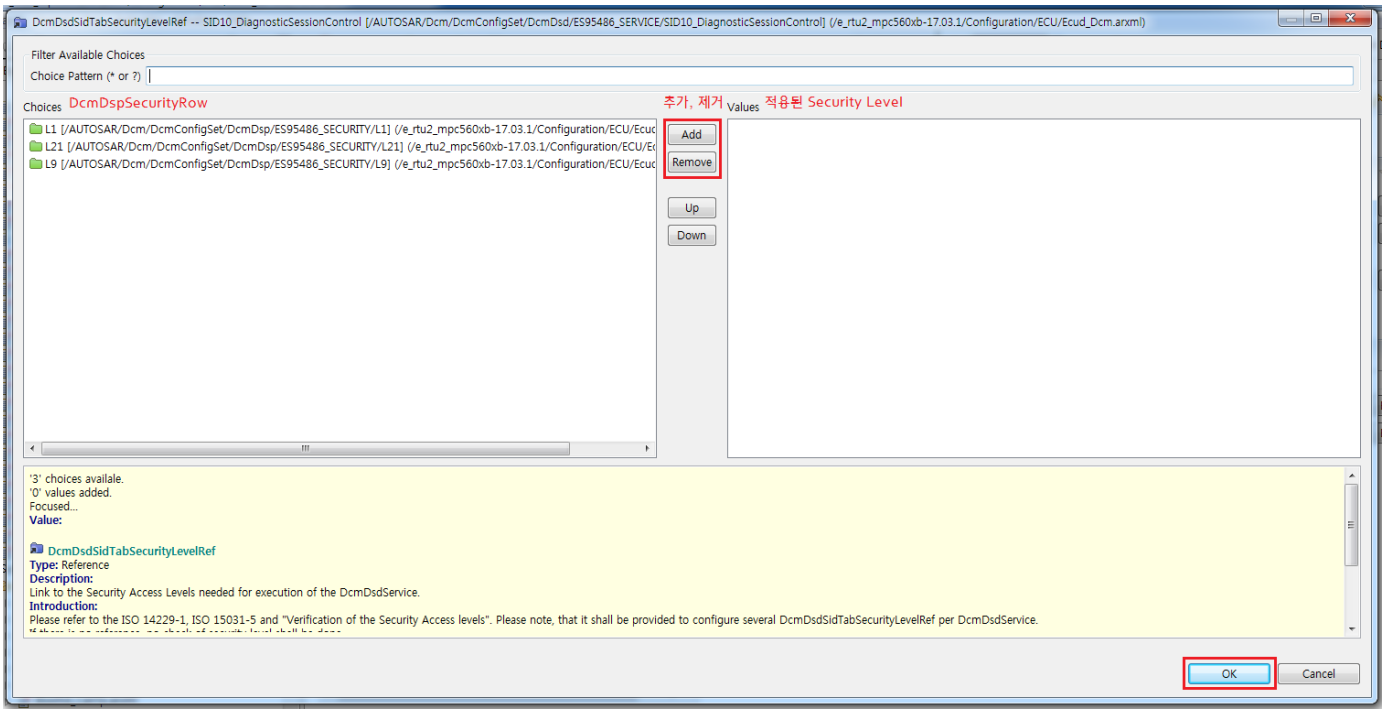


2) When the window appears as shown in the figure above, select the Session Level to be referenced in the Service (or SubService) through the [Add] and [Remove] buttons, and click [OK] to apply. The list in Values on

the right is the referenced Session Level.

How to Change Security Level

1) Find the Sid Tab Security Level Ref. (or Security Level Ref.) of the Service (or SubService) Container whose Security Level to be changed and click on the [Browse...] button.



2) When the window appears as shown in the figure above, select the Security Level to be referenced in the Service (or SubService) through the [Add] and [Remove] buttons, and click [OK] to apply. The list in Values on the right is the referenced Security Level.

11.6. OBD services

Path: DcmConfigSet [DcmConfigSet] > DcmDsd [DcmDsd] > ES95486_SERVICES [DcmDsdServiceTable]

Navigator

- DcmDsd
  - Service Request Manufacturer Notification [1]
  - Service Request Supplier Notification [1]
  - Service Table [1]
    - ES95486_SERVICES
      - Service [34]

Container Details - DcmDsdService

Index	Short Name	Role	Used	Sid Tab Fnc	Sid Tab Servic...	Sid Tab Subfu...	Sid
14	SID34_RequestDownload		true		0x34	false	
15	SID35_RequestUpload		true		0x35	false	
16	SID36_TransferData		true		0x36	false	
17	SID37_RequestTransferExit		true		0x37	false	
18	SID38_RequestFileTransfer		false		0x38	false	
19	SID3D_WriteMemoryByAddress		true		0x3D	false	
20	SID3E_TesterPresent		true		0x3E	true	
21	SID85_ControlDTCSetting		true		0x85	true	
22	SID86_ResponseOnEvent		false		0x86	true	
23	SID87_LinkControl		false		0x87	true	
24	SID01_ReqCurrentPowerTrain		true		0x01	false	
25	SID02_ReqFreezeFrameData		true		0x02	false	
26	SID03_ObtainDTC		true		0x03	false	
27	SID04_ClearDTC		true		0x04	false	
28	SID06_ReqOnboardMonitorR...		true		0x06	false	
29	SID07_ObtainDTC		true		0x07	false	
30	SID08_ReqControlOnBoardSy...		true		0x08	false	
31	SID09_ReqVehicleInfo		true		0x09	false	
32	SID0A_ObtainDTC		true		0x0A	false	
33	SID10_DiagnosticSessionCon...		true		0x10	true	

When using OBD services, they must be added to the existing DcmDsdServiceTable.